KNOWLEDGE MEDIA

KMi

INSTITUTE

A Document-Centric Semantic Annotation Environment to Support Sense-Making

Technical Report KMI-06-13
26th July, 2006

Bertrand Sereno

Sereno, B. (2006). A Document-Centric Semantic Annotation Environment to Support Sense-Making. Unpublished Doctoral Thesis, Knowledge Media Institute, The Open University, Milton Keynes, UK . Submitted May 2005, Approved July 2006 [http://kmi.open.ac.uk]

The Open University

KNOWLEDGE MEDIA KMi INSTITUTE

A Document-Centric Semantic Annotation Environment to Support Sense-Making

Bertrand Sereno

Thesis submitted in partial fulfilment of the requirements for the degree of Doctor of Philosophy

Submitted on the 31st of May 2005

The Open University

Acknowledgements

I realise as I am about to write this page how many persons have had an impact on my research in the last years, and how grateful I am to each of them, for having supported me in many occasions, and especially in the most difficult ones.
I would like first of all to thank the Knowledge Media Institute and the Open University, for having welcomed me in 2001 and believed in me. Both have given me all the tools I could ever need to pursue this research.
I would like to gratefully thank Simon Buckingham Shum for his supervision and support throughout these years. I have benefited immensely from his help and wisdom. This work would not have been possible without his support and without the many challenging and fruitful discussions that we had.
I am also indebted to my second adviser, Enrico Motta, for, once more, the fruitful discussions that we have had during our supervision meetings. As for Simon, I cannot find the words to express how much I have learnt from him. He also contributed immensely to make this PhD experience the most enjoyable possible.
I would like to mention and acknowledge the Advanced Knowledge Technologies (AKT) project that funded this research. AKT is an Interdisciplinary Research Collaboration (IRC) sponsored by the UK Engineering and Physical Sciences Research Council.

I would also like to thank all the participants of the two experiments presented in this thesis (many of them participating in both of them.) They gave me a lot of their time and their contribution had a major impact. I would also like to thank the anonymous reviewers of the different articles I submitted at several stages of my research: their advice and criticisms gave me additional ideas to improve this work.
I would like to mention my colleagues and fellow PhD students of the Knowledge Media Institute, and in other departments of the Open University, who also contributed to make the experience a lot better. The support and the smiles I received from them each and every time made all the difference.
These special thanks are also directed at my friends, on whom I have, and more than once, poured my doubts, my fears, my worries and my frustration. They have done all they could (and even more) to keep me on track and to make me feel supported. I must tell them here and now how well they succeeded and how grateful I feel.
Finally, I would say *merci*, to my parents and my family. *Merci* for having supported me through all these years. And *merci* for having given me the strength to follow my dream.

Publications

The work reported in this thesis has been presented and/or published in the following documents.

Conference and journal publications:

- Bertrand Sereno, Simon Buckingham Shum, and Enrico Motta. ClaimSpotter: an Environment to Support Sensemaking with Knowledge Triples. In *Proceedings of the 2005 International Conference on Intelligent User Interfaces (IUI05)*, San Diego, CA, USA IUI05, pages 199–206.

- Simon Buckingham Shum, Victoria Uren, Gangmin Li, Bertrand Sereno, and Clara Mancini. Computational Modelling of Naturalistic Argumentation in Research Literatures: Representation and Interaction Design Issues. *To Appear in the International Journal of Intelligent Systems (IJIS) Special Issue on Computational Models of Natural Argument*, 2005.

- Bertrand Sereno, Simon Buckingham Shum, and Enrico Motta. Semi-Automatic Annotation of Contested Knowledge on the World Wide Web. In *Proceedings of the Alternate Track (Papers and Posters) of the 13th International World Wide Web Conference (WWW2004)*, New York City, NY, USA. International WWW Conference Committee, May 2004.

- Bertrand Sereno, Simon Buckingham Shum, and Enrico Motta. Semantic Annotation Support in the Absence of Consensus. In *Proceedings of the 1st European Semantic Web Symposium (ESWS 2004)*, Heraklion, Crete, May 2004.

Unpublished articles:

- Bertrand Sereno, Simon Buckingham Shum, and Enrico Motta. Supporting the Construction of Networks of Claims. In *Presented at the 'Journée ATALA: Modéliser et Décrire l'Organisation Discursive à l'Heure du Document Numérique', held with the 'Semaine du Document Numérique' (SDN 2004)*, La Rochelle, France, June 2004.

- Victoria Uren, Bertrand Sereno, Simon Buckingham Shum, and Gangmin Li. Interfaces for Capturing Interpretations of Research Literature. In *Presented at the Distributed and Collective Knowledge Capture Workshop, held with the Knowledge Capture Conference (KCAP03)*, Sanibel, FL, USA. October 2003.

- Bertrand Sereno, Simon Buckingham Shum, and Enrico Motta. Towards a Semantic Hypertext Tool to Support Scholarly Reading and Annotation. In *Proceedings of the Hypertext'03 Workshop on Scholarly Hypertext*, Nottingham, UK. 2003.

Abstract

Prototype Internet infrastructures for scholarly publishing are offering powerful new services over the interconnected ideas and arguments in a literature. However, such services depend on documents being semantically annotated with readers' interpretations, which up until now has been a manual process due to the complexity of such analysis. This thesis investigates the challenge of designing computer-support for document annotation in the context of potentially diverse, contested views about a text's significance, as typifies scholarly research.

An interaction design approach is followed to progressively understand the dialogue between the end-users and an appropriate annotation environment. A preliminary analysis of the annotators' goals if followed by an experiment to identify the activities performed in this sense-making task, and a desk research phase, in which approaches to support each of these activities are identified.

An active document annotation environment is then presented. It is built on an open, extensible architecture that can incorporate new text analysis components as required to overlay annotations onto the original text to draw attention to sections which may be particularly significant. Facilities to filter and navigate the document in novel ways, to record annotations or reuse existing ones, and to provide pointers to related documents and annotations based on connections mediated by semantic annotations are offered.

The tool is finally evaluated in an experimental setting, resulting in a dataset that allowed quantitative and qualitative analyses of the end-users' products and process. The analysis characterises how the semantic annotation scheme is used by novices and experts, and how the user interface's rendering of system and end-user annotations shapes interaction. The thesis assesses critically the strengths and weaknesses of the work, providing justification for further cycles of the approach, and concluding with research questions meriting further investigation.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

Prototype Internet infrastructures for scholarly publishing are offering powerful new services over the interconnected arguments of a literature [Buckingham Shum et al., 2000, 2005]. These services depend on documents being annotated with readers' potentially contentious interpretations of their ideas, opinions, claims, rebuttals... and with assessments of their significance, an annotation that will be achieved with shared principles and languages. In other words, these services depend on documents being semantically annotated.

While this annotation has been until now a manual process due to the complexity of the analysis of the significance of scholarly documents, this thesis investigates the challenge of designing computer-support to assist it. This process can be seen as a cognitive task, in which readers have to contextualise these interpretations and assessments in the light of their own research interests, in order to transform them into knowledge they will be able to reuse, for instance to write their own document.

In this introductory chapter, we present the different domains this research relates to. We detail the context in which this work is carried out and we introduce several approaches already available to support the annotation of documents with their interpretations. We finally present our interaction design approach to the creation of an annotation environment and list the contributions of this thesis.

1.1 Scholarly documents enrichment

The annotation of scholarly documents (i.e. their enrichment [Sumner and Buckingham Shum, 1998, Motta et al., 2000]) can be assimilated to literature modelling, a cognitive task that requires the mental construction of a networked structure of arguments. We define this modelling task as the intersection of several research areas, which we introduce shortly. Figure

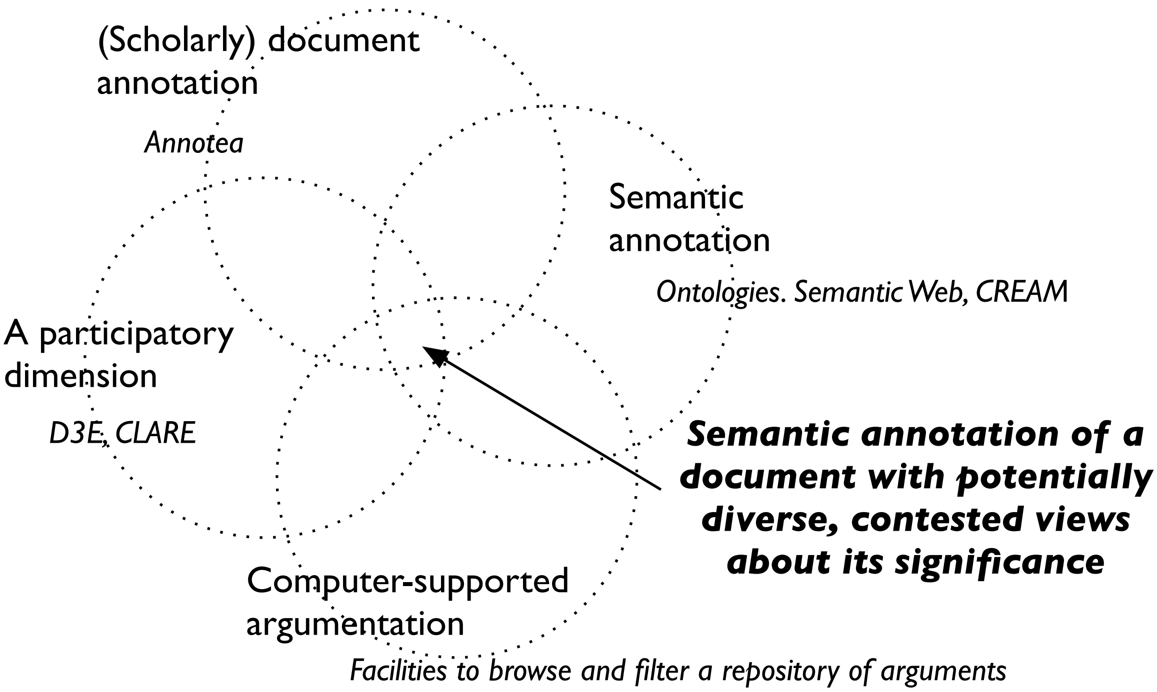


Figure 1.1: Our annotation task lies at the intersection of several research areas.

1.1 represents these areas graphically.

1.1.1 Annotation

The annotation of a scholarly document is a process that is not fundamentally different from the annotation of a 'normal' document. Although a scholarly document does exhibit strong characteristics, such as a specific goal which is often to persuade readers of the validity and the relevance of the work it reports [Swales, 1990], its annotation is still a means to record the ideas and interpretations we (as readers) have retained, including connections to additional scholarly documents, reformulations of the authors arguments, assessment of its significance or 'warning' signals to indicate key passages. In other words, the annotation of a scholarly document (i.e. our interaction with it) helps us make sense of its contents [Weick, 1985] and transform it into knowledge [O'Leary, 1998] that is contextualised according to our own research interests.

1.1.2 Semantic annotation

The Scholarly Ontologies project (or ScholOnto) [Buckingham Shum et al., 2000, 2005] offers an infrastructure and a language to model the potentially contentious interpretations of the content and connections of a scholarly document (it is presented in chapter 2, c.f. page 13.) In ScholOnto, annotations are not recorded as free-form text but are expressed as triples

(called *claims*.)

These claims connect *concepts* with *discourse relations* organised in an *ontology*. They can be submitted by the author of a document, if she wishes to reformulate her own arguments; they can also record the (potentially challenging) interpretations made by her peers.

While the product of an annotation - notes - is not constrained in a typical scenario, it becomes - in ScholOnto - more constrained in the sense that it has to be broken down into structures of ‘notes’ (or *concepts*) articulated via *(discourse) relations*, in order to create a networked structure of arguments. While the difficulty to write in such a formalism increases, flexibility is also improved, as these ‘notes-concepts’ can be reused for different interpretations and by different annotators.

The task annotators face can be reframed as an ontology-supported document annotation, or semantic annotation: (i) it is a document annotation, for personal content (concepts and claims) is added to a document; (ii) it is also a semantic annotation, for it (more precisely, the claim triples) contains some semantics, provided by the ontology of discourse relations.

Semantic annotation tools

Ontology-supported document annotation tools are already existing as part of the Semantic Web initiative [Berners-Lee et al., 2001]. They are usually supporting the annotation process by identifying elements to pre-populate the ontology. However, these applications are dealing with a specific brand of ‘annotation’: they are concerned with the ‘translation’, in a formal language, of information that is already contained in the document itself and which suffers little contention (such as person names, project names or locations.) This information is unlikely to be debated amongst different annotators.

Modelling a networked structure of arguments (i.e. ScholOnto concepts and claims), on the other hand, is a more cognitively demanding task. Such a structure involves contentious information that results from a personal sense-making process, in which an annotator interprets the information contained in a document and retains only the elements that are relevant to the task she is pursuing. It contains opinions about what is relevant, and not merely ‘translations’ of an author’s argument.

This personal dimension lies at the core of this research. Our goal is not to provide “*structured collections of information and sets of inference rules that (computers) can use to conduct automated reasoning*” [Berners-Lee et al., 2001], but rather to assist *humans* in their sense-making daily activity by offering them possibilities to record, interconnect and debate their interpretations. We therefore do not position this research as part of the ‘Semantic Web’ initiative, but rather as part of a 'Semantics *on the* Web' initiative.

1.1.3 A participatory dimension

These notions of contention and debate highlight the fundamental role of participation in the annotation process. Annotators add their own content to a document, representing their assessment of its contributions and (optionally) their critique of their peers' own assessments. As such, they contribute to the creation of a body of knowledge encompassing the document and the debate it has fostered.

1.1.4 Computer-supported argumentation

We hypothesise that annotators will only take the time and effort needed to record their interpretations if this task is facilitated and if services are provided (such as possibilities to browse and filter a repository or argument interpretations.) A recent article investigates these facilities [Buckingham Shum et al., 2005].

1.2 An interaction design approach

An efficient user interface is thus of paramount importance to ensure annotators invest the time and effort needed. This interface should initiate and sustain a productive dialogue with the annotator (or end-user.) The study, design and facilitation of this dialogue is the essence of interaction design [Cooper and Reimann, 2003]. This dialogue, or interaction, can also be defined as:

> "*...any communication between a user and computer, be it direct or indirect. Direct interaction involves a dialogue with feedback and control throughout performance of the task. Indirect interaction may involve batch processing or intelligent sensors controlling the environment. The important thing is that the user is interacting with the computer in order to accomplish something.*" [Dix et al., 2004, page 4]

We have therefore to understand the nature of this semantic annotation task and then to identify what tools can efficiently support it.

1.2.1 Semantic annotation tools for ScholOnto

Annotation in ScholOnto is currently supported by two user interfaces, ClaiMaker and ClaiMapper. In a few words (these applications, and the project itself, are presented in the following chapter), the former proposes a menu-and-form-based interface to record annotations, while the latter mimics a sketching pad letting readers 'draw' their annotations as graphs, in which nodes are assimilated to concepts and edges are assimilated to the discourse relations connecting these concepts into claims.

These two applications separate two activities that should be performed in parallel: reading a document and annotating it. This separation is a potential source of problems: annotators have to constantly switch their attention between the document that is currently annotated - whether in printed form, on a desk or displayed on the computer in a different application (running separately) - and the environment in which they can record their interpretations.

1.2.2 From passive to active semantic annotation tools

Another problem with these applications lies in the limited support they offer, in terms of what they actually suggest to the annotator to create concepts and claims. The original document is doubtless a very valuable source of information, and making use of it would help end-users record their semantic annotations. Of course, we cannot claim to translate the original author's argument into ScholOnto concepts and claims automatically and propose these to an annotator who could then position herself with respect to this 'formalised translation' of the original author's argument. This would require a deep level of scholarly prose analysis and comprehension that is not yet possible.

One of our research challenges is to clarify how far we can go, in terms of the (document) content-based support that we can provide. Such support would make the environment 'active', by contributing to the annotation process, proposing suggestions extracted from the document for instance and initiating a stronger dialogue, thus a stronger interaction, with the user.

Bringing the document back to the centre of the annotation process (it is of course at the centre of the process with ClaiMaker and ClaiMapper; however, the very fact that it is not part of the annotation interface tends to 'relegate' it into the background) raises several challenges related to the interaction with the computer and the way information is displayed. What can the computer actively contribute to assist this process, and what sources of document-based

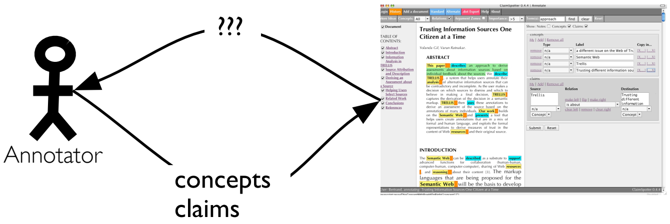


Figure 1.2: As annotators are interacting with the computer by submitting their concepts and claims, we investigate what computer-based support can be brought.

support it can provide, are at the heart of our thesis (c.f. figure 1.2.) While we review the literature on relevant text analysis techniques to identify what content-based support we can bring to support the semantic annotation of a document, this thesis investigates as its primary focus the dialogue between humans and the annotation environment.

1.3 Research questions

We define our research questions as follows:

(i) What document-based support can we provide to assist annotators in their interpretation, or claim modelling, task?

(ii) In what way is the process of interpreting a document, through the elicitation of concepts and their articulation in claims, influenced by the possibility to access its content in an interface, to visualise it and to modify its representation, and by the availability of additional resources?

We propose to bring the document back to the center of the annotation process. We also propose to investigate the benefits of an analysis of the content of the document to support its annotation, using overlaid suggestions to draw attention to elements which may be particularly significant. We approach this research process as an interaction-design problem. By focussing on the dialogue between the user and the designed artefact, our aim is to support sense-making without disrupting it. Finding out ways to present information in a *calm* way is an additional challenge this thesis investigates.

1.4 Methodology and outline

This section introduces the methodology we shall follow throughout this research process. We must bear in mind first of all that this annotation task exhibits strong characteristics (presented in section 1.1, page 1) that makes it different from other existing annotation tasks in the following ways: (i) it is a scholarly annotation task that, in contrast to most document annotation approaches, involves a formalisation of the 'notes' added to a document; (ii) it is an ontology-supported annotation task that, in contrast to most semantic annotation projects, requires a more cognitively demanding sense-making activity rather than a mere 'translation' of the information contained in a document; (iii) it has a fundamental participatory dimension; and finally (iv), it has to be efficiently supported, in order to enable readers/annotators to reap the benefits of their efforts.

An additional difficulty directly related to this unique positioning is that we do not know (yet) which tasks and activities are involved in a ScholOnto annotation process. This thesis is indeed the first study of annotators actually authoring annotations and the first aspect we have to research is an analysis of the tasks carried out by annotators to make sense of a document; in more interaction design oriented words, we have to understand the dialogue between the end-user and the artefact that we are going to design (c.f. section 1.2, page 4.)

1.4.1 Methodology

To understand this dialogue, we make use of some elements of a model called *goal-directed design*, defined in [Cooper and Reimann, 2003].

Goal-directed design model

This model puts the goals, motivations and expectations of the prospective end-users (defining altogether their *mental models*) at the centre of the design. Goals and tasks are different: Cooper and Reimann define the former as being driven by human motivations only, and thus as not being likely to change very often; the latter are instead a more transient notion, based on the technology available. In our context, an end-user's goal could be "*I want to become more knowledgeable about my research area, by gaining a clearer picture of the arguments expressed in its core publications*"; while an end-user's task could be "*I will use ClaiMapper to record what I have retained from this paper and to draw a connection to this paper I have read last week.*"

Different approaches are available to represent goals as a model (a model is defined as a set of inter-related concepts; it enables not only the visualisation of its constituents, but also of their inter-relationships.) Personae, for instance, can be used to represent fictional characters who will be prospective users of the designed artefact. The creation of an additional 'story' explaining how these personae have come to the application being developed, their expectations, their circumstances and their personal *goal*, can be used as a discussion basis

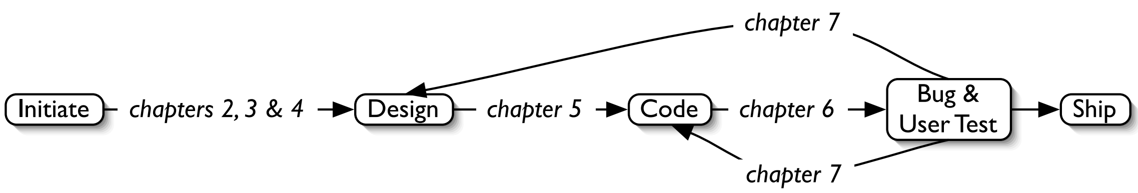


Figure 1.3: Design cycle of our active document-centric semantic annotation environment.

in the design phase. Personae for a semantic annotation tool could involve an experienced researcher, a PhD student or a member of a publisher's reviewing panel (c.f. section 7.4, page 235.)

We have mentioned earlier that we would not use a full goal-directed design model, but only elements of it. Our own approach to understanding the goals of the prospective users of an active document-centric semantic annotation environment combines an analysis of the goals underlying a modelling process with an analysis of the tasks performed to carry this process out. The latter is enhanced by an additional research phase.

Iterative design and prototyping model

Software engineering is not only concerned with the development of a computer-supported artefact, but also with the satisfaction of certain criteria, most commonly the input (the resources applied to the creation of the artefact) or output of a software development task (the quality attributes exhibited by the result of the task) [1].

This iterative process relies on frequent tests to ensure that the design satisfies the requirements identified by users. It results in several prototypes (proposing each or only a few functionalities) being built and evaluated, in order to improve the overall quality of the product. Each prototype serves as the basis for the next iteration of the design life-cycle. This process is called evolutionary prototyping [Cooper and Reimann, 2003, page 6], and it is used to build our environment (c.f. figure 1.3.)

1.4.2 Outline

This section presents the chapters of this dissertation in more detail.

Chapter 2, page 13

We start our design phase with a detailed presentation of the Scholarly Ontologies project to set up the context in which this work is carried out. We then present an analysis of end-users'

[1] `http://www.cs.utexas.edu/~sahilt/research/SEMyths.html`

goals using a taxonomy of networked structures of arguments that can be built by fictional personae in ScholOnto. Two semantic annotation interfaces for ScholOnto, ClaiMaker and ClaiMapper, are presented and critically assessed. We end this chapter with our first research question and our motivation for an alternative interface providing content-based support.

Chapter 3, page 29

To identify the tasks carried out by end-users to reach their goals, we devise an initial annotation experiment. Chapter 3 introduces a paper-based study, in which a group of subjects is asked to read a scholarly article and to answer four questions about its contributions and its connections to the literature. This experiment helps us identify the tasks performed by annotators. It also reveals two interesting observations: (i) potentially interesting snippets can be spotted from the text and proposed to the annotators' consideration; and (ii) a given snippet can be used for different purposes by different annotators. These are transformed into an initial set of requirements for our environment.

Chapter 4, page 39

In chapter 4, we refine our initial tasks analysis with a detailed study of supporting approaches for sense-making. We first define a ScholOnto annotation process as a sense-making process and break it into several activities. For each of these, we review the research literature to critically assess supporting approaches. This review features approaches to scaffold annotation, approaches to support reading a document on-screen; approaches to extract candidate ScholOnto concepts and claims from the document; approaches to extract coherent passages; approaches to discover related documents; and finally approaches to support the participatory dimension of our sense-making process. We give our final set of requirements at the end of this chapter, and propose to make a selection of these approaches available in our document-centric annotation environment.

Chapter 5, page 103

Chapter 5 details our coding phase and the integration of our goals and tasks analysis into an environment, ClaimSpotter. Its architecture and its user interface are reported. We also present a virtual tour illustrating a fictional annotator making sense of a scholarly document, using the different sources of support available.

Chapter 6, page 153

In chapter 6, we define and evaluate ClaimSpotter usability. We test the impact of ClaimSpotter on a pool of annotators asked to annotate a document of their choice. Their interactions with the tool are recorded and analysed. We conclude with a revised understanding and perception of the sense-making process.

Chapter 7, page 219

In this chapter, we give a critical assessment of the strengths and weaknesses of ClaimSpotter based on the evaluation study of the previous chapter. Weaknesses are discussed further and future research questions are identified (which would be additional design-code-test loops in our development life cycle (c.f. figure 1.3.)) As an initial step towards a refined analysis, we report on a rapid design cycle, in which we present preliminary work on one of these questions. We conclude this chapter with a presentation of two scenarios in which ClaimSpotter and the underlying ScholOnto can be used, followed by three scenarios in which ClaimSpotter can be used without the ScholOnto model.

Chapter 8, page 239

We give in chapter 8 our final conclusion of this dissertation.

1.5 Contributions

The contributions of this thesis are:

- an analysis of the dialogue taking place between an active document-centric semantic annotation environment and its end-users.

  To analyse and understand this dialogue, a preliminary analysis of annotators' goals is performed, followed by an experiment to identify the tasks they perform to reach these goals. A detailed study frames the annotation process as a sense-making process, composed of several activities run in parallel. We review the literature to find out which approaches are the most potentially suitable for our annotation environment.

  Our ClaimSpotter prototype is then presented. ClaimSpotter (i) is built on an open, extensible architecture which can incorporate new text analysis components as required; (ii) uses text analysis techniques to overlay annotations onto the original text to draw attention to sections which may be particularly significant; (iii) offers facilities to filter and navigate the document in novel ways; (iv) facilitates the recording of new semantic annotations or the reuse of existing ones; and (v) provides pointers to related documents and annotations based on connections mediated by semantic annotations.

- a data-set illustrating how end-users author semi-formal annotations of scholarly documents.
- an evaluation study, in which a definition and an assessment of the impact and usability of our annotation environment are given, using qualitative and quantitative evaluation techniques. This study also compares the use of the annotation scheme by experts (with the scheme) and beginners. It provides the grounds for an assessment of the strengths and weaknesses of the current prototype and highlights additional shortcomings, leading the way to an additional design phase and a richer understanding of the dialogue taking place between artefact and end-users.

Chapter 2

The Scholarly Ontologies project

The Scholarly Ontologies project (or ScholOnto) proposes an approach to represent knowledge which by essence is open to interpretation and debate, such as the arguments defended in scholarly publications [Buckingham Shum et al., 1999]. We introduce in this chapter the language with which models of these arguments are built (as we have seen, via annotation.) We then analyse the goals of prospective ScholOnto users with a taxonomy of models. We conclude with a presentation of current ScholOnto annotation interfaces and present in more detail our first research question.

2.1 Language

These argument models, or literature models [1], represent readers' interpretations of the contents of scholarly publications. They are captured in ScholOnto with several directed triples (or claims) {*source concept*, ***discourse relation***, *destination concept*}, in which *concepts* are unconstrained - in their content - but the ***discourse relation*** (or ***relation***, or ***relation type***) is chosen from a formal ontology of discourse relations. The language proposed in ScholOnto can therefore be considered semi-formal, as constraints are put only on the relations used to connect concepts.

We focus on the notions of concept, claim, claim space, annotator, and finally debate and discussion. These need to be well understood as they are a central part of our annotation environment.

[1] We have also referred to them earlier as 'networked structure of arguments.'

2.1.1 Concepts

Concepts are not to be taken in a strict acceptation: they are closer to the idea of 'tags', rather than 'strictly and carefully crafted elements of knowledge that have to be agreed upon', as in more classical ontologies (concepts and ontologies are briefly presented in section 2.1.2.)

An optional type can be added to a *concept*, to be chosen among *analysis*, *approach*, *assumption*, *data*, *definition*, *evidence*, *hypothesis*, *language*, *methodology*, *model*, *opinion*, *phenomenon*, *problem*, *solution* or *theory*. It is possible to assign as many of these types as desired to a single concept, by creating as many instances of this concept and assigning the different types to each of these instances. Concepts are also grounded in documents: documents provide the evidence for concepts.

The following text strings are 'valid' examples of *concepts* and illustrate the freedom left to an annotator [2]:

- *ScholOnto*
- *Semantic Web*
- *Enhancing a document with machine readable information*
- *Finding out where an idea comes from is a particularly difficult problem/Opinion*
- *One third of the people polled in the survey did not answer/Data*

The last two concepts have an optional type added to them, while the first three do not. A new annotator could create an instance of each of these concepts and contextualise it with another type of her choice. For instance, a similar concept '*One third of the people polled in the survey did not answer*' could be tagged as a *problem* by another annotator.

Figure 2.1 gives a graphical representation of several concepts grounded in their respective documents. The figure also shows how different annotators can retain different elements from a single document and how these interpretations coexist.

2.1.2 Claims

An ontology of discourse relations organises how concepts can be articulated [3]. It contains relations such as ***addresses***, ***proves*** or ***is consistent with*** (figure 2.2 lists the relations avail-

[2] Any text string is acceptable in ScholOnto, since it does not have to be validated.

[3] An ontology defines a mediated view on the concepts of a domain and their interrelationships, in other words '*an explicit and formal specification of a conceptualization*' [Gruber, 1993]. It does in this respect share aspects with other hierarchical structures such as controlled vocabularies, taxonomies or thesauri.

An ontology shares the notion of explicit and unambiguous concepts with a controlled vocabulary, but adds the notion of relationships holding between these entities. A taxonomy provides a hierarchical structure of

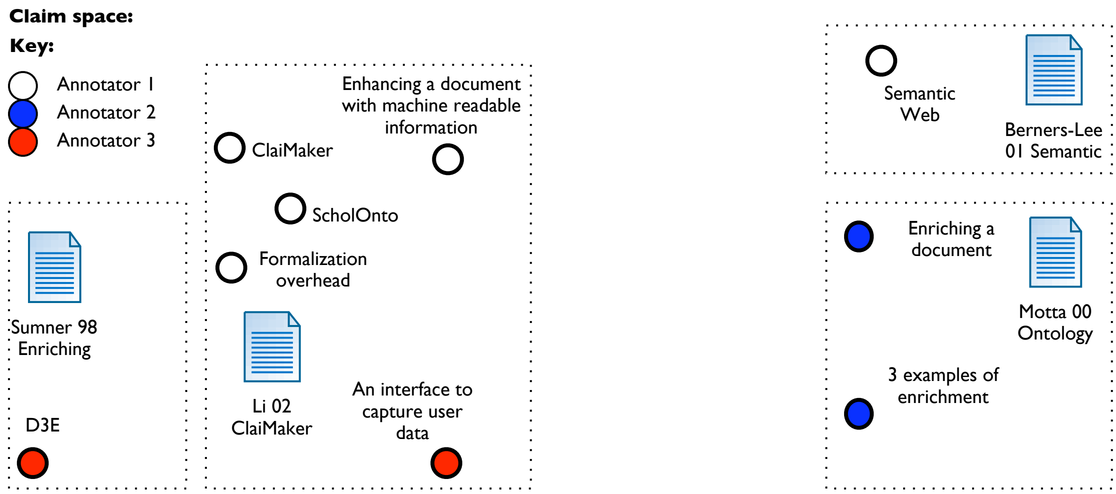


Figure 2.1: ScholOnto concepts are grounded in documents and act as 'tags' representing one's interpretation of their contents. In this example, the concepts associated to document *li02claimaker* by *annotator1* and *annotator3* show the different aspects both of them have retained from this document.

able.) The set of relations currently in the ontology is conceived to capture the contents of documents containing a problem or an approach proposed to tackle it. It is acknowledged that not all research areas would be modelled appropriately with this set of relations: for these areas, a possibility to translate the ontology to make it suitable to their vocabularies is offered.

Claims are written in ScholOnto by connecting *concepts* (although it is possible to connect claims into other claims; more about this later.) Claims are directed triples: they have a source and a destination element. As for *concepts*, claims are attached to documents and documents provide the necessary evidence to back up claims.

Here are several claims which can be created (we use a {*source concept*, ***relation type***, *destination concept*} representation for claims throughout this thesis):

- {*ScholOnto* , ***is about***, *Semantic Web*}
- {*Enhancing a document with machine readable information*, ***supports***, *Semantic Web*}
- {*ClaiMaker*, ***addresses***, *Formalisation overhead*}

---

concepts organised with parent-child relationships, which an ontology extends with richer sets of relations which can break the tree-like structure and replace it with a cyclic structure. A thesaurus, finally, is the closest structure to an ontology as it incorporates 'transversal', *is related to*, links to connect concepts in different regions of the structure; an ontology defines however a richer relationship set.

Different types of ontologies can be created: an upper ontology can for instance be written to conceptualise more generic statements about the 'world' which can be reused in different contexts; a domain ontology, on the other hand, can describe the specific entities of a given 'world' (or domain) and their interrelationships. The ScholOnto ontology of discourse relations can be seen as a mixture of a generic and domain-specific ontology that defines how potentially competing world views can be debated.

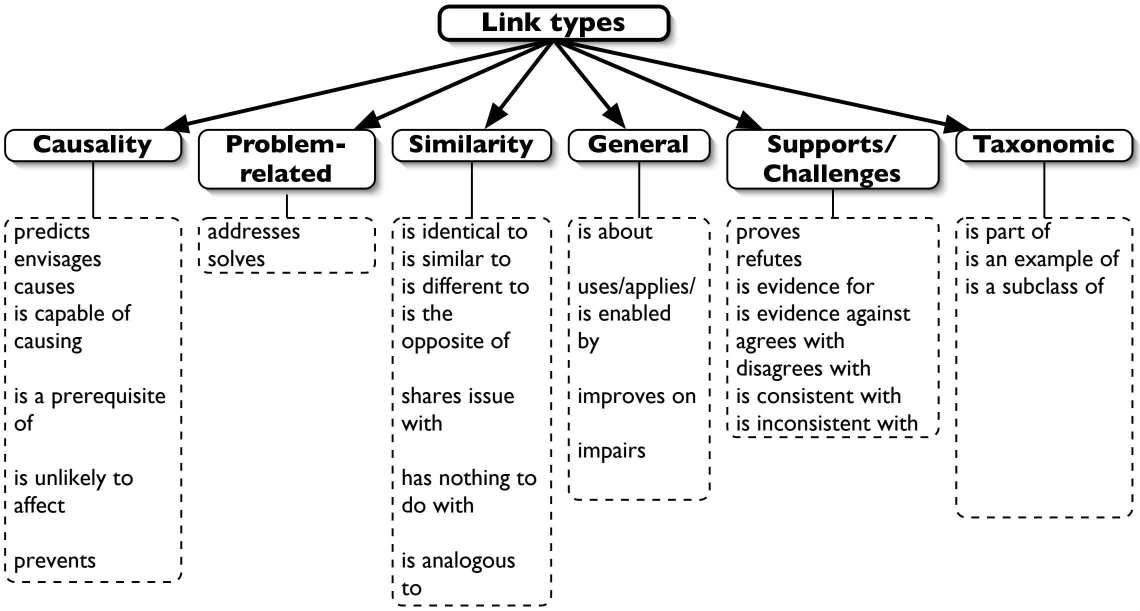


Figure 2.2: Relation types are organised into six broad categories.

- {*One third of the people polled in the survey did not answer*, ***is evidence for***, *this problem is not perceived as crucial/Opinion*}

Figure 2.3 shows several claims drawn by annotators over several documents. Claims can connect concepts defined by a single annotator (or by separate annotators), over a single document (or between separate documents.)

2.1.3 Claim spaces

Figure 2.3 also shows how annotations become interconnected, as annotators start to reuse their peers' concepts and connect them in their own claims. A networked structure of arguments (or 'claim space') emerges, modelling explicit connections between interpretations of the arguments defended in a corpus of documents.

By filtering the content of this network of semi-formalised statements (ways to filter and query a claim space through specialised services are reported in [Buckingham Shum et al., 2005]), queries can be issued to answer particular questions. It is possible for instance to list the concepts grounded in a single document or to retrieve the different approaches that have been proposed to address a particular problem, by following ***addresses*** claims pointing to a *problem* concept, for instance. The explicitness of the relations used to connect concepts ensures (provided that annotators agree with their signification) that the emerging network is understood coherently.

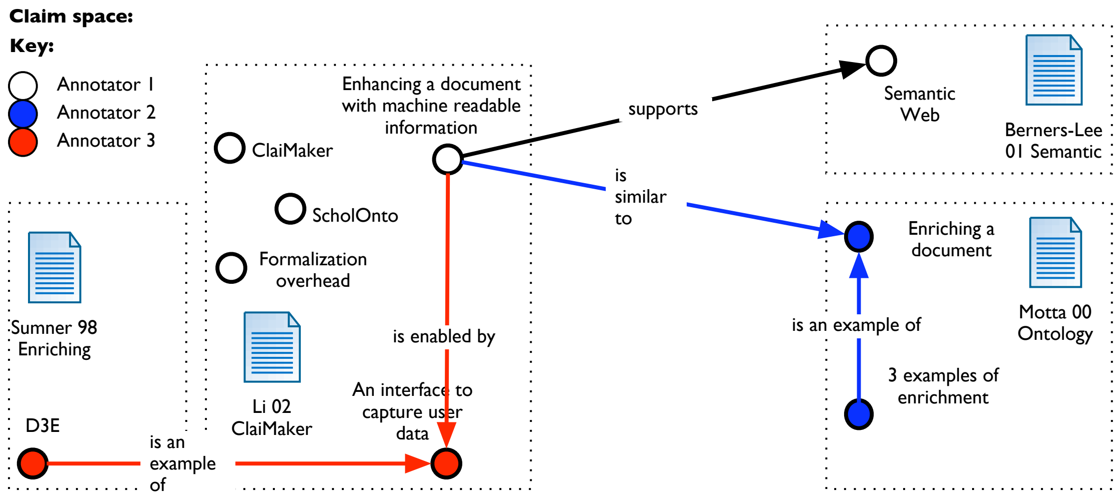


Figure 2.3: Claims submitted by different annotators contribute to the creation of a collective memory. They are directed triples {*source*, ***relation type***, *destination*} which can connect concepts defined over separate documents, made by separate annotators: the claim {*Enhancing a document with machine readable information*, ***is similar to***, *Enriching a document*}, for instance, is defined by *annotator2* between a concept defined a peer and one of hers.

2.1.4 Annotators

An annotator may be either reading a document and wanting to record what she retains of its argument; or wishing to express its argument in ScholOnto terms to increase its visibility (in a scenario in which claim spaces would be shared and publicly visible.)

Shallow and deep interpretations

An annotator may spend little time annotating if the document is only of normal relevance to her work, or if she is only interested in a particular point (for instance if she is knowledgeable about the domain and wants to remember one single fact about a document.)

Conversely, an annotator may have to, or may want to, spend more time annotating, separating and organising the arguments defended in a paper that is of great importance to her. We may also imagine a student spending more time on the annotation as part of a training process. We return to this notion of goals in section 2.2, page 19.

A participatory effort

Annotating a document is also a participatory process. As concepts and claims are signed with their author's name, their significance can be assessed in the light of their creator's name and reputation. Future annotators can also benefit, should they wish to, from their peers' opinions: a claim can be for example reused (duplicated) by a new annotator if she wants to model a similar interpretation. A claim may also debated and discussed.

2.1.5 Debate and discussion

While, in principle, a document may include aspects upon which everybody agrees on - as for instance the founding and uncontroversial statements defining a community of researchers -, it may also contain more contentious statements, triggering debate and discussion. For example, a single paper may be interpreted in different ways by researchers coming from competing schools of thought.

Agreement on the interpretations is not enforced in ScholOnto, and it is possible to record contesting interpretations. To give an example showing how this is supported by the language, let us assume that a claim between *Approach 1* and *Approach 2* has been submitted, using the relation ***is similar to***. This annotator wants to express her belief that *Approach 1* and *Approach 2* are similar (by doing so, she may be reformulating the author's statement or recording an approval of this statement.) Let us imagine now that another annotator wants to express her belief that *Approach 1* and *Approach 2* are not similar, but different. She can consider several alternatives:

- She can first of all create another claim connecting *Approach 1* and *Approach 2* using the appropriate relation: this relation would stand on equal ground with the first one. For instance, she can submit a claim {*Approach 1*, ***is different to***, *Approach 2*}.

- She can also decide, in a strategy similar to the one explained in figure 2.4, to create a concept containing *evidence* to back up her opinion [4], and connect it with the claim she disagrees with, using for instance a relation ***is evidence against*** [5] She can therefore submit a concept *evidence* and then a claim {*evidence*, ***is evidence against***, *{Approach 1,* ***is similar to****, Approach 2}*}

- A third possibility can be to model her claim with a relation ***is different to*** and then to connect the two claims together (hers using ***is different to***, and the one she disagrees with, which used ***is similar to***) with a relation ***is different to***. She has to submit a claim {*Approach 1*, ***is different to***, *Approach 2*} and another claim {*{Approach 1,* ***is different to****, Approach 2}*, ***disagrees with***, *{Approach 1,* ***is similar to****, Approach 2}*}.

While these alternatives are all supported by the ScholOnto language, they exhibit a difference in terms of the strength with which they convey an opinion: the first one is probably

[4] She first has to bring a supporting document into the claim space, since concepts in the ScholOnto model are grounded in documents.

[5] We note that claims can also connect claims to concepts, or claims to claims.

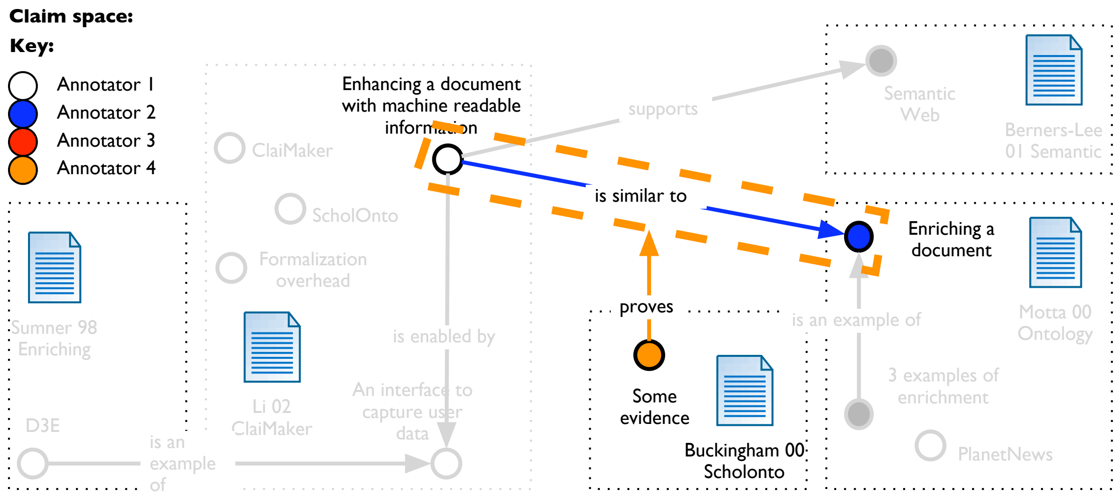


Figure 2.4: Discussion and debate in the claim space can be achieved by chaining claims, i.e. using a claim as one end of another claim. In this example, *Annotator 4* agrees with *Annotator 2*'s claim and adds some *evidence* to strengthen (or ***prove***) it.

the kinder one (as it does not model explicitly a disagreement but only a difference of opinion) while the third one is the most vehement one.

2.2 Literature models (∼ goals analysis)

We come back now to the content of these annotations (which we have eschewed in our shallow and deep interpretations paragraph) in order to analyse the end-users' goals underlying a ScholOnto modelling process.

2.2.1 Models dimensions

We propose to study two dimensions to differentiate the families of models (or annotations) we envisage to be submitted: commitment and granularity.

Descriptive models vs. committed ones

The first dimension we consider is the level of commitment. As concepts and claims are signed and visible, situations in which an annotator does not want to commit herself too strongly can arise, whether because she is not feeling secure enough about what she is about to commit, not familiar enough with the domain or not confident enough with the formalism yet. In these situations, she may decide to submit a few basic concepts and claims only - 'basic' being taken in the sense 'non-committed.'

Descriptive statements Although ScholOnto is more concerned with interpretations of arguments rather than representations of the key concepts of a domain, no rule prevents annotators from modelling the latter. Such models would include for instance statements about the state of the art in the field, or positions that have become accepted enough to be characteristic of a discipline. Although this knowledge may still be of interest to a newcomer in the discipline, it would bear little commitment from its creator.

Committed statements At the other end of the spectrum, annotators may feel confident with the paper they are annotating and state their opinion about its argument, taking position with its author if needed. These claims would model more explicitly what the annotator believes of the work being described and would convey more commitment.

There are of course various shades of grey between these two extremes. An annotator may feel confident about a particular point of the work being described and submit a strongly-committed ScholOnto claim, while at the same time feel less confident about another point and (provided she wants to say something about it) submit a less descriptive claim about it.

Unstructured models vs. structured ones

The second dimension we consider to characterise ScholOnto annotations is their level of granularity. As concepts are unconstrained, no rule prevents an annotator from considering them as notepad sheets, containing a single ‘concept’ gathering all their comments about a document.

Even though these are perfectly valid, they are not the most ‘interesting’ ones as they have limited reusability (either by the same annotator, or by fellow annotators.) By breaking such notes into smaller ones, they would become more reusable (for they contain less information) and would be more easily ‘connectable’ in a new claim.

There is of course a trade-off here, for which there is no satisfactory answer. Concepts should be explicit enough to represent a significant statement from an annotator, and at the same time be generic enough to be reusable. It is impossible to define a single strategy: there will be times when fine-grained concepts will be more appropriate (including for instance names of languages or methodologies); there will also be times when coarse-grained concepts will be kept in order to model an opinion about a particular aspect (if this opinion cannot be expressed with one of the discourse relations available in the ontology, i.e. cannot be broken down into a claim.)

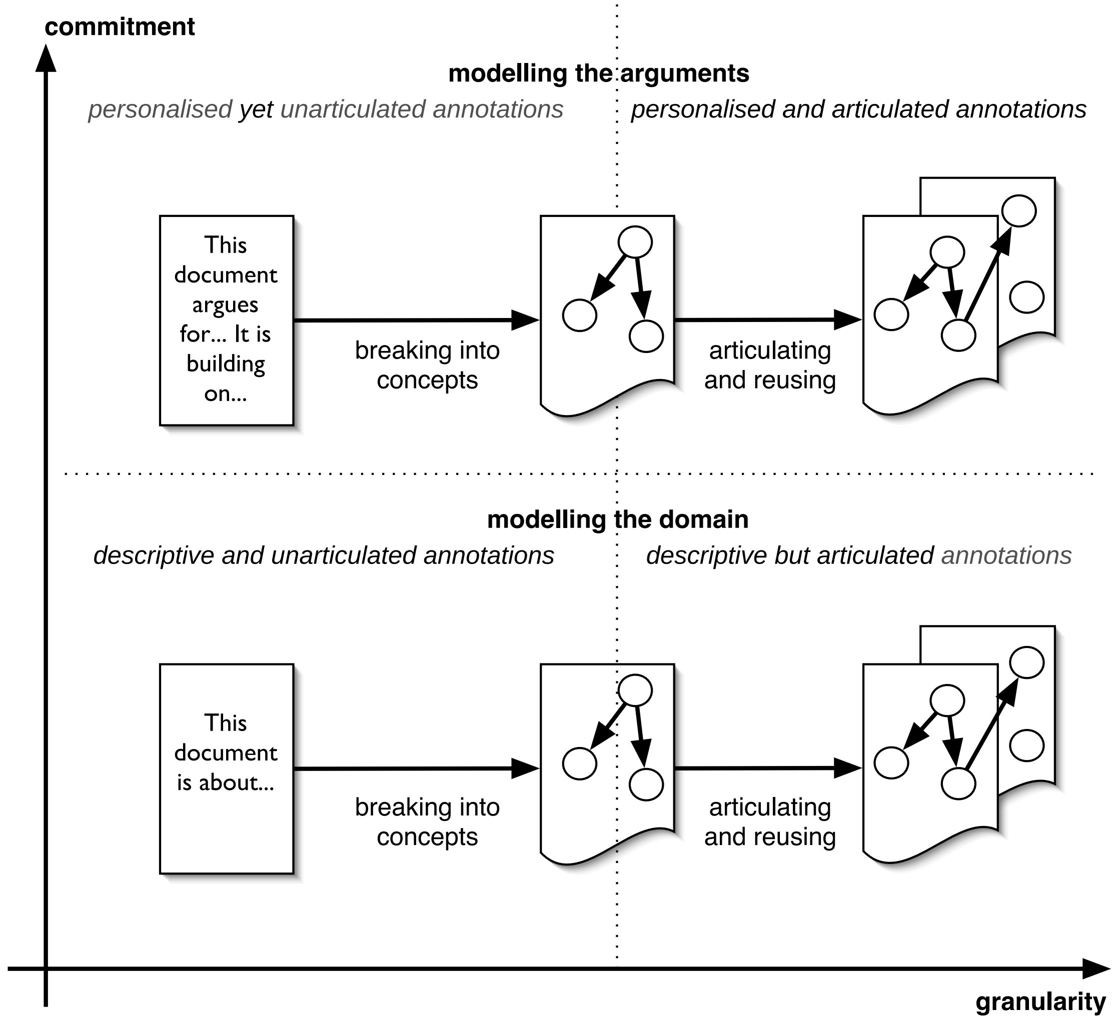


Figure 2.5: Characterising differences between annotations in terms of two dimensions, commitment and granularity, breaks the space of possible literature models into four different categories. Annotations in the upper right one are the most desirable ones.

2.2.2 A taxonomy of models

Bringing these two dimensions together breaks the space of models into four areas, as represented in figure 2.5. Commitment is on the vertical axis, and granularity is on the horizontal one. We give a few examples of annotations for each of these zones, which we pick from the observation study presented in chapter 6 (the full data for this evaluation study is available in appendices C (c.f. page 251) and D (c.f. page 261.)) Assessing how descriptive or committed an annotation is, and how coarse- or fine-grained it is, is however a personal judgement. The examples we give below, and their categorisation, should be seen as this dissertation author's interpretation only.

Descriptive, coarse-grained models

The lower left quarter contains what we could term the 'worst' kind of model, in ScholOnto terms. They are made of descriptive statements (concepts and claims) about the document bearing little commitment from the annotator, and they are composed of 'large', coarse-grained, concepts.

Examples of interpretations in this category include:

- *two types of behaviour of entities on the Web: first, same entity means different types of things on different domains; second, some entities mostly appear on a certain domain and are not likely to appear on the other domains*
- *the following seven PSMs: Hill-Climbing, Propose & Backtrack (P&B), Propose & Revise (P&R), Propose & Exchange (P&E), Propose & Genetic-Exchange (P&GE), Propose & Restore-Fea*

Let us restate that although these annotations seem to be descriptive, and therefore that they seem to bear little commitment from their authors, they may still be of interest to an annotator. The point of this discussion is not to judge their quality, but only to study their levels of reusability and commitment.

The second concept could have advantageously been expressed as a set of eight concepts (a concept *PSM methods* and seven concepts for each PSM method, connected with seven claims connecting each 'PSM method' to the concept *PSM methods*: {*Hill-Climbing*, ***is an example of***, *PSM methods*}, {*Propose & Backtrack*, ***is an example of***, *PSM methods*}... Each concept would be more reusable, in a different context, than the single original one. This model would still be descriptive, but finer-grained.

Descriptive, fine-grained models

Annotations in the lower right quarter are just as committed, but they are also broken into multiple nodes, which are potentially more reusable. The interconnectedness of the claim space is also increased.

Annotations such as:

- *VIPERS*
- *WWW*
- {*ScholOnto* , ***is about***, *Semantic Web*}

express noncommittal statements connecting fine-grained concepts (which may have been reused from another annotation, and which may be reused in a future annotation.)

Some relation types require less commitment, while others bear more expressiveness and therefore require a greater confidence to be used. The claim given above could have been made stronger, had its author wanted to, with a stronger characterisation of the link between the concepts *ScholOnto* and *Semantic Web*: claims such as {*ScholOnto* , ***proves***, *Semantic Web*} or {*ScholOnto* , ***is evidence for***, *Semantic Web*} would imply more commitment. We come back to this aspect on our evaluation chapter (c.f. chapter 6, page 153.)

Committed, coarse-grained models

On the upper left quarter, we find models which are this time more interesting in their content (for they express committed statements) but which are composed of broad entities.

For instance:

- {*a research project aiming to explore the potential of spontaneous social behaviour and playful group interaction in public spaces*, ***uses/applies/is enabled by***, *mobile technologies*}

is a claim showing commitment from the annotator, as it details a requirement of an approach. The source concept however is not easily reusable. It could be broken down into several smaller ones, such as *spontaneous social behaviours*, to increase reusability.

Committed, fine-grained models

Annotations in the upper-right corner are both committed and reusable. In this area, we also find 'chained' claims, i.e. claims connecting existing claims. As they express positions taken with other claims (coming from other annotations), they show a strong level of commitment by their author.

Example of such claims include:

- {*Hypertext nodes*, ***is analogous to***, *cinematic shots*}

- {*In the Bristol trial, the awareness of the presence of other players was correlated with how much our participants enjoyed the game as well as with how engaged they felt*, ***is consistent with***, *{Presence awareness of many other people,* ***is capable of causing****, feel-good factor}*}

The first claim belongs to this category, for the relation chosen to connect the concepts is strong (we argue that a claim {*Hypertext nodes*, ***is about***, *cinematic shots*} would have been weaker) and the concepts it uses are fine-grained. The second claim may not use a 'strong' relation, but it can still be considered as belonging to this category of annotations, for its author relates her interpretation to a previously submitted statement.

2.2.3 Discussion

While we have hypothesised that models in the upper right corner would be potentially the most desirable ones (providing both committed statements and a greater inter-connectedness of the claim space, and therefore more 'paths' to access a particular concept), we are not rejecting the other models 'flavours.' There are times when an annotator does not want to, or cannot, commit herself strongly and break her stream of ideas into committed and fine-grained statements (if a particular ***relation*** is missing, for instance.) We are however interested in identifying what can be done to assist annotators to move towards this upper-right corner when appropriate, by helping them formalise their interpretation.

The formalisation gulf

If interpreting a document is an activity that requires effort, interpreting a document with the ScholOnto framework is going to require even greater effort. Formalising an interpretation in ScholOnto - in other words, translating it into in a set of concepts and claims - adds this particular problem: the elicitation of what to use as *concepts*, how long (or detailed) should they be, and which ***relation*** should be used to connect them. The previous paragraphs have shown examples of concepts and claims and potential 'problems' with them.

Translating can also mean losing some aspects of an original interpretation, as it has to be expressed, and articulated, within the constraints of a rigid set of relations. The more fragmented the annotation becomes (moving towards the right side of the taxonomy), the more choices have to be made: more concepts have to be created and articulated with relations. These problems do not only exist in ScholOnto. They are likely to be faced by newcomers to any application requiring formalisation, as noted by Shipman and McCall:

> "*Users are hesitant about formalization because of a fear of prematurely committing to a specific perspective on their tasks; this may be especially true in a collaborative setting, where people must agree on an appropriate formalism.*" [Shipman and McCall, 1994]

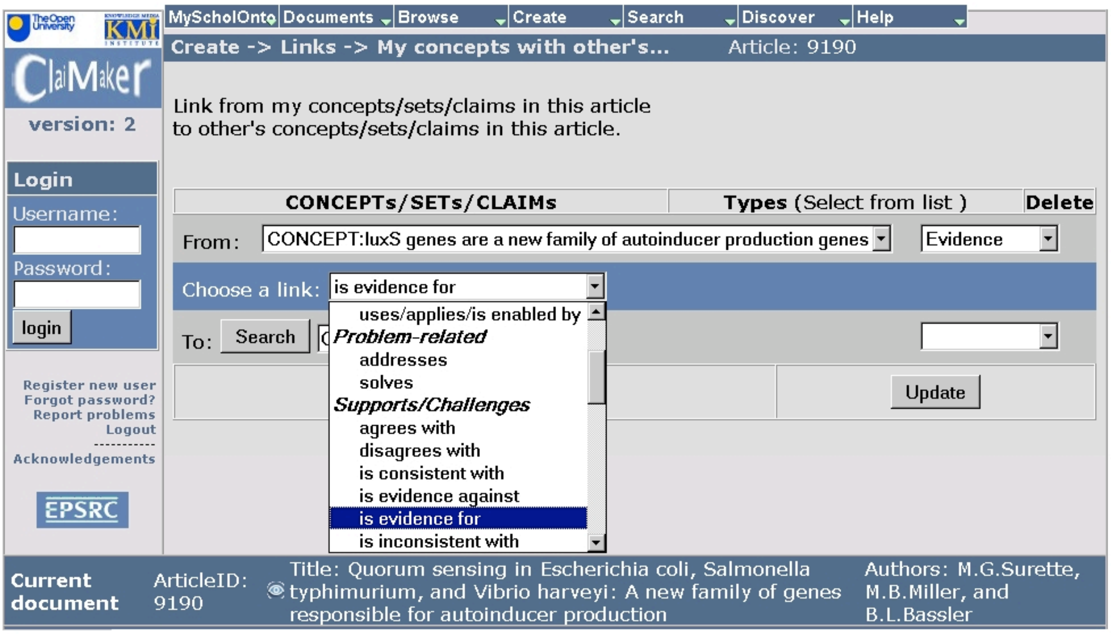


Figure 2.6: ClaiMaker is the original ScholOnto interface, aimed mostly at expert users. Annotation is done separately from reading, and users have to constantly switch their attention between a printed copy and the screen. The annotation process is also split into multiple steps: the annotator has to go through several menus (top) to decide on the kind of object she wants to create, to input it and to submit it.

Committing to a specific perspective in ScholOnto can be re-expressed as ‘deciding on a set of appropriate concepts and articulations between these concepts.’ This is the single most important difficulty of the framework. A formalisation gulf can emerge, between the richness offered to document authors to express their ideas, and the rigidity imposed by the fixed set of relations offered to annotators to build their networked structure of arguments. The main goal of our (and of any) annotation interface is therefore to assist the construction of an as strong as possible bridge over this gulf, to help annotators cross it and build their models.

2.3 Annotation interfaces

We now turn to the existing ScholOnto capture interfaces and to the support they provide to model concepts and claims. We have presented these interfaces in [Uren et al., 2003].

ClaiMaker

ClaiMaker was developed to initiate the population of the repository of annotations and to enable the project team members (experts with the formalism) to gain experience with the modelling task. ClaiMaker is a Web interface to the ScholOnto repository, in which concepts

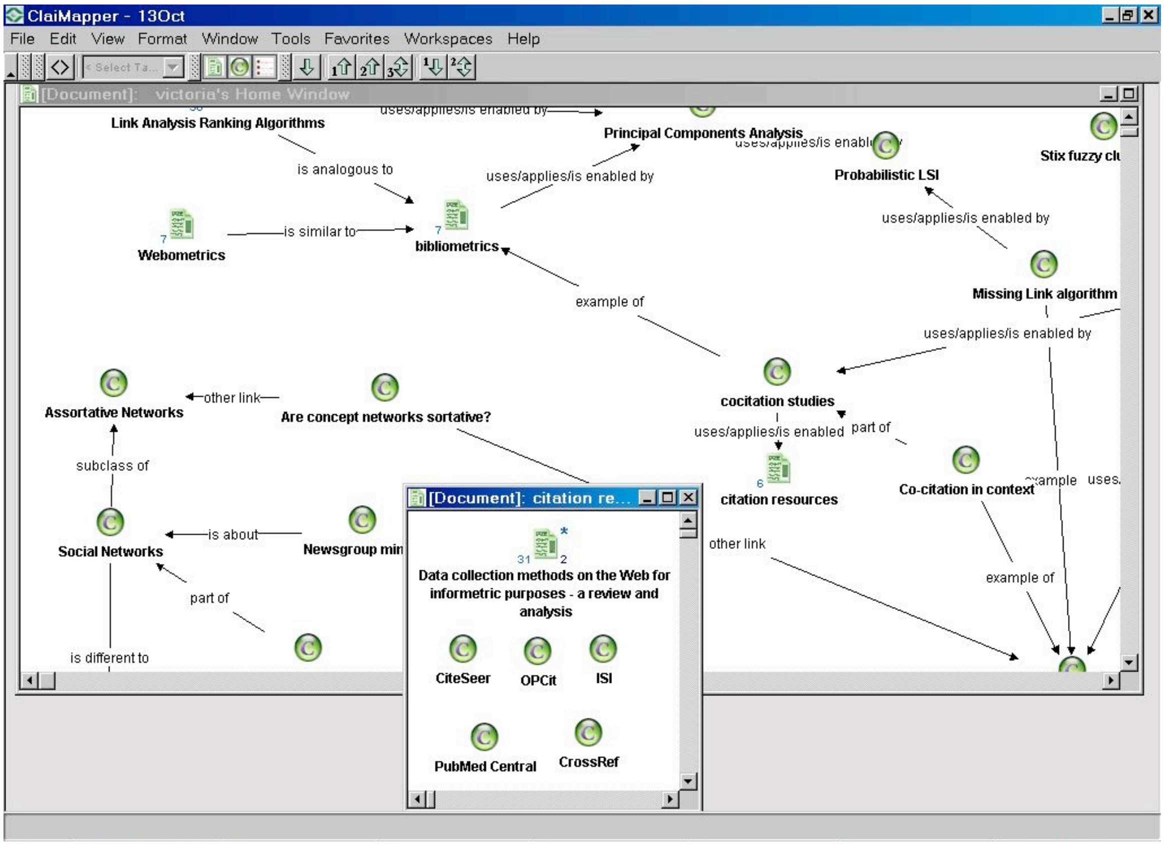


Figure 2.7: In ClaiMapper, the creation of concepts and their arrangements into claims is immediate as the interface, derived from Compendium, provides a graphical hypermedia system to 'draw' the thinking process. However, the original document is still not visible on-screen, resulting in a constant same attention-swapping process, as in ClaiMapper.

and claims are recorded via menus and forms. Figure 2.6 gives an example of a work session within ClaiMaker.

ClaiMaker suffers from several interaction breakdowns [6]. First, its interface implies a large amount of navigation in different menus and sub-menus. Since the main screen displays only one type of information at a time (the concept or the claim that is being worked on), it forces users to separate the annotation process into multiple steps and to keep in mind a clear idea of where they are going (in addition to the literature model they have to build.) In other words, it transforms the construction of a network (in which multiple branches can be built in parallel) into a long sequence of disconnected (from an interaction design, i.e. 'end-user dialogue' point of view) *create concept*, *create concept*, *connect concepts*. . . steps.

ClaiMapper

ClaiMapper improves this dialogue by mimicking the concept of a sketching pad, letting annotators draw nodes (concepts) on a map (current working window) and connect these nodes by drawing edges between them (thus creating claims between the source and destination nodes.) Figure 2.7 gives an example of a work session within ClaiMapper.

However, as with ClaiMaker, the connection between the document and the product of its annotation (concepts and claims) is lost, as the document is not displayed. The end-user artefact dialogue suffers from this disconnection.

ClaimFinder

An additional tool called ClaimFinder facilitates navigation in the claim space by displaying for any concept its neighbourhood. Figure 2.8 presents the results of a query on a text string: matched claims are shown in their neighborhood (i.e. the ones they are connected to and the ones connected to them.) ClaimFinder can be used conjointly with ClaiMaker or ClaiMapper. It adds one more interface to the annotation process however, increasing the cognitive load.

2.4 First research question

We propose an active document-centric semantic annotation environment to support the semantic annotation of a document, i.e. to help annotators create 'upper-right corner' models when possible and to (at least, partly) address the formalisation gulf. We want the document to be a central part of the annotation process in order to avoid the disconnectedness experienced with both ClaiMaker and ClaiMapper.

Spotting suggestions to support a two-steps annotation process

Our approach to support the annotation process relies upon the following question, the first of our two research questions (c.f. page 6):

(i) What sources of knowledge do we have, and which elements of the document can we find and suggest to annotators?

[6]An interaction breakdown occurs when 'something' forces the end-user to focus on an unwanted element of an application and to lose her train of thoughts. Such unwanted interruptions must be tracked to ensure the application is running as smoothly as possible.

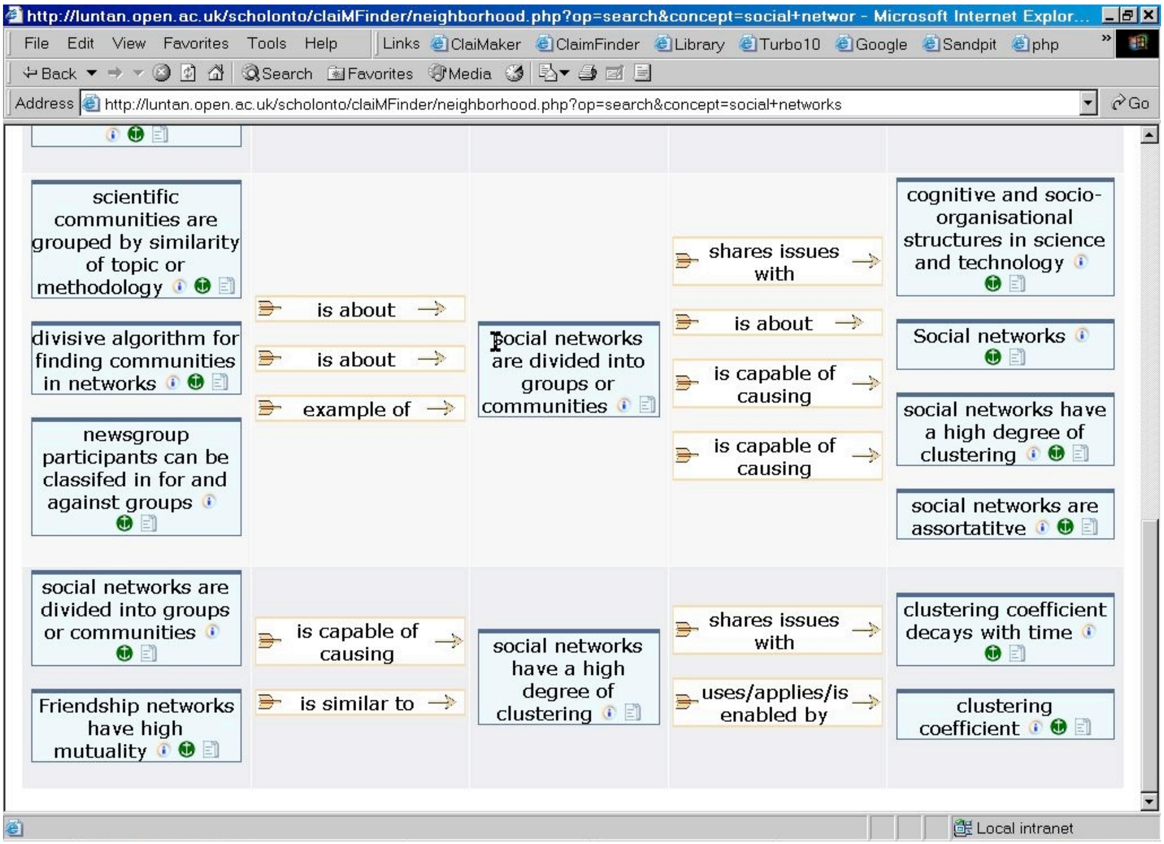


Figure 2.8: Queries can be specified in ClaimFinder to filter and browse a network of claims (in this example, a query '*social networks*' has been issued.)

We motivate an interaction design approach to study the dialogue between end-users and the annotation environment, in order to provide supporting information in a calm way. With our models taxonomy in mind, we can now define this calm way as a two steps annotation process.

In a first stage, we hope to facilitate an initial, 'simple' annotation of the document (including for instance, concepts and claims about its contributions, the problem it is tackling and the approach it has used) by proposing content-based support. This annotation could later be completed by a 'more complex' one, in which annotators expand their own claim space by taking position with their peers' annotations. The following chapter extends this preliminary analysis with a study of the tasks carried out by annotators to create their literature models.

Chapter 3

Task analysis

This chapter features our analysis of the tasks carried out in an annotation process. We report on the findings of a paper-based study, in which we investigate the behaviour of a group of annotators building a model of a document content and connections to the literature. This study gives us additional insight on the motivations and expectations of our end-users and leads us to identify an initial set of requirements for our environment.

3.1 Experimental protocol

Seven persons (two professors, one research fellow and four PhD students) are given a short article [1] and a marker. They are asked to answer a questionnaire (given in table 3.1) on a separate sheet of paper, formulating free-text answers. As the participants come from different communities, the unique article is chosen to appeal to all of them.

If they plan to make use of the text to formulate their answers, participants are explicitly asked to highlight, for each question, the parts they are going to use. Figure 3.1 gives an example of a highlighted paper after the experiment.

It can be noted that the task in this experiment is not to formulate ScholOnto concepts and claims, but rather to answer a set of open-ended questions. We decide not to include the formulation of concepts and claims at this stage, since we want to focus on the use annotators make of the document. Answers to the questionnaire can however be easily translated into concepts and claims: they already express committed statements (belonging to the upper half of the taxonomy, c.f. figure 2.5, page 21) about the argument of this document.

[1] Extracting and Visualizing Semantic Structures in Retrieval Results for Browsing. Katy Borner. In *Proceedings of the 5th ACM International Conference on Digital Libraries*. Association for Computing Machinery, 2000.

q1: what is the problem tackled in this document?
q2: how does the work presented try to address this problem?
q3: what previous work does it build on?
q4: what previous work does it critique?

Table 3.1: Questionnaire.

3.2 Evaluation

Appendix A (c.f. page 241) lists the components of the scholarly document that are used to answer each question. Annotators are displayed in the x axis, sentences in the y axis, and the couple (x, y) shows if the annotator x has used the component y, and if so, for which question $q\_$. Table 3.2 lists the answers to the questionnaire.

Our first observation is that every participant makes use of the document to answer their questions. This seems to confirm that reuniting the product of a sense-making process (the literature model) with the data on which it is built (the document) is a good starting assumption.

3.2.1 Repartition

The most obvious result, emphasised by the presentation of the results in table A.1 (c.f. page 246) is the distribution over the document of the elements used to answer the questions: most answers are found in the first half of the document. The abstract is very important in this experiment, enabling participants to answer (by polling all the answers) three of the four questions. Had we given only the abstract and the introduction to answer the questionnaire, instead of the full document, we would have had an answer to each question.

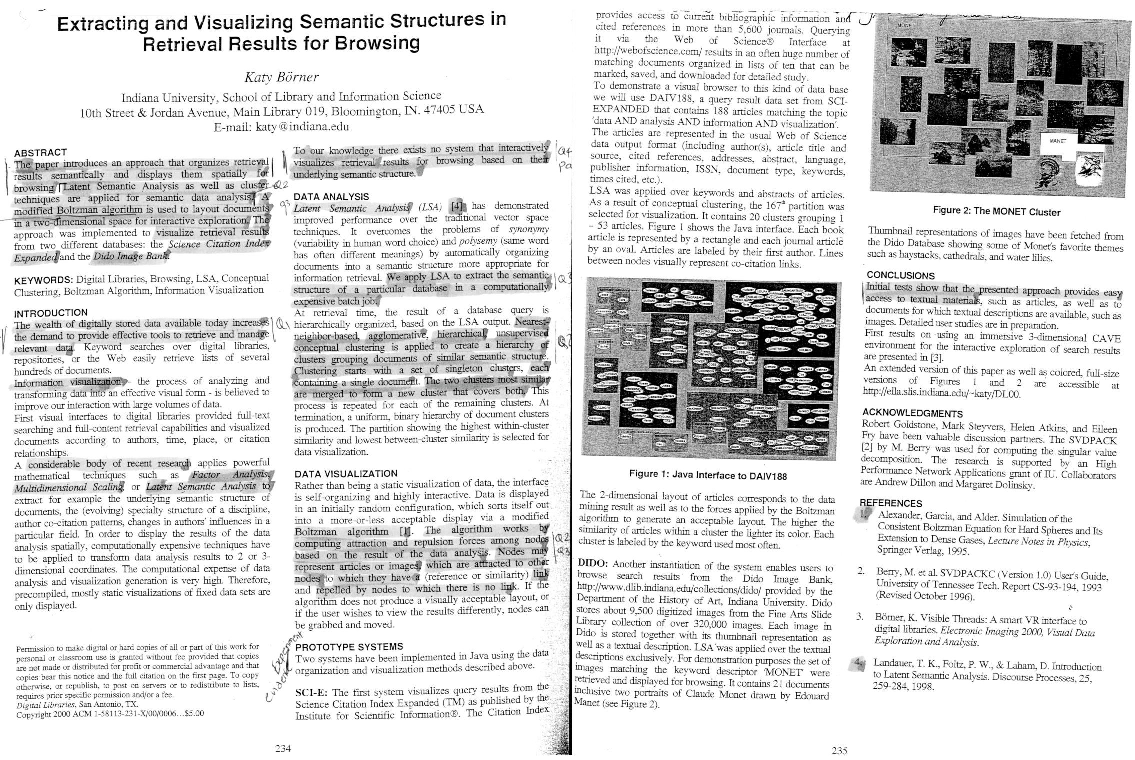
3.2.2 Nature of the article components used

Diving into further detail, an interesting aspect to notice is that the answer to each question is found in different locations. Different components of the original document (by components, we mean elements such as the title, headers, sections, paragraph and sentences) are also used. To answer the first question, participants use the title, but also a few sentences of both the abstract and the introduction. Question $q2$ is answered with sentences spread over three sections, in addition to the abstract. Similar phenomena are observed for the answers to questions $q3$ and $q4$.

Annotator $a6$ marks the whole sections 'Data analysis' and 'Data visualisation' as relevant to answer question $q3$, while annotator $a7$ also uses the title of the article. Nevertheless,

q1	What is the problem tackled in this document ?
a1	The huge amount of data digitally stored requires effective tools to retrieve and manage relevant data
a2	The vast amount of information available online causes hundreds of documents to be retrieved by a specific query (information overload); Although visualisation can facilitate browsing the results, the computational cost is usually high, which results usually in pre-compiled, mostly static, visualisations.
a3	It introduces an approach that organises retrieval results semantically and displays them spatially. The problem is tackled because digitally stored data available today is vast and we require a tool to retrieve and manage data.
a4	How to make large sets of results from queries to digital libraries comprehensible to the user
a5	Interactive, dynamic visualisation o large data based on their semantic structure; going beyond pre-compiled visualisations; provide effective tools to retrieve and manage relevant data
a6	How to visualise the semantic structure of retrieved documents
a7	How to provide effective tools to retrieve and manage data; How to provide interactive visualisation of search results based on underlying semantic structure
q2	How does the work presented try to address this problem ?
a1	Results are semantically retrieved and displayed spatially for browsing. LSA and cluster techniques are used for semantic data analysis. A modified Boltzman algorithm is used to layout documents in a 2D space.
a2	In order to solve the problem, the authors propose the use of semantic visualisation that facilitates spatial browsing
a3	It uses latent semantic analysis [LSA] (4); and cluster techniques are used for grouping documents of similar semantic structure. It also uses Boltzman algorithm [1] for data visualisation. It is used to layout documents from 2-dimensional space to facilitate interactive exploration
a4	Information visualisation: displaying the results spatially to help the user interact with it; it combines 2 methods; latent semantic analysis and a modified Boltzman algorithm to extract semantic structure from the results, and then arrange them in space
a5	Using LSA and cluster techniques and Boltzman
a6	Documents are clustered using LSA, then laid out vertically using a modified Boltzman algorithm - nodes ‘attract’ and ‘repel’ dynamically and self-group on the display
a7	Use LSA to extract semantic structure; modify Boltzman algorithm and cluster nodes
q3	What previous work does it build on ?
a1	LSA, Boltzman, SCI-E, DIDO
a2	They use LSA to process the retrieved results. Nearest neighbour clustering is also used. Boltzman algorithm is used to spatially organise the results.
a3	It subscribes to LSA and Boltzman algorithm
a4	Latent Semantic analysis for extracting the structure. Boltzman algorithm for laying out.
a5	Latent Semantic Analysis. Boltzman algorithm. Search methods. Clustering techniques
a6	LSA. Boltzman.
a7	Modify Boltzman algorithm. Java. LSA
q4	What previous work does it critique ?
a1	Vector space technique: poor performance. All previous work display data statically.
a2	They argue that existing mathematical techniques are computationally expensive.
a3	They critique on systems saying ‘no system interactive visualises retrieval results for browsing based on their semantic structure’.
a4	They do not *explicitly* critique any work but they identify a problem common to many current visualisations: the algorithms used are so computationally expensive that only static displays can be produced
a5	None
a6	Earlier attempts are implicitly criticised as being ‘static’ and not interactive, not using the underlying semantic structure. Though this is a weak and unsubstantiated critique
a7	Approaches that are computationally expensive and can only generate static representations

Table 3.2: Answers to the questionnaire.



Extracting and Visualizing Semantic Structures in Retrieval Results for Browsing

Katy Börner

Indiana University, School of Library and Information Science
10th Street & Jordan Avenue, Main Library 019, Bloomington, IN. 47405 USA
E-mail: katy@indiana.edu

ABSTRACT
The paper introduces an approach that organizes retrieval results semantically and displays them spatially for browsing. Latent Semantic Analysis as well as cluster techniques are applied for semantic data analysis. A modified Boltzman algorithm is used to layout documents in a two-dimensional space for interactive exploration. The approach was implemented to visualize retrieval results from two different databases: the *Science Citation Index Expanded* and the *Dido Image Bank*.

KEYWORDS: Digital Libraries, Browsing, LSA, Conceptual Clustering, Boltzman Algorithm, Information Visualization

INTRODUCTION
The wealth of digitally stored data available today increases the demand to provide effective tools to retrieve and manage relevant data. Keyword searches over digital libraries, repositories, or the Web easily retrieve lists of several hundreds of documents.
Information visualization - the process of analyzing and transforming data into an effective visual form - is believed to improve our interaction with large volumes of data.
First visual interfaces to digital libraries provided full-text searching and full-content retrieval capabilities and visualized documents according to authors, time, place, or citation relationships.
A considerable body of recent research applies powerful mathematical techniques such as *Factor Analysis*, *Multidimensional Scaling*, or *Latent Semantic Analysis* to extract for example the underlying semantic structure of documents, the (evolving) specialty structure of a discipline, author co-citation patterns, changes in authors' influences in a particular field. In order to display the results of the data analysis spatially, computationally expensive techniques have to be applied to transform data analysis results to 2 or 3-dimensional coordinates. The computational expense of data analysis and visualization generation is very high. Therefore, precompiled, mostly static visualizations of fixed data sets are only displayed.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
Digital Libraries, San Antonio, TX.
Copyright 2000 ACM 1-58113-231-X/00/0006...$5.00

To our knowledge there exists no system that interactively visualizes retrieval results for browsing based on their underlying semantic structure.

DATA ANALYSIS
Latent Semantic Analysis (LSA) [4] has demonstrated improved performance over the traditional vector space techniques. It overcomes the problems of *synonymy* (variability in human word choice) and *polysemy* (same word has often different meanings) by automatically organizing documents into a semantic structure more appropriate for information retrieval. We apply LSA to extract the semantic structure of a particular database in a computationally expensive batch job.
At retrieval time, the result of a database query is hierarchically organized, based on the LSA output. Nearest-neighbor-based, agglomerative, hierarchical, unsupervised conceptual clustering is applied to create a hierarchy of clusters grouping documents of similar semantic structure. Clustering starts with a set of singleton clusters, each containing a single document. The two clusters most similar are merged to form a new cluster that covers both. This process is repeated for each of the remaining clusters. At termination, a uniform, binary hierarchy of document clusters is produced. The partition showing the highest within-cluster similarity and lowest between-cluster similarity is selected for data visualization.

DATA VISUALIZATION
Rather than being a static visualization of data, the interface is self-organizing and highly interactive. Data is displayed in an initially random configuration, which sorts itself out into a more-or-less acceptable display via a modified Boltzman algorithm [1]. The algorithm works by computing attraction and repulsion forces among nodes based on the result of the data analysis. Nodes may represent articles or images, which are attracted to other nodes to which they have a (reference or similarity) link and repelled by nodes to which there is no link. If the algorithm does not produce a visually acceptable layout, or if the user wishes to view the results differently, nodes can be grabbed and moved.

PROTOTYPE SYSTEMS
Two systems have been implemented in Java using the data organization and visualization methods described above.

SCI-E: The first system visualizes query results from the Science Citation Index Expanded (TM) as published by the Institute for Scientific Information®. The Citation Index

234

provides access to current bibliographic information and cited references in more than 5,600 journals. Querying it via the Web of Science® Interface at http://webofscience.com/ results in an often huge number of matching documents organized in lists of ten that can be marked, saved, and downloaded for detailed study.
To demonstrate a visual browser to this kind of data base we will use DAIV188, a query result data set from SCI-EXPANDED that contains 188 articles matching the topic 'data AND analysis AND information AND visualization'.
The articles are represented in the usual Web of Science data output format (including author(s), article title and source, cited references, addresses, abstract, language, publisher information, ISSN, document type, keywords, times cited, etc.).
LSA was applied over keywords and abstracts of articles. As a result of conceptual clustering, the 167th partition was selected for visualization. It contains 20 clusters grouping 1 - 53 articles. Figure 1 shows the Java interface. Each book article is represented by a rectangle and each journal article by an oval. Articles are labeled by their first author. Lines between nodes visually represent co-citation links.

Figure 1: Java Interface to DAIV188

The 2-dimensional layout of articles corresponds to the data mining result as well as to the forces applied by the Boltzman algorithm to generate an acceptable layout. The higher the similarity of articles within a cluster the lighter its color. Each cluster is labeled by the keyword used most often.

DIDO: Another instantiation of the system enables users to browse search results from the Dido Image Bank, http://www.dlib.indiana.edu/collections/dido/ provided by the Department of the History of Art, Indiana University. Dido stores about 9,500 digitized images from the Fine Arts Slide Library collection of over 320,000 images. Each image in Dido is stored together with its thumbnail representation as well as a textual description. LSA was applied over the textual descriptions exclusively. For demonstration purposes the set of images matching the keyword descriptor 'MONET' were retrieved and displayed for browsing. It contains 21 documents inclusive two portraits of Claude Monet drawn by Edouard Manet (see Figure 2).

Figure 2: The MONET Cluster

Thumbnail representations of images have been fetched from the Dido Database showing some of Monet's favorite themes such as haystacks, cathedrals, and water lilies.

CONCLUSIONS
Initial tests show that the presented approach provides easy access to textual materials, such as articles, as well as to documents for which textual descriptions are available, such as images. Detailed user studies are in preparation.
First results on using an immersive 3-dimensional CAVE environment for the interactive exploration of search results are presented in [3].
An extended version of this paper as well as colored, full-size versions of Figures 1 and 2 are accessible at http://ella.slis.indiana.edu/~katy/DL00.

ACKNOWLEDGMENTS
Robert Goldstone, Mark Steyvers, Helen Atkins, and Eileen Fry have been valuable discussion partners. The SVDPACK [2] by M. Berry was used for computing the singular value decomposition. The research is supported by an High Performance Network Applications grant of IU. Collaborators are Andrew Dillon and Margaret Dolinsky.

REFERENCES
1. Alexander, Garcia, and Alder. Simulation of the Consistent Boltzman Equation for Hard Spheres and Its Extension to Dense Gases, *Lecture Notes in Physics*, Springer Verlag, 1995.
2. Berry, M. et al. SVDPACKC (Version 1.0) User's Guide, University of Tennessee Tech. Report CS-93-194, 1993 (Revised October 1996).
3. Börner, K. Visible Threads: A smart VR interface to digital libraries. *Electronic Imaging 2000, Visual Data Exploration and Analysis*.
4. Landauer, T. K., Foltz, P. W., & Laham, D. Introduction to Latent Semantic Analysis. Discourse Processes, 25, 259-284, 1998.

235

Figure 3.1: Areas used to answer each question. The $a1$, $a2$, $a3$ and $a4$ marks indicate the components used to answer questions $q1$, $q2$, $q3$ and $q4$, respectively.

sentences are by far the most commonly used component.

3.2.3 Use of article components

While different components can be used to answer a particular question, we also notice that a given component can be used to answer different questions (confirming Bishop's findings [Bishop, 1998].) This seems to reflect the differences one can see in what makes the contribution of a paper and - to relate this to our literature-modelling task - the differences one can find in the content of these annotations.

Sentences can be used to answer different questions

Sentences #1, #9 and #25 are used to answer no less than three different questions (c.f. table 3.3; full results are given in table A.1, page 246):

S-#	Document component	a1	a2	a3	a4	a5	a6	a7
1	The paper introduces an approach that organizes retrieval results semantically and displays them spatially for browsing.	q2	q2	q1	q2	q2 q3	q1	

S-#	Document component	a1	a2	a3	a4	a5	a6	a7
9	A considerable body of recent research applies powerful mathematical techniques such as Factor Analysis, Multidimensional Scaling, or Latent Semantic Analysis to extract for example the underlying semantic structure of documents, the (evolving) specialty structure of a discipline, author co-citation patterns, changes in authors' influences in a particular field.	q3	q1 q4		q4			
25	Data is displayed in an initially random configuration, which sorts itself out into a more-or-less acceptable display via a modified Boltzman algorithm [1].		q3	q2 q3	q3	q3	q2	q2 q4

Table 3.3: Results from the paper-based experiment | Extract. (1/4)

This phenomenon also highlights how difficult it can be to automatically identify the areas of a document where the contribution or the work it builds on are stated.

The fact that similar sentences (sentences #3, #4 and #25) are used to answer questions $q2$ and $q3$ is also interesting (c.f. table 3.4.) A possible explanation may be that these sentences play two roles: for instance, sentence #25 both explains the approach proposed and makes a connection to previous work (strongly indicated by its citation signal.)

S-#	Document component	a1	a2	a3	a4	a5	a6	a7
3	A modified Boltzman algorithm is used to layout documents in a two-dimensional space for interactive exploration.	q2	q3	q3		q2 q3		
4	The approach was implemented to visualize retrieval results from two different databases: the Science Citation Index Expanded and the Dido Image Bank.	q3						

S-#	Document component	a1	a2	a3	a4	a5	a6	a7
25	Data is displayed in an initially random configuration, which sorts itself out into a more-or-less acceptable display via a modified Boltzman algorithm [1].		q3	q2 q3	q3	q3	q2	q2 q4

Table 3.4: Results from the paper-based experiment | Extract. (2/4)

Sentences can be used consistently

As a counter-example, some sentences are used very consistently among the participants. Sentence #5 is used four times out of four to answer question $q1$ (c.f. table 3.5.)

S-#	Document component	a1	a2	a3	a4	a5	a6	a7
5	The wealth of digitally stored data available today increases the demand to provide effective tools to retrieve and manage relevant data.	q1		q1	q1	q1		

Table 3.5: Results from the paper-based experiment | Extract. (3/4)

While it is difficult to assess why a particular sentence is picked, a surface-based analysis may give us some cues of its relevance. For instance, this sentence is the first one of the first section of the body of the document and, as such, is likely to set the context of the work being reported. Additional cues can be found in its constituents: an expression as '*increases the demand to*' indicates that a phenomenon ('*the wealth of digitally stored data available today*') is creating a need. Such cues may be worth keeping in mind to identify other relevant sentences.

Scientific attribution

Scientific attribution is also involved in the consideration of one sentence instead of another to answer a question [2]: work can be attributed to the author herself (if she mentions work that she has previously accomplished), to a community in general (for background statements which are accepted by the community she belongs to) or to a particular researcher (via citations for instance.)

[2] It is presented in more detail in our review of the argumentative zoning approach by Teufel and Moens, c.f. page 86.

Statements can be attributed with language expressions readers can recognise. For instance, sentence #25, attributing (in the end of the sentence) a work to a particular author (using a citation signal), is used to answer questions $q3$ and $q4$. In this case, the citation signal is a very good indicator. However, it is not always needed to attribute a contribution to another researcher or to a community. Sentence #3 also mentions work that belongs to another researcher, but does not feature a citation; readers still recognise it consistently and use it to answer question $q3$. Sentence #2 also refers to work being carried out by an external author, and readers similarly recognise this attribution and use it to answer question $q2$ (c.f. table 3.6.)

S-#	Document component	a1	a2	a3	a4	a5	a6	a7
2	Latent Semantic Analysis as well as cluster techniques are applied for semantic data analysis.	q2	q3	q2		q2 q3		q1

Table 3.6: Results from the paper-based experiment | Extract. (4/4)

There is not, as can be seen from the repartition of the components picked, a direct correlation between the scientific attribution of a sentence and its selection to answer a particular question. Annotators do not (and will not be asked to) build a complete map of the scientific attribution of each sentence, but only pick the bits they need to answer their questions (later, to build their ScholOnto model.) This means that some sentences that would be highly useful may not be picked (if an annotator feels she has already answered a question.) It also means that sentences exhibiting no *à priori* strong characteristic may be picked, for a reason we cannot predict.

Common elements

Answers to the questionnaire (c.f. table 3.2), although different, share some aspects. For instance, elements such as 'retrieval' and 'querying' are frequently found in answers to $q1$, while nouns or noun groups like 'LSA' and 'Boltzman algorithm' are common in answers to $q3$. Extracting and suggesting these two notions could be helpful, letting annotators decide whether they are relevant or not. These elements (an acronym and a noun group (based on a proper noun)) are also available in the keywords of the document, and this may have influenced annotators (although it seems like nobody makes use of them, c.f. table A.1, page 246.) Keywords, proper nouns and noun groups are potential sources of relevant information to answer questions about the argument of a document, and therefore, to create ScholOnto

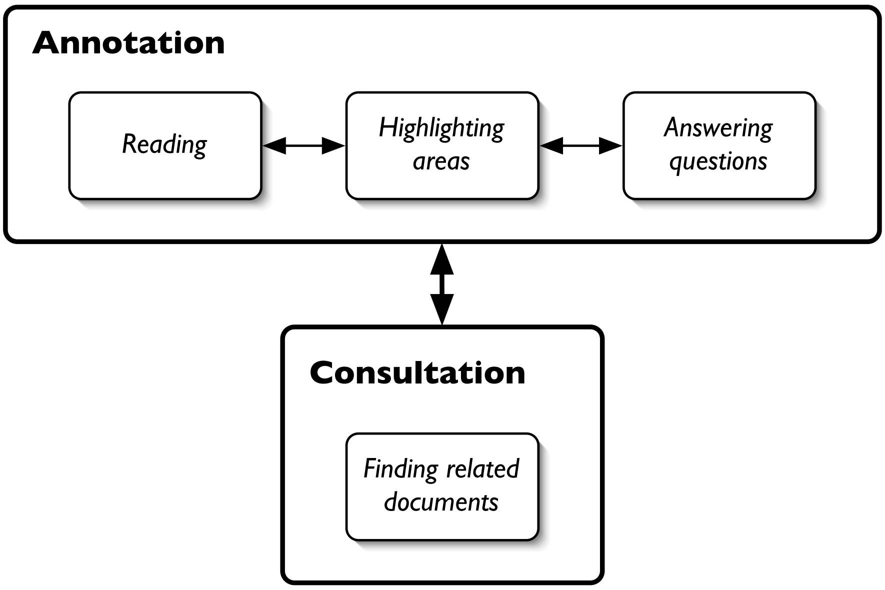


Figure 3.2: The tasks carried out by the participants of the paper-based annotation experiment can be organised in two main activities.

concepts.

3.3 Current analysis

We present in this section the conclusion we have drawn from this task analysis.

3.3.1 Tasks and activities

Several tasks are performed in a literature modelling process:

- reading the document
- identifying relevant areas of the document
- answering questions about the content and connections of the document

An additional task can be added:

- finding related documents. While this is not properly identified as a task in the course of this experiment, it can nevertheless be reasonably inferred from the fact that a few annotators highlight the 'References' area (c.f. figure 3.1 for an example.) It can also be inferred that had these papers been available, annotators would have consulted them to get further information.)

Figure 3.2 organises these tasks into an activity diagram. The first three tasks can be gathered in an 'annotation' activity, and the last one in a 'consultation' activity. We expect these two activities to be carried out in parallel: the annotation of a document (i.e. the identification of concepts and claims) may trigger a need to consult additional documents, while the consultation of external documents may yield a new perspective (i.e. new concepts and claims) on the document being annotated.

3.3.2 'Identifying relevant areas'

This task is the one this dissertation focusses on. The behaviours we have witnessed in this preliminary experiment give us reasonable hope that some kind of document-based support spotting and highlighting potentially relevant snippets from the text and its presentation to annotators consideration are 'compatible' with their natural approach to this process.

3.3.3 Initial set of requirements

We conclude this chapter with two observations that we turn into an initial set of requirements for our environment.

Claims-worthy snippets exist in a document…

Certain snippets are taken into account by annotators to formulate their answers. While we have seen how difficult it is to assess whether a given snippet is of interest or not (since any piece of information may be of interest to at least one annotator, we cannot guess what they want to record), a supporting environment should provide several strategies (e.g. a 'toolbox' of strategies) to highlight different elements, letting annotators decide which one among these works best for them.

but should be only suggested

The variety observed in the answers also confirms that annotations are potentially different for each annotator. We deduct from this observation that this content-based support must be provided at a suggestion level only. We have seen how differing components can be used to answer a single question and how, conversely, a single element can be used to answer different questions. Therefore, the role of these suggested snippets must not be to provide *the* definitive answer in the sense of the definitive passages to model as concepts and claims. Their role must instead be to provide pointers to *potentially* relevant areas, leaving the final decision to consider them (or not) to each annotator, according to her own research interests.

In the following chapter, we advance our design phase to include a deeper analysis of the sense-making process and of its tasks, and a literature review to identify and assess approaches to support it efficiently. We finally motivate our approach and identify a final set of requirements.

Chapter 4

Sense-making analysis

In this chapter, we reframe a ScholOnto annotation process as a sense-making process. This framework enables us to present this process as a set of tasks (some of which have been identified in the previous chapter) and to classify and review existing approaches to support each of them. Based on this framework, we motivate our approach and present a final set of requirements for our active document-centric annotation environment.

4.1 Annotation as a sense-making process

To move from a scholarly document to a set of concepts and claims (c.f. figure 4.1), users assess the significance of its text by answering a question that can be formulated as '*What makes this particular article relevant to my work?*' Addressing this problem requires analysing and making sense of the document in order to create a claim space, that can be assimilated to:

> "*a sense-making forum in which the objects of discussion are visual representations of their understanding of a situation, a problem, or an objective... an openly reflexive forum in which communities of knowing explicitly talk about their understandings.*" [Boland and Tenkasi, 1995]

Sense-making can be composed of the following activities: (1) information gathering, (2) information re-representation, (3) insight from representation and manipulation (of this knowledge), resulting in (4) a knowledge product or action [1]. Figure 4.2 maps these four activities to the tasks and activities we have identified in the previous chapter.

It can be noted that the transition between the two is fairly easy. The 'annotation' activity can be mapped to the 're-representation' (and, arguably, the 'knowledge product or action')

[1] Stuart Card keynote address at IUI2005, San Diego, CA, USA

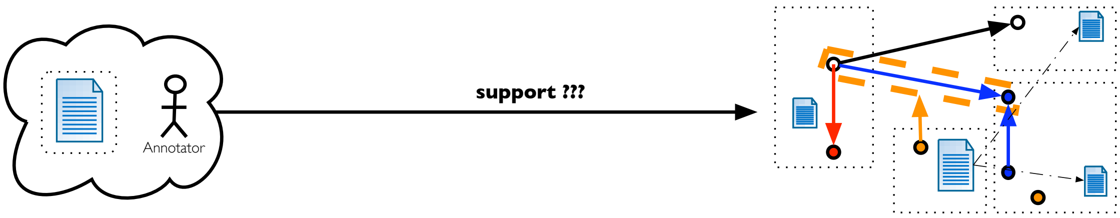


Figure 4.1: We aim at supporting the transition from annotators' interpretation of a scholarly document to a network of interconnected concepts and claims.

step of Stuart Card's taxonomy, while our 'consultation' activity can be mapped to the 'information gathering' step. In terms of tasks, it can be noted that we have added two tasks to our diagram: 'scaffolding' and 'reusing peers' annotations.'

The 'scaffolding' task is actually not an addition, as it is present in our initial paper-based study. The questionnaire provides scaffolds to guide the annotation. We have decided to make it explicit in our tasks list. The second task added, 'reusing peers' annotations' reflects the participatory dimension of a ScholOnto annotation process. Although ScholOnto notes can be kept private in theory, their availability opens up several possibilities in terms of debate and discussion (and admittedly several difficulties too; we will come back to these later in this dissertation, in chapter 7.)

We have organised our literature review along these activities. We review approaches to scaffold annotation (c.f. section 4.3, page 52), approaches to support reading (c.f. section 4.4, page 59), approaches to support the extraction of concepts and claims (c.f. sections 4.5, page 66 and 4.6, page 75), approaches to support the discovery of related documents (c.f. section 4.7, page 90) and finally approaches to support participatory argumentation (c.f. section 4.8, page 94.)

4.2 Sense-making mediums

We begin our literature review with a presentation of the different mediums in which sense-making can be achieved, that we have organised by their increasing level of formality (moving on an axis similar to the granularity axis of our annotation taxonomy, c.f. section 2.2.2, page 21.)

Documents are obviously a central part of a sense-making process: they become contextualised with the ideas we retain and reformulate from their contents. The 'simplest' approach to reformulate (or re-represent, to reuse one of Stuart Card's steps) knowledge is to annotate a document [Ovsiannikov et al., 1999].

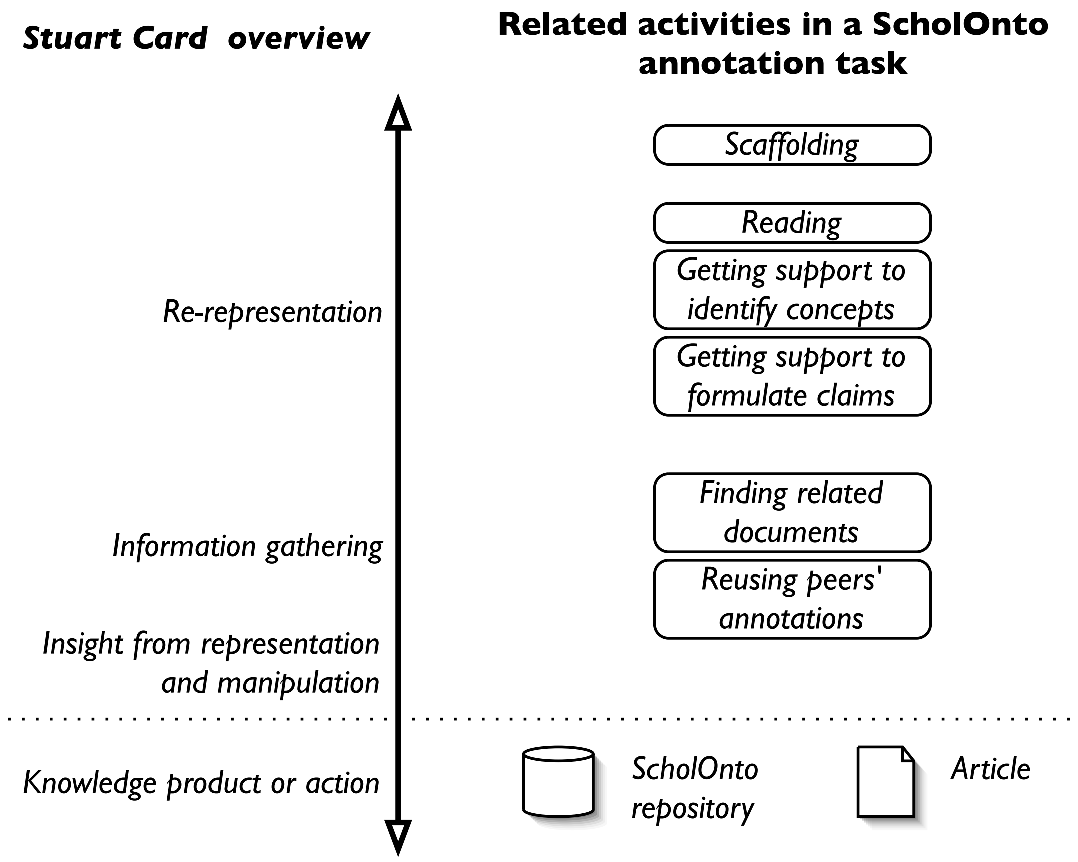


Figure 4.2: Annotating a document in ScholOnto can be assimilated to a sense-making process.

4.2.1 Document annotation

Notes can be written to clarify and structure the information we process, and to '*make thinking visible*' [Bell, 1997]. In a study of physical annotations (written on textbooks) [Marshall, 1997], Catherine Marshall enumerates the roles played by such notes: procedural signalling for future attention (to designate the areas of a text book which can be 'safely' dismissed for the exam), place-markings and aiding memory (to designate areas to remember), problem-working locations (where a particular problem (e.g. an equation) is worked on (with additional figures or recopied theorems) at the place it is stated in the textbook), records of interpretative activity (to write down interpretations of the content of the textbook), traces of the reader's attention (if an aspect of the work is particularly difficult to grasp) and incidental reflections of the material circumstances (c.f. figure 4.3 for an example of notes used to 'transform text into knowledge' [O'Leary, 1998].)

In her conclusion, she lists several annotation facilities that should be developed to let readers feel as confident as possible with annotation in a computerised context. These include (1) in-situ annotations, distinguishable from the original document; (2) non-interpretative marking resulting from the difficulties met in understanding a text; (3) a fluidity in the different forms (symbols, colour, pen types) used to annotate; (4) informal codings to let readers structure as much or as little as they want their annotations; (5) the possibility to consult annotated versions of the documents, and (6) finally an integration with reading as an activity.

Current annotation applications support, partly or wholly, these principles. Annotea [Kahan et al., 2001] (c.f. figure 4.4), YAWAS [Denoue and Vignollet, 2000], CritLink [Yee, 2002], Haystack [Karger et al., 2003] and a system presented in [Brush et al., 2002] are introduced later in this chapter. These tools usually offer possibilities to create, characterise, share and retrieve notes.

Our own ScholOnto notes (i.e. sets of concepts and claims) share a similar freedom in their content, but they can also be reused and interconnected. This, however, puts a greater burden on the user, as she has to elicit what to use as concepts, and which relations to use to connect them.

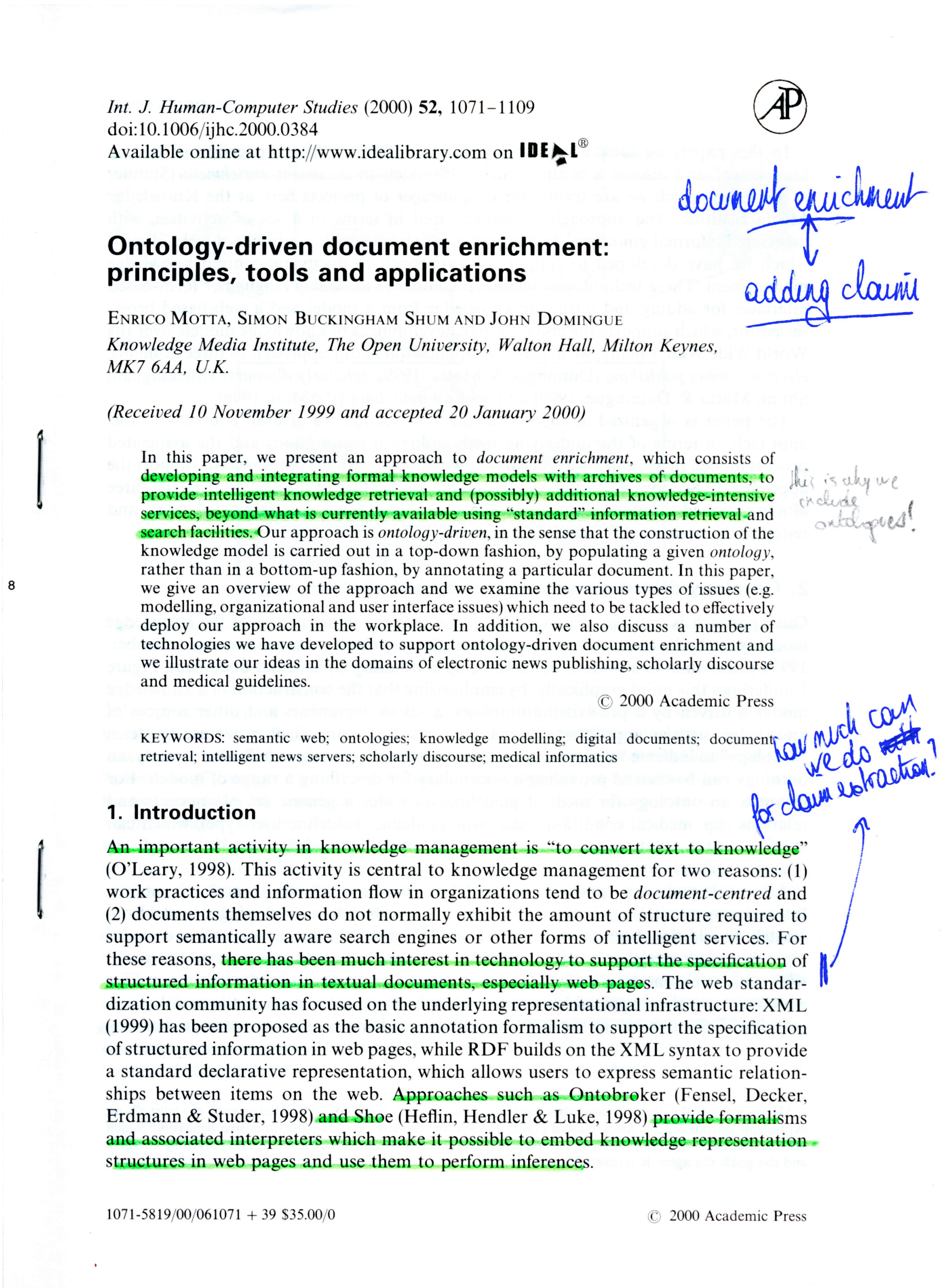
4.2.2 Discussion spaces

Making sense of a document can also be supported in a discussion space. The focus is obviously more on the idea of dialogue between the annotators. A discussion space offers an area where participants can discuss to answer a particular, well-defined, question, such as '*Is this document relevant?*' They can express their arguments and point of views in free-text form. They can also categorise them further via the addition of a type such as 'agreement' or 'disagreement.'

SpeakEasy provides an asynchronous discussion environment, in which students (eighth graders) can explain their ideas in response to a challenge or a problem related to a particular science topic [Hoadley and Linn, 2000]. Their peers can comment on these ideas, add their own information, thus contributing to the learning of a complex phenomenon. D3E is a publishing framework dedicated to research papers [Sumner and Buckingham Shum, 1998]. It has been applied to discuss and debate the contents of scholarly articles submitted to an online journal, JIME [2]. Figure 4.5 gives an example of an article under review.

ScholOnto annotations are similarly focussed on a particular, well-defined question such as '*What makes this particular paper relevant to my work?*' This question can be refined into more explicit ones such as '*What is the problem tackled in this paper?*' or '*What approach is proposed to address this problem?*' Compared to D3E utterances however, ScholOnto annotations are interconnected and potentially reusable.

[2] The Journal of Interactive Media in Education is accessible at `http://jime.open.ac.uk/`



Int. J. Human-Computer Studies (2000) **52**, 1071–1109
doi:10.1006/ijhc.2000.0384
Available online at http://www.idealibrary.com on IDEAL®

Ontology-driven document enrichment: principles, tools and applications

Enrico Motta, Simon Buckingham Shum and John Domingue
Knowledge Media Institute, The Open University, Walton Hall, Milton Keynes, MK7 6AA, U.K.

(Received 10 November 1999 and accepted 20 January 2000)

In this paper, we present an approach to *document enrichment*, which consists of developing and integrating formal knowledge models with archives of documents, to provide intelligent knowledge retrieval and (possibly) additional knowledge-intensive services, beyond what is currently available using "standard" information retrieval and search facilities. Our approach is *ontology-driven*, in the sense that the construction of the knowledge model is carried out in a top-down fashion, by populating a given *ontology*, rather than in a bottom-up fashion, by annotating a particular document. In this paper, we give an overview of the approach and we examine the various types of issues (e.g. modelling, organizational and user interface issues) which need to be tackled to effectively deploy our approach in the workplace. In addition, we also discuss a number of technologies we have developed to support ontology-driven document enrichment and we illustrate our ideas in the domains of electronic news publishing, scholarly discourse and medical guidelines.

© 2000 Academic Press

KEYWORDS: semantic web; ontologies; knowledge modelling; digital documents; document retrieval; intelligent news servers; scholarly discourse; medical informatics

1. Introduction

An important activity in knowledge management is "to convert text to knowledge" (O'Leary, 1998). This activity is central to knowledge management for two reasons: (1) work practices and information flow in organizations tend to be *document-centred* and (2) documents themselves do not normally exhibit the amount of structure required to support semantically aware search engines or other forms of intelligent services. For these reasons, there has been much interest in technology to support the specification of structured information in textual documents, especially web pages. The web standardization community has focused on the underlying representational infrastructure: XML (1999) has been proposed as the basic annotation formalism to support the specification of structured information in web pages, while RDF builds on the XML syntax to provide a standard declarative representation, which allows users to express semantic relationships between items on the web. Approaches such as Ontobroker (Fensel, Decker, Erdmann & Studer, 1998) and Shoe (Heflin, Hendler & Luke, 1998) provide formalisms and associated interpreters which make it possible to embed knowledge representation structures in web pages and use them to perform inferences.

1071-5819/00/061071 + 39 $35.00/0 © 2000 Academic Press

Figure 4.3: The notes added to a document participate to a sense-making process in which the text is transformed and contextualised for the reader's goals. Procedural signals can be used to indicate relevant passages [Marshall, 1997].

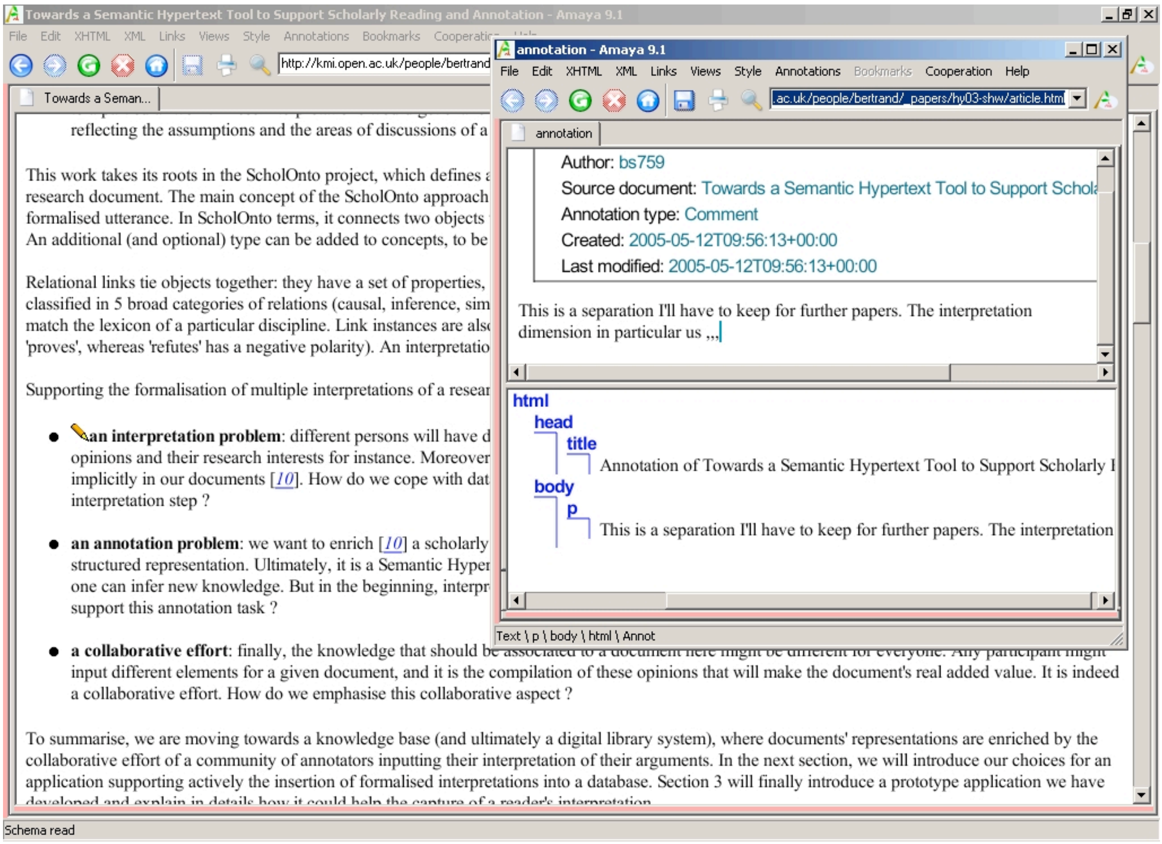


Figure 4.4: Annotations can be added to any location of a document in Amaya, a prototype Web browser released from the W3C that implements the Annotea annotation protocol.

4.2.3 Structured representations

The final medium that we consider relies on structured representations.

Trails of notes

The use of a structured representation to support sense-making dates back from Bush's Memex, a vision of a system with which we can create, display and rearrange at a later stage trails of ideas [Bush, 1945]:

> "*Consider a future device for individual use, which is a sort of mechanized private file and library. It needs a name, and, to coin one at random, 'memex' will do. A memex is a device in which an individual stores all his books, records and communications, and which is mechanized so that it may be consulted with exceeding speed and flexibility. It is an enlarged intimate supplement to his memory.*"

ScholOnto is inspired by this vision, enabling annotators to create 'trails' of concepts, connected with the relations defined in the ontology. By creating a structured representation

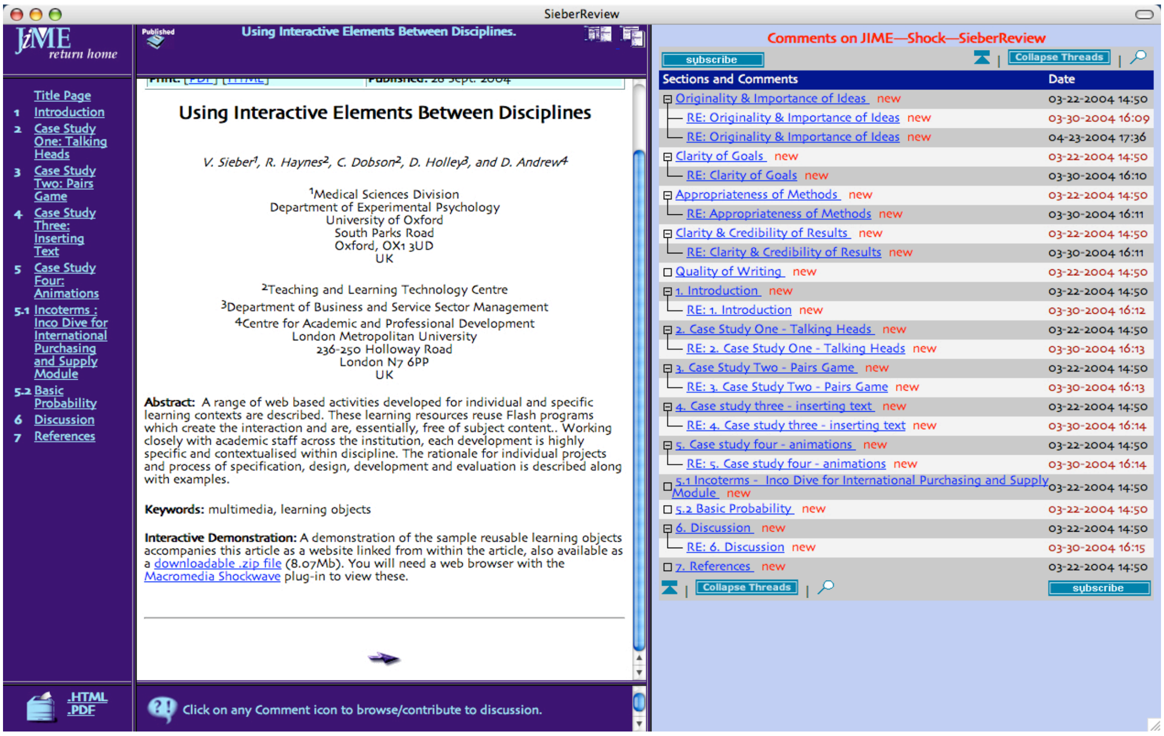


Figure 4.5: JIME relies on D3E to provide a space where authors and reviewers can discuss to answer whether the reviewed paper is a contribution. The content of the document can be explored via the links generated in the left frame.

of interpretations of a scholarly document, it becomes possible to reuse nodes at a later stage and thus to build additional connections between documents.

Structures of notes

An extension of this trail vision is found in Engelbart 'framework for augmenting human intellect' [Engelbart, 1962]. Pre-visioning hypertext systems, he proposes an approach to represent the steps of a reasoning process (via statements such as 'collecting evidence', 'taking positions' and 'motivating them') not only as a serial process as in the 'trail of ideas' vision, but also as a sequential process, i.e. as a process in which steps do not have to be organised one after the other, but can be arranged in a conceptual network. Possibilities to create sub-structures of ideas, to link them, to look for some particular kinds of links arriving at one statement, to emphasise particular statements and to order them according to multiple criteria are also mentioned.

> *"You can designate as many different kinds of links as you wish, so that you can specify different display or manipulative treatment for the different types."*

ScholOnto builds on this idea by providing a rich taxonomy of 'links' (discourse relations) to connect 'notes' (concepts.)

Idea processing systems

NoteCards is a practical realisation of Engelbart's vision. It is a pioneering hypertext environment to connect, organise, store and retrieve thoughts encoded in electronic cards [Halasz et al., 1987]. Four objects are defined: (1) notecards, similar to paper notecards, in which any amount of information can be stored; these cards can be differentiated according to the amount of information they contain; (2) links to connect notecards into networks or structures of ideas (user-defined labels can be assigned to links.); (3) browsers, containing automatically computed graphical representations of structures of cards and letting users navigate within this structure; and (4) file-boxes, used to organise large number of cards by displaying them as a list.

ScholOnto is also strongly influenced by this project, as it provides types and links to connect multiple 'notes.' It is interesting to notice that the formalisation problem ('*How do I break my ideas into coherent nodes?*') that we have seen at the end of the second chapter [Shipman and McCall, 1994], is mentioned in a study of NoteCards, because of the need to separate ideas into delimited elements. Halasz et al. report on the problems faced by users in an 'idea processing' task, and in particular on their difficulties to segment their ideas and information into notecards: questions about the amount of information to put into a notecard and difficulties to interconnect them with links are noted.

Concept maps

Concept maps form another structured representation that can efficiently support sense-making. They are informal views (meaning that no semantics is associated with them) of a question or a domain [Novak, 1998]. Typically used to represent the understanding of a scientific domain by groups of children, they can also help answer a particular question by making sense of a document or of a group of documents (c.f. figure 4.6.)

Concept maps are composed of concepts and propositions. Concepts, '*perceived regularities in events or objects, or records of events or objects, designated by a label*', are articulated into propositions. Propositions, '*statements about some object or event in the universe, either naturally occurring or constructed*', connect two or three concepts and are directed. If a proposition [A, supports, B] exists, it does not necessarily follow that [B, supports, A.] Concepts can also be connected in multiple propositions.

ScholOnto structures are similar to concept maps, with the exception that the relations one can select are fixed. They are similarly created to answer one question (c.f. section 4.2.)

Another ScholOnto-related feature of these maps is the presence of cross-links. Cross-

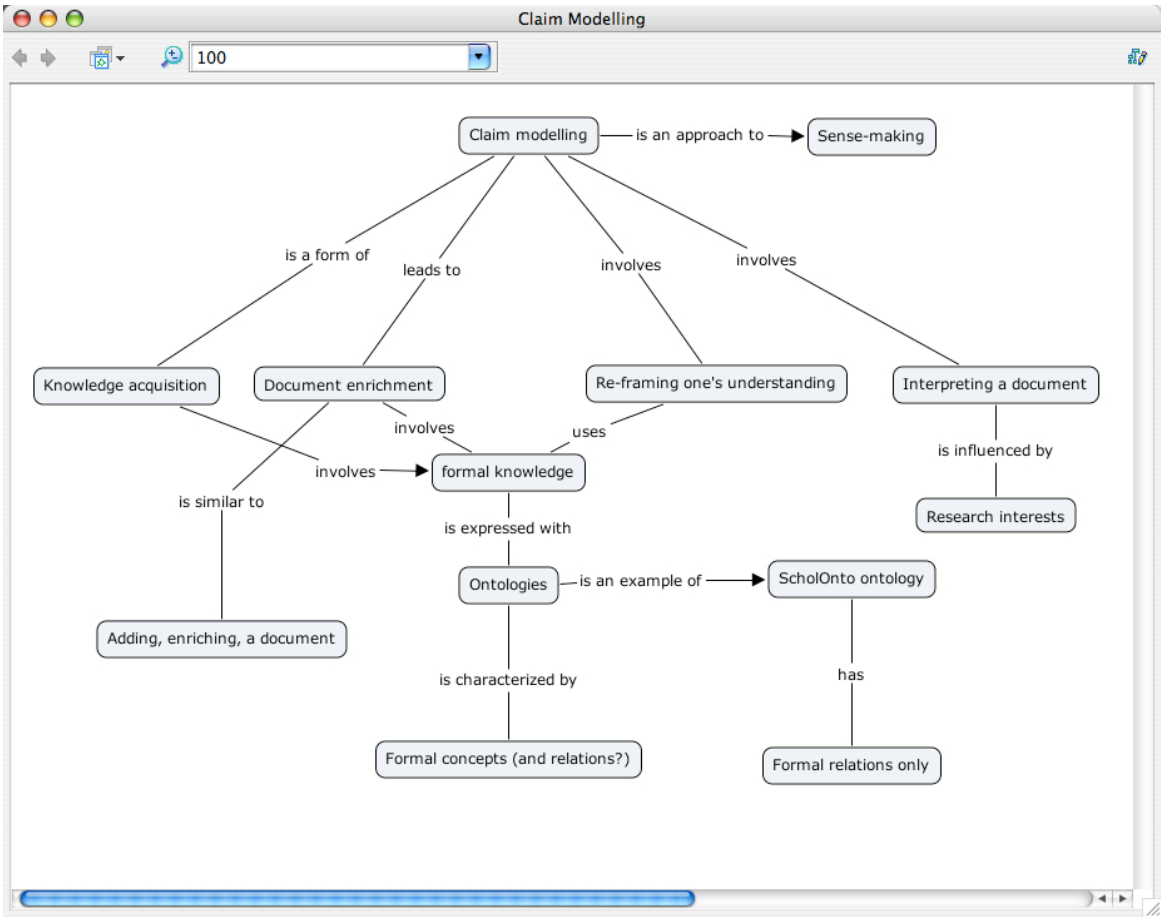


Figure 4.6: A concept map created to answer the question '*What is claim modelling?*' Generic concepts are at the top, while concepts that are more specific are at the bottom of the tree-like structure. The proposition [Knowledge acquisition, involves, formal knowledge] is a cross-link, a connection between two concepts coming from two different branches of the structure.

links create propositions connecting concepts in different areas of the map. They are represented by horizontal connections in a map, differing from vertical ones that indicate the hierarchical organisation of the concepts. Used to articulate indirectly related concepts, they represent unexpected associations, or '*creative leaps*' between concepts. In a ScholOnto scenario, links could be created between concepts defined between two *à priori* unrelated documents, in order to represent similar creative leaps.

Dialogue maps

Dialogue maps are used to record (and make sense of) the different positions of a group of stakeholders aiming to propose a solution to a wicked problem. Wicked problems are problems for which no clear solution appears at first, and for which any answer is a compromise of many factors [Kunz and Rittel, 1970]. They have the following characteristics:

1. The problem is not understood until after formulation of a solution.

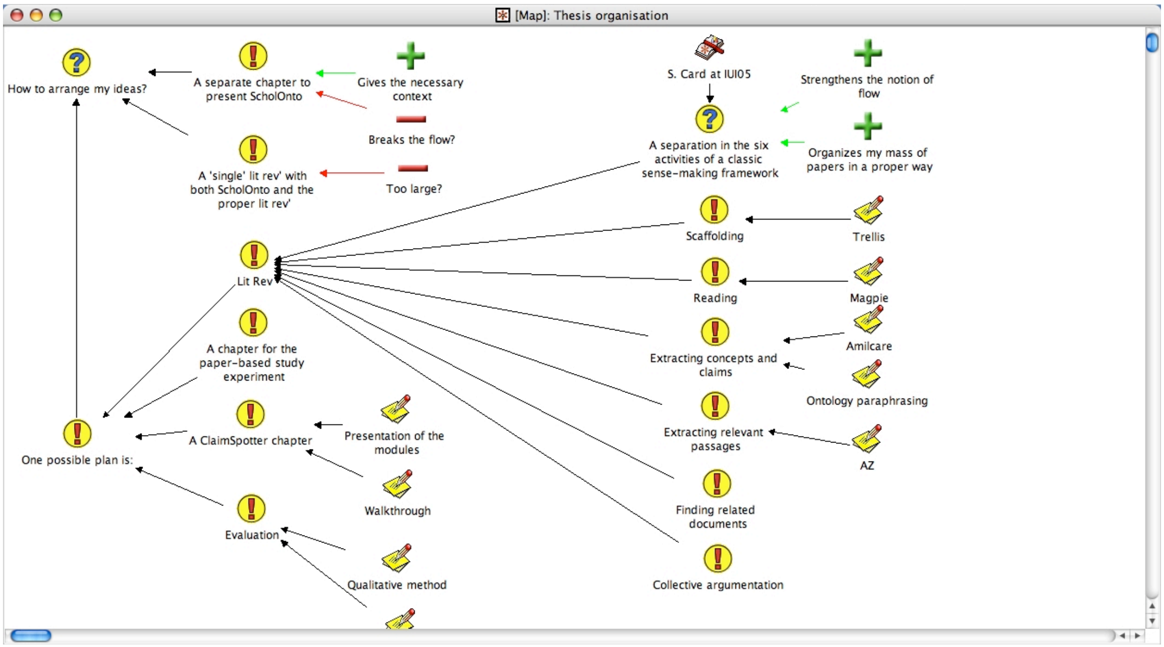


Figure 4.7: Decision rationale (an activity supported by dialogue mapping) in Compendium: *questions* (represented with a '?' symbol) can be broken into smaller *questions*, *answers* ('!') proposed and motivated with *supporting* ('+') or *dismissing* ('-') advice. *Notes* can also be added.

2. Stakeholders have radically different worldviews and different frames for understanding the problem.

3. Constraints and resources to solve the problem change over time.

4. The problem is never solved.

Models are proposed to capture the relevant arguments expressed in debates between stakeholders. The IBIS model [Halasz et al., 1987], for instance, represents the argument surrounding a topic as a structured representation of nodes (with types *issues*, *positions* and *arguments*) interconnected with links (*responds-to*, *questions*, *supports*, *objects-to*, *specializes*, *generalizes*, *refers-to* and *replaces*.) A link can connect an *argument* with a *position*, capturing the fact that the former *supports*, or *objects-to* the latter. *Issues* can *question* an *argument*, *specialize* or *replace* another *issue*. Implementations of this formalism include gIBIS [Conklin and Begeman, 1988], QuestMap and the recent Compendium [Selvin et al., 2001], which includes a participatory approach to the construction of a dialogue map (c.f. figure 4.7.)

A dialogue map is also similar in spirit to a ScholOnto claim space: both represent the possibly contesting views of different participants, in answer to a question that has no easy solution. ScholOnto proposes an asynchronous variant, in which annotators express them-

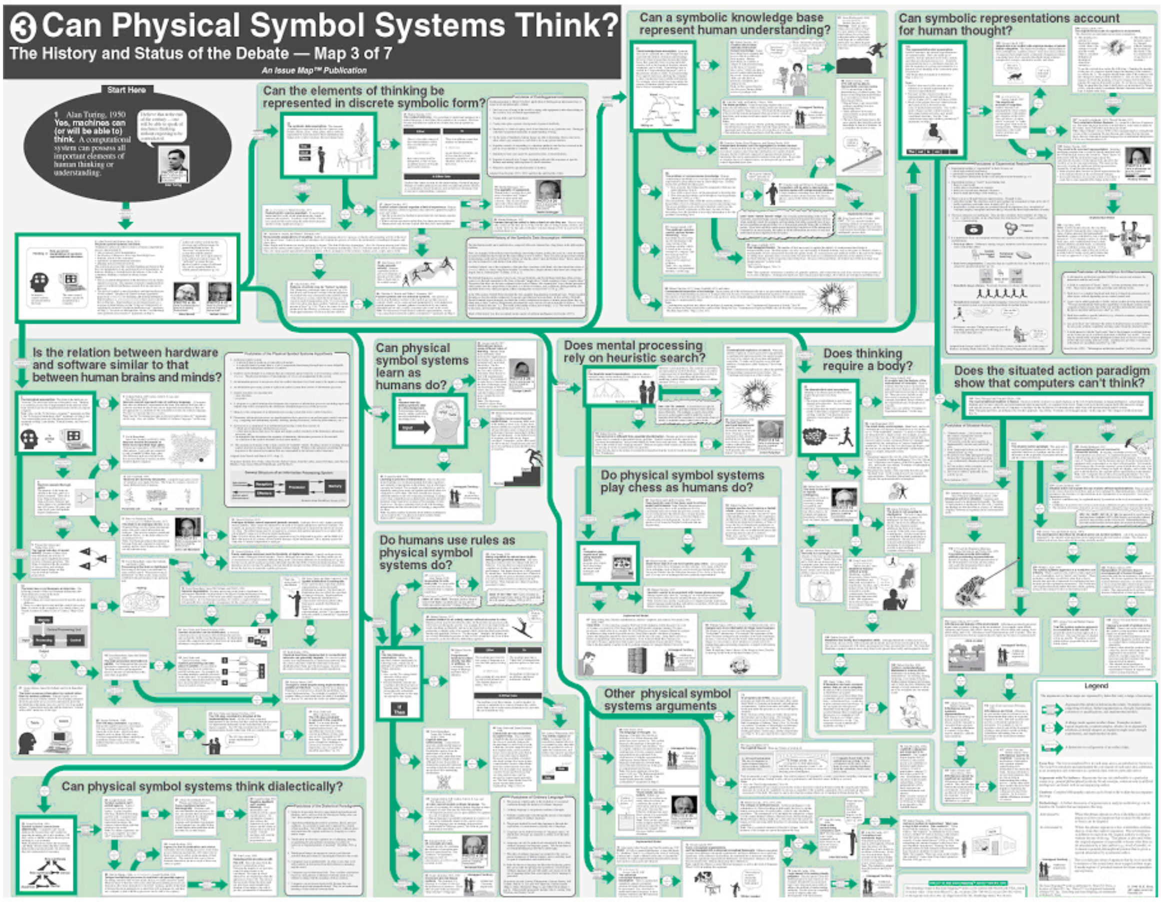


Figure 4.8: The 'Great Debates' project organises the arguments of generations of researchers by summarising their arguments (into nodes boxes) and connecting them together (via un/supporting links.) In addition to 'boxes' representing the different claims, additional boxes can be added to introduce more detail or a definition.

selves and debate at different times [3].

Argument maps

Argument mapping environments can be used to structure participants' understanding of a domain, by making the structure of an argument explicit [van Gelder, 2002]. For instance, Horn's 'Great Debates' project [Horn, 2003] represents in an explicit way the claims made by the researchers of different communities and their interconnections, in answer to a broad question such as '*Can computers think?*' (c.f. figure 4.8.) The links between the different nodes of the map indicate the transitions from a claim to another claim, as they have happened over time.

SenseMaker SenseMaker provides a workspace in which evidence can be spatially (by moving text strings over a workspace) and categorically organised (by arranging them into

[3]The ClaiMapper environment we have presented earlier, c.f. figure 2.7, page 26, is built on Compendium.

theories and conceptual categories, or groupings (also called claim frames)) [Bell, 1997]. Students' main activities are to place the evidence 'dots' into the appropriate claim frames and to organise them (a larger claim frame for a theory can contain smaller ones.) Evidence can also be categorised in multiple claims. Students are moreover asked to provide notes about both the evidence and the claims created, to justify their decisions and their arguments.

CSILE CSILE provides a 'community knowledge space', in which different participants can record their theories, opinions, evidence... into notes and organise them into structures [Scardamalia, 2004]. Notes can be enriched with further information such as annotations, citations to additional notes and links to citing notes. CSILE adds support to create multiple perspectives on these notes and structures via the definition of views, which gather information collected by a group of persons or the information of a particular type (such as 'problems.')

CLARE CLARE is another collaborative argument construction environment that relies on two separate components - RESRA and SECAI - to build representations of an artefact (typically a scholarly paper) [Wan and Johnson, 1994]. A scholarly document is first represented as a set of nodes (corresponding to sections and sub-sections.) Learners create summary and evaluative nodes out of selected text by picking the correct node type (*problem...*) and providing the relevant information. Link primitives can then be added between nodes.

Belvedere Belvedere is a system to support 12-15 years olds in their critical discussion of science issues [Suthers et al., 2001]. It is a graphical environment in which ideas and their relationships to each other are represented with diagrammatic shapes and arranged spatially in a workspace. Mechanisms to enable several students to work on the same map (with multiple machines and input devices) are provided, making use of a set of colours to identify the portions created by each user.

SEAS SEAS is an approach to support the capture of analytic thought [Lowrance et al., 2001], defined as '*the examination and separation of a complex situation, its elements, and its relationships*.' Lowrance et al. mention several factors likely to influence an analyst, such as '*her own knowledge about the facts and assumptions, the knowledge of her peers, the hypothesis that can be drawn from the facts and the evidence supporting those hypotheses.*' Hierarchically structured sets of inter-related questions are proposed to an analyst to assist

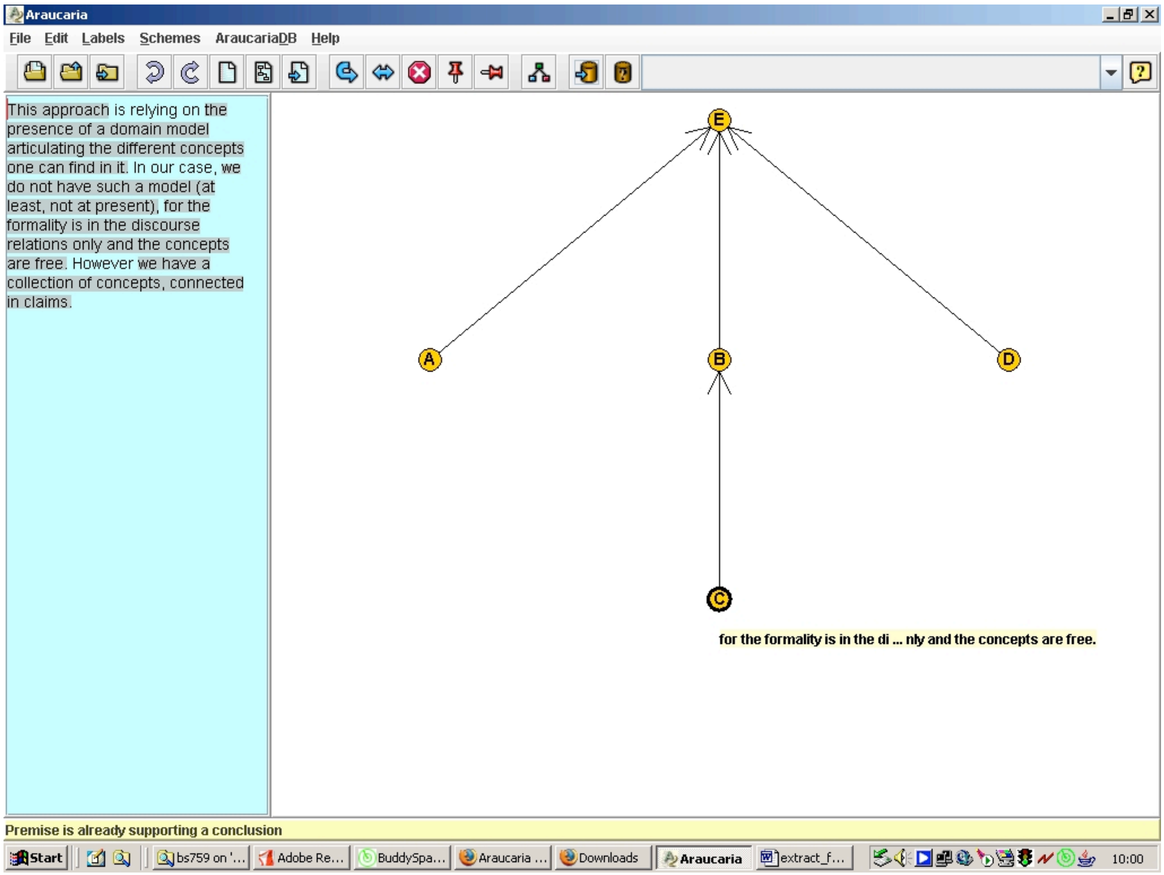


Figure 4.9: Modelling an argument map in Araucaria: nodes are created from text extracts and relations are drawn between the nodes.

the formulation of her answer to a complex question [4].

Reason!Able In Reason!Able [van Gelder, 2002], argument trees - composed of *claims*, *reasons* and *objections*, and *inferential relationships* holding between them - can be constructed in a graphical environment. Support to evaluate the intrinsic quality of the arguments modelled is also proposed (we detail it later.)

Araucaria Araucaria is a modelling interface to represent arguments [Prakken et al., 2004] that can be articulated using different schema. Figure 4.9 gives an example of the creation process of an argument map. Text fragments (on the left side) are selected and converted into nodes on a 'drawing pad' (on the right side.) Nodes can then be structured via the definition of relationships between them.

Trellis Trellis focuses on the notion of trust users have in diverse sources of information on the Web, enabling them to record their information analysis [Gil and Ratnakar, 2002]. The nodes of a structured Trellis representation are statements summarising information,

[4] The experiment reported in chapter 7, c.f. page 219, is a first step in this direction.

connected using a set of discourse, logical and temporal connectives. Controversial and dismissed statements can also be added. An analysis can be shared, and reused partly or wholly in further analyses.

The claim spaces built in ScholOnto are similar in spirit to the structured representations created by these applications: they structure the analysis into multiple nodes that can be articulated with relations. They also support a variant of group argumentation, recording the different (and possibly contesting) interpretations made of scholarly documents.

4.2.4 Discussion

This section has presented three mediums to support sense-making at three different levels of formalisation: informal notes, focussed dialogues and rich structures. Although ScholOnto models are closer to the last of these approaches, they also share some aspects with the former two. Concepts (identified and grounded into documents) are informal pieces of information which have no constraint on their content. The models also function as 'discussion rooms' in which annotators can 'discuss' and take position with their peers' claims (c.f. example in section 2.1.5, page 18.)

We focus now on the five activities identified in the sense-making process (c.f. figure 4.2) and successively present approaches that can efficiently support each of them.

4.3 Scaffolds

The first activity of our annotation process is related to scaffolds. By scaffolding the annotation process, we can provide some sort of structure to either constrain the user in telling her what not to do with her data, or, in a more positive way, to give her hints about ways to organise her data in a satisfying manner. Four elements can be used: (i) types to characterise its content; (ii) templates to guide her thinking; (iii) advice modules giving suggestions based on the current state of her knowledge, and (iv) detailed guidelines.

4.3.1 Item types

Characterising the role played by an 'item' (a note, a node…) is an approach found in the three sense-making medium we have presented.

Types in document annotation tools

In document annotation tools, a role is used to further characterise the content of a note. Annotea [Kahan et al., 2001] offers types such as *advice*, *change*, *comment*, *example*, *explanation*, *question* or *see-also*. Haystack [Karger et al., 2003] offers types such as *reminder*, *problem*, *complaint*, *idea* and *plan*. Annotations in YAWAS [Denoue and Vignollet, 2000] include categories (*point of view*, *article title*, *definition* and more) but also types (*person name*, *technology*...) and additional stances expressing the annotator's point of view (such as *agreement* or *disagreement*; these types can be customised in a configuration file.) Note types are also available in CritLink [Yee, 2002]: it offers another source of scaffolds via the definition of connections between notes that can be instantiated in hyper-links.

Types in discussion spaces environments

Types are also used to characterise the contributions made in discussion spaces environments. The motivation underlying a comment is therefore kept in a more explicit way. In SpeakEasy [Hoadley and Linn, 2000], comments have to be categorised using a semantic label, to choose between *and*, *or*, *but*, *i.e.* and *?*. In D3E [Sumner and Buckingham Shum, 1998], an optional categorisation of the nature of a comment can be provided, to choose between *agreement* and *disagreement*.

Types in structured representations

In a structured representation, types can be added to its two constituents: its nodes and its edges. They can be defined freely by the user or constrained.

In the IBIS method [Halasz et al., 1987], a wicked problem is broken down into *issues*, *positions* and *arguments*, forcing thinking out by encouraging each participant to state her position using arguments. Typed links are also available (we have listed them in our presentation of IBIS earlier.) Typed nodes can be created in Belvedere [Suthers et al., 2001], including for instance *objects*, *theories*, *hypotheses*, *claims*, *warrants*, *observations* or *laws*. These objects can be combined with relations such as *supports*, *causes*, *explains*, *then*, *and*, *conflicts* or *negates*: a typical Belvedere statement is (copied from the cited paper): *(The oldest rock on the Galapagos islands has been measured as being between 2 and 3 million years old by radioisotope dating, explains, The Galapagos islands are 2-3 millions old.)* In Reason!Able [van Gelder, 2002], *claims*, *reasons* and *objections* are combined to represent the elements of an argument. In Trellis [Gil and Ratnakar, 2002], statements (expressing an analysis of the information found in a particular Web source) can be connected with relations

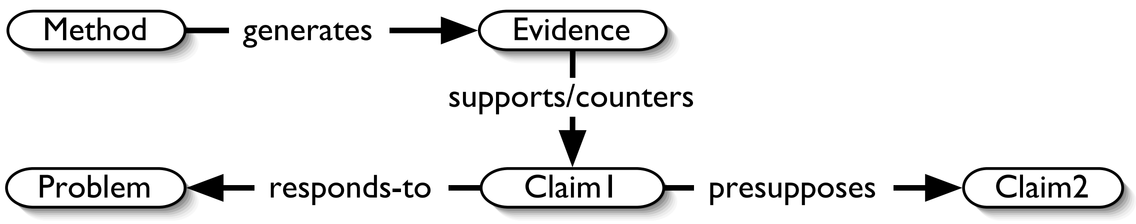


Figure 4.10: A canonical form in the RESRA formalism connects node primitives with link primitives. Filling this canvas provides support by telling learners what they *should* extract from a paper.

such as *results in*, *is supported by*, *is elaborated in*, *before*, and enriched with a subjective indication of their creator's reaction (*surprise*, *accurate*.)

RESRA (for 'REpresentational Schema of Research Artefacts'), a part of CLARE [Wan and Johnson, 1994], is a semi-formal representation language. It defines summary (*method*, *evidence*, *claim* or *problem*) and evaluation node primitives (*critique*, *question* and *suggestion*.) It also uses link primitives to define relationships between two nodes (for instance, *generates*, as in {*a method*, *generates*, *evidence.*})

4.3.2 Question-oriented templates

Assistance can also be brought by providing question-oriented templates. In Compendium [Selvin et al., 2001], the identification of notes (leading to a dialogue map) is achieved by an expert user called a facilitator. Support is provided in the form of question-oriented templates that help to guide the discussion. These templates are composed of a set of issues that have to be addressed and can be provided by the participants themselves.

In SenseMaker [Bell, 1997], a frame library containing a menu of frames that can be instantiated by the students is provided. Bell reports on students finding this feature useful: "*features of the interface (e.g., evidence being placed within claim frames) allows for student self-expression of ideas and can promote individual reflection on prior knowledge*."

CSILE provides support via scaffolds that "*give ideas defined roles in such processes as theory refinement and constructive criticisms*." They can be assimilated to blank canvases to guide thoughts into these difficult processes. Another source of support is given by a clear statement of the purpose of the current debate. Keeping this purpose visible at all times encourages either contributions to this debate or the "*start of a new field of inquiry*" related to it.

CLARE

The canonical forms provided in CLARE also act as 'blank canvases.' Different canonical forms can be created to suit different kinds of papers, according to the nature of the research being reported. A typical form can include a *method generating evidence*, *supporting* a particular *claim*, *presupposing* itself another *claim* and *responding to* a *problem* (c.f. figure 4.10.) Canonical forms can be used to ensure that learners combine primitives in an appropriate manner. The authors report on the population of this canvas as a "*meaningful learner experience,*" as "*RESRA node primitives, such as the problem node, may only be implied rather than explicitly stated by the author*."

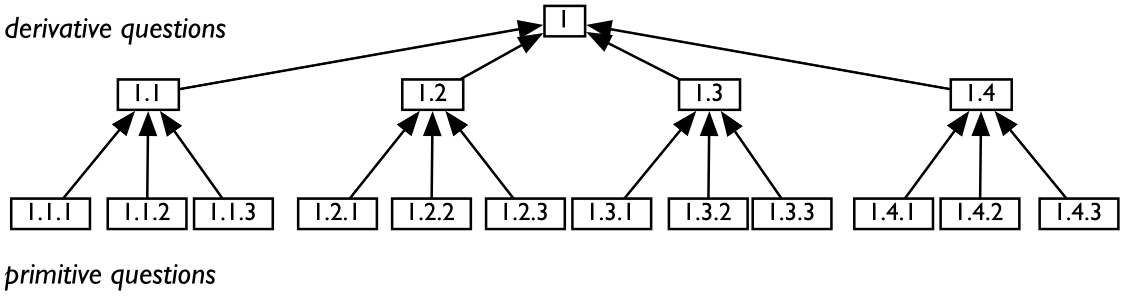
A structure in ScholOnto could ensure that annotators create a few predefined concepts (such as a problem, an approach and an opinion) and connect them with appropriate relations (such as ***addresses***.)

SEAS

Structured arguments in SEAS [Lowrance et al., 2001] are represented as hierarchies of questions: *primitive* questions (at the bottom of the hierarchy tree) are answered directly, via the selection of an entry in a multiple-choice question or the choice of a value. *Derivative* questions are then obtained from these questions and answered via heuristic methods based on their answers (one possible inference being '*take the maximum answer as the conclusion when combining several questions assessed along a continuous scale*.') Figure 4.11 gives an example of an argument template composed of primitive and derivative questions.

Argument templates - providing the backbone for the arguments - can be authored via a top-down or a bottom-up approach. In the latter case, the analyst starts with the conditions resulting in a positive answer to the question being asked (for instance, '*Is this country becoming increasingly...*') and clusters these into coherent collections. In the former case, the analyst starts with a main question, attempts to break it into smaller ones and repeats this process until the question can be answered directly. If it cannot be answered directly, discovery tools (e.g. link to Web pages, database queries) can be invoked to assist her.

In ScholOnto, an argument template could be created for a given type of scholarly paper, containing a question '*Is this article relevant to your research?*' This question could be broken down in several primitive questions such as '*Is the problem addressed relevant to your research?*' and '*Is the approach proposed relevant?*' Annotators would instantiate the template for a particular paper.



1.1.1: *Is this government becoming increasingly unstable or weak?*
1.1.2: *Is increasing conflict over policy/issue areas having a destabilizing effect?*
1.1.3: *Is decreasing public confidence in the leadership or government policies having a destabilizing effect?*

1.1: *Is political instability increasing?*

1: *Is this country headed for a political crisis?*

Figure 4.11: A hierarchy of questions in SEAS: primitive questions (1.1.1, 1.1.2, 1.1.3...) take fixed values as answers and contribute to answering derivative questions (1.1, 1.2,... 1) via heuristics.

Trellis

Two recent extensions to Trellis, Tree Trellis and Table Trellis, are proposed to scaffold annotators' statements and make them amenable to computer analysis [Chkolvski et al., 2005].

Tree Trellis simplifies the Trellis language by (i) removing conjuncts such as 'AND'; (ii) reducing the set of connecting primitives (used between user statements) to 'pro' or 'con'; and (iii) serialising the presentation of statements to avoid the duplication of the initial question in users' statements for instance. Figures 4.12 and 4.13 show similar arguments in, respectively, the original (rich) Trellis and Tree Trellis.

Table Trellis goes even further by 'reducing' statements to a table, in which the different alternatives considered are listed in the y-axis, and the different features enabling the comparison are identified by the analyst and put along an x-axis (c.f. table 4.14.) The analysts' task is then to assess the value of each feature (on the x-axis) for each alternative (on the y-axis.)

Chkolvski et al. report that statements inputted in Table Trellis are more likely to support automatisation, for most of the complexity of dealing with natural language is removed. Annotators in Table Trellis have indeed to break their 'argument' down (as features), and to assess a value for each of these. As both features and their values are typically expressed with a few words only, more automatisation and clustering of arguments can be achieved.

While the goal of ScholOnto is not to structure annotators' statements, we are still in-

Macintosh is more usable than Windows?
  Macintosh is more usable than Windows **is supported by**
  Macintosh platform has a more stable OS
    Macintosh platform has a more stable OS **stands though contradicted by**
    Windows aims to surpass other platforms in security and stability
  Macintosh is more usable than Windows **is supported by**
  Macintosh, as compared to Windows, has a friendlier UI

Figure 4.12: An analysis in the original (rich) Trellis. The richness of this formalism makes it difficult to structure the statements submitted by the analysts, as there are many ways to express opinions (with combinations using 'AND' for instance.) This results in an increased difficulty to translate them into machine-readable form.

Macintosh is more usable than Windows?
  pro: Macintosh platform has a more stable OS
    con: Windows aims to surpass other platforms in security and stability
  pro: Macintosh, as compared to Windows, has a friendlier UI

Figure 4.13: Tree Trellis limits the formalism to two connectives, 'pro' and 'con.' This view is similar to the IBIS formalism, with position and argument nodes merged.

Best computer?				
Type	**Design**	**Costs**	**Troubleshooting**	**User experience**
Apple	Excellent	Moderate	Excellent	Excellent
PC	Medium	Moderate	Moderate	Poor

Figure 4.14: Table Trellis constrains the formalism even more, by asking analysts to structure their argument as a set of features ('Design', 'Cost), for which values must be provided for each alternative considered ('Apple', 'PC.')

```
(def-advice support competitor
  :advice ''Could the empirical data that supports one theory also support
   the other ?''
  :arguments (?t1 ?t2 ?d)
  :test (:and (theory ?t1) (theory ?t2) (empirical ?d)
    (:not (:same-as ?t1 ?t2)) (supports* ?d ?t1)
    (:not (supports* ?d ?t2))))
```

Figure 4.15: This advice ("*Could the empirical data that supports one theory also support the other?*") is triggered in Belvedere and presented to the student, if some empirical data is supporting one theory, and if a different theory (which is not supported by the data) is found.

terested in making concepts reusable when possible. A *Table ScholOnto* may be useful to compare two approaches, by listing comparison features such as 'problem addressed', 'costs', 'quality of results'... and asking annotators to specify a value for each alternative.

4.3.3 Advising modules

Intelligent advisers can also be considered to scaffold the train of thoughts of an analyst. In Belvedere [Suthers et al., 2001], an advising module to "*stimulate critical discussion that would not take place otherwise*" is available on demand. The different advices are phrased not as imperative statements such as '*Replace this!*', but instead as suggestions and questions, thus leaving the possibility to students to dismiss them. Advice is based on the structure and category of the nodes (the objects they are connected to and their type.) Figure 4.15 gives an example of the criteria needed to trigger an advice.

If several pieces of advice are triggered by the elements in the map, they are ranked according to a set of preferences. The first one is given and the other ones are available on demand. This advice has two aims: scaffolding the construction activity on the part of users, to help them submit more comprehensive and robust models, and detecting illegal constructions (resulting from an incorrect association (semantically-speaking) of the primitives.)

The Socrates Assistant in Reason!Able [van Gelder, 2002] offers two sources of support to students in their handling of arguments: (i) it helps them identify and evaluate arguments submitted by their peers, and (ii) it helps them evaluate their own arguments. This support is brought by switching on Socrates on the current object in the display, which triggers contextual advice about it. Advices such as '*Consider what can be said in support of this premise, and what can be said against it*' help students frame their arguments by reminding them to include evidence and counter-evidence, resulting in models that are more robust.

Support could be provided in ScholOnto with an advising module. It could suggest annotators to consider modelling a *problem* concept when an *approach* concept is created, or to consider modelling the alternative *approaches* (addressing the same *problem*) they are aware of (in different scholarly articles.)

4.3.4 Guidelines

The last form of scaffolds we consider in this section is the definition of a set of guidelines. In Horn's 'Great Debates' project [Horn, 2003], nodes do not only contain 'simple' (in their expression) labels, but also longer summaries or reformulations of the arguments proposed and defended by researchers participating to the debate. As these reformulations have to be written in simple and declarative sentences to maximise their understandability, this process is tedious and difficult [Horn, 2003, page 171]:

> "*It is very easy for academic writers to hopelessly complicate sentences. This does not make for good argumentation mapping. Many is the time that our writers remarked that they wished that the debate protagonists whom they were summarising would have written in such a succinct style.*"

A set of guidelines is provided to help maps writers decide whether to include (or not) a given claim, rebuttal or counter-rebuttal (c.f. table 4.1.)

4.3.5 Discussion

We have presented different approaches to scaffold sense-making. These approaches include the definition of types that can be added to the various items she creates and manipulates, the provision of question-oriented templates to guide her thinking, advising modules to provide help when requested (and enforce the created artefacts are more robust) and finally the definition of a set of guidelines to follow. These approaches can be efficiently implemented in an annotation interface, to support the sense-making process in two directions, as we have noted: by indicating what is not allowed, and by proposing suggestions to create more comprehensive and robust ScholOnto literature models.

4.4 Reading

For the second activity of our sense-making workflow (c.f. figure 4.2, page 41), we move back to the document. The digital realisation of a document on a display lets us envisage additional supporting facilities such as its contextualisation (with respect to a particular domain

Use only published arguments
Arguments should be published in a formal way to be considered. This includes conference proceedings, but dismisses electronic fora.
Use only arguments that lie within the scope of the map
The arguments added to the map must be significant to the main question that is being answered.
Seek out historically earliest or best known
The first instance of a claim is retained and the following ones are dismissed. Unless one of the following has become (over the years) the *de facto* one, in which case, it becomes the one mentioned (the original one being added in a side note.)
Avoid loosely drawn arguments
Arguments which are not presented in an appropriate way are not considered.
Avoid repetitive, nitpicking or duplicative arguments
Avoid forbiddingly technical discussion
They were dismissed because they could not be represented easily within the map. Nothing would prevent them from being included in a different map for a different audience however.
Summarize the author's published claim
Even if authors significantly change their position after some time, their original position, their original claims are kept in the map to show the evolution of ideas.
Avoid tentative arguments
Arguments which are not being defended in an authoritative way are not considered.
Include some rather ancient argument
Some historical arguments are provided to extend the historical scope of the debate.
Include some experimental results
The concrete scope of the debate is also extended by including the most famous experiments' results.
Include a small sample of outrageous and humorous arguments
Finally, a number of what has been called 'stranger' claims are added to foster debate and discussion.

Table 4.1: Criteria for including a claim, rebuttal or counter-rebuttal node in a 'Great Debates' map [Horn, 2003].

or task.) Another possibility lies in the 'integration of annotation with reading as an activity', which Marshall notes as being important to make good annotation environments [Marshall, 1997] (c.f. page 41.)

4.4.1 Content exploration

Facilities to explore a document and to facilitate its transformation into knowledge [O'Leary, 1998] are important to support an efficient sense-making process.

The SuperBook environment facilitates the exploration of (the ideas expressed in) a document by transforming it into a collection of nodes that can be browsed in any order [Remde et al., 1987]. *Document windows* in SuperBook include a title, a page of text, a table of contents (automatically generated, from the formatting macros in the original text), a word look-up function and figures. From the main window, a reader can jump directly to any given section. As the fraction of text being displayed is associated with its heading, the reader always knows where she is in the document. The *table of contents* also shows different levels of detail, enabling her to browse a section to access any of its sub-sections. A *full-text index* is accessible: querying it updates the table of contents by displaying for each section how many occurrences of the current word or combination of words have been found, enabling her to discover the distribution of the query term across the document.

The repartition of query terms can also be visualised to identify the relevant parts of a document in the TileBars project [Hearst, 1995]. TileBars uses an approach called TextTiling (which we present later, c.f. page 81) to break a document into topically-coherent units (in which only a single sub-topic is discussed), thus helping readers to decide both which documents to read and which parts of them to focus on. Figure 4.16 gives an example of the contextualised documents returned to a query.

3Book [Card et al., 2004] provides an interactive representation of a book, enhanced with additional facilities to support sense-making. For instance, a reader can bookmark a location in a book, 'extract' pages and display them side-by-side to compare them. It is also possible to issue queries over the content of the book: sentences containing the query words are highlighted and easily accessible [Chi et al., 2005]. Results are enhanced by the discovery of additional terms related to the query terms - resulting in more (relevant) sentences being highlighted. When a query is issued, the conceptual index of the book is adapted: pages containing relevant terms are clustered and presented at the top of the index.

4.4.2 Contextualisation

The 3Book approach shows the benefits of contextualisation via querying: relevant snippets of the original document are highlighted and can be focussed to access potentially relevant information. This contextualisation can also be supported via the definition of a user profile, the automatic discovery of the most salient terms of a document and the use of a domain ontology.

User profile contextualisation

The Reader's Helper is a document reading environment providing support for the skimming and the extraction of information [Graham, 1999]. On the left part of the screen (c.f. figure

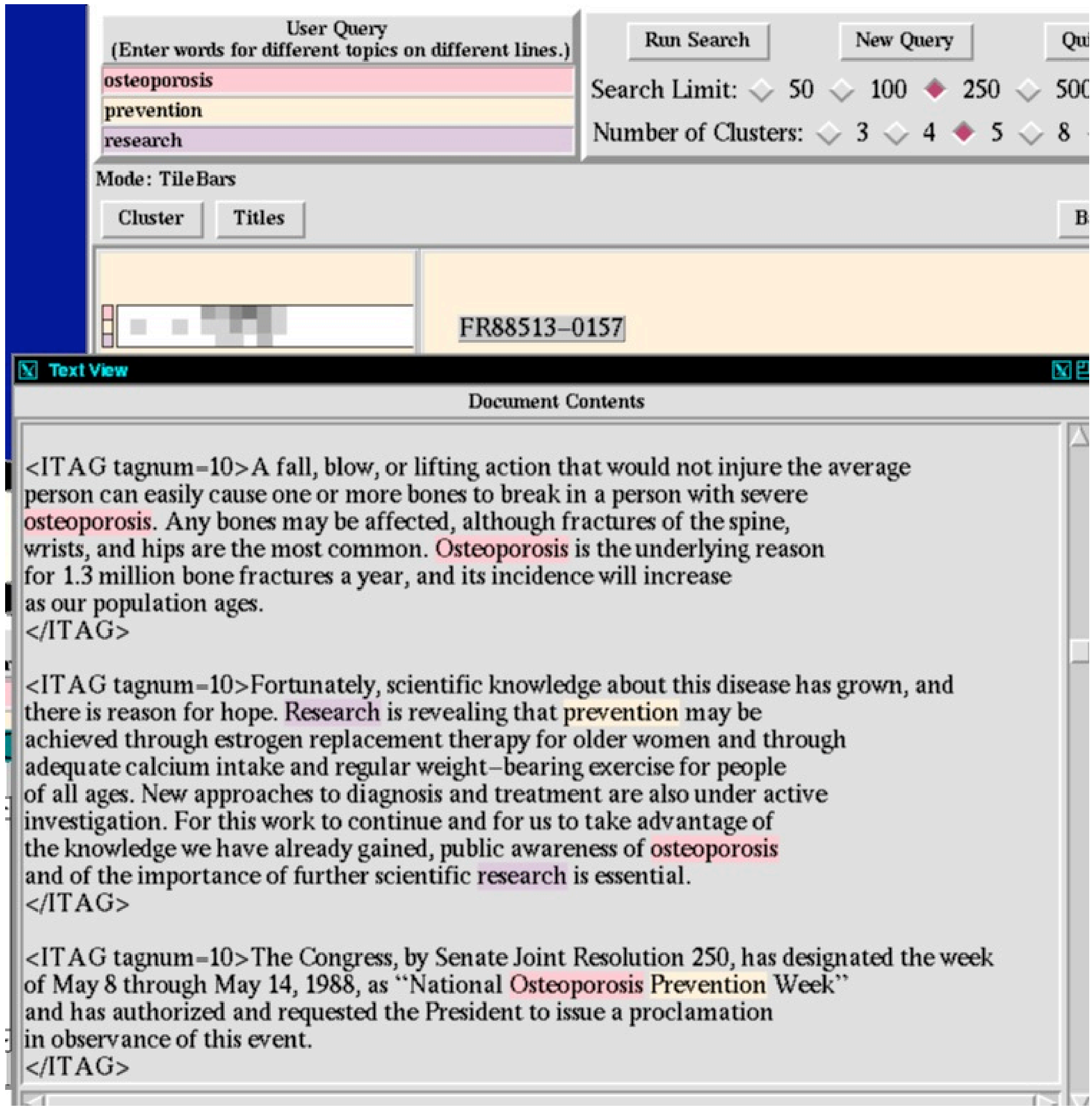


Figure 4.16: TileBars highlights, for each document retrieved, the 'blocks' (blocks are groups of adjacent word tokens, c.f. page 81 for more detail) in which each term appears.

4.17), a ThumbBar displays two thumbnail versions of the original document: readers navigate through the document by dragging a lens up and down these representations, mimicking a scroll-bar (the corresponding parts being displayed on the main, central frame.) Topics of interest are stored in a user profile and displayed in the right frame, and are associated an on/off button. Switching a particular topic on is reflected on the ThumbBar: its occurrences are highlighted by red lines, displaying where it is located in the document. Readers can thus create their own combination of topics, assess their distribution across the document and jump directly to the relevant locations.

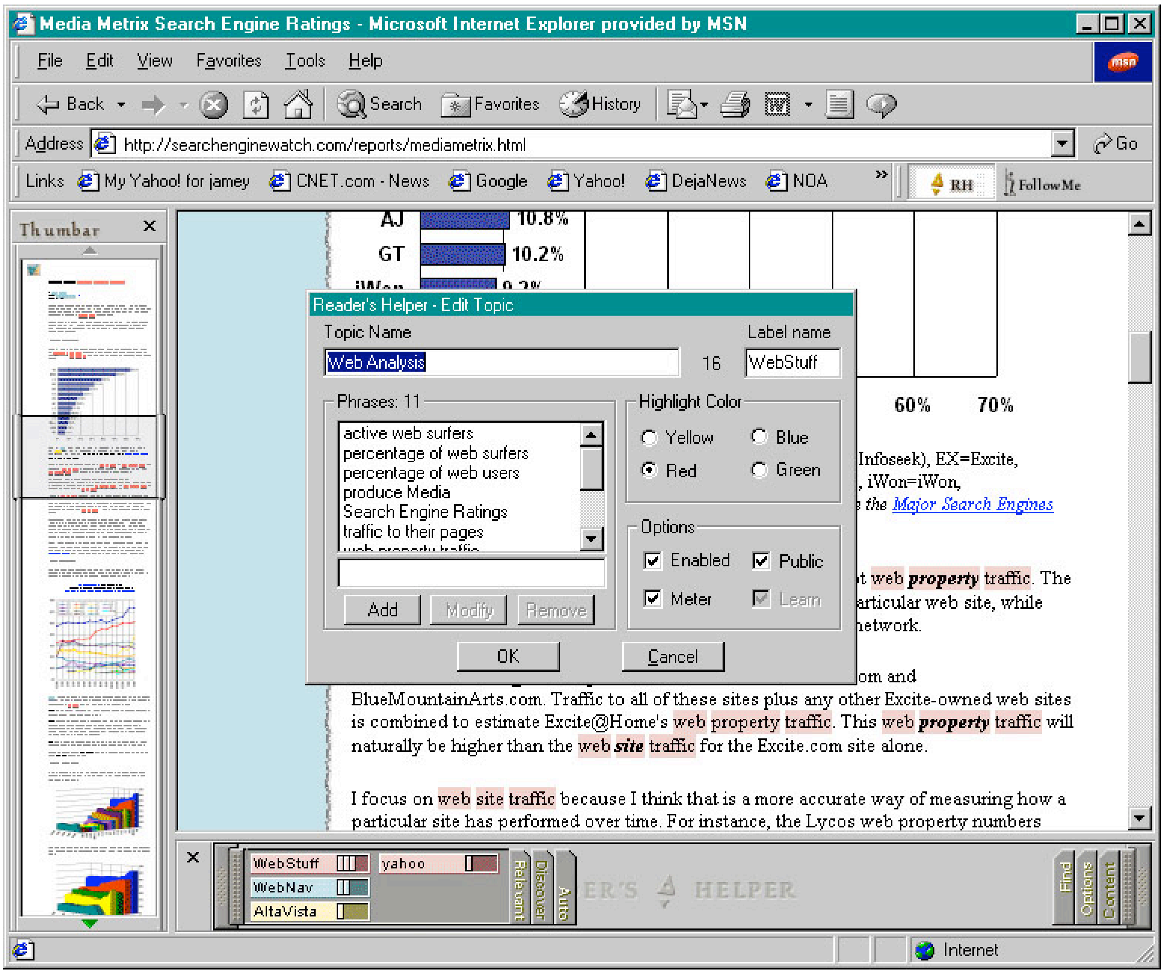


Figure 4.17: The Reader's Helper provides an environment to facilitate the contextualisation of a document. User-defined keywords are highlighted throughout the text and their repartition is shown on the ThumbBar (left frame), enabling a user to find out where they are mentioned.

Salient terms

ASHRAM is a document retrieval system that incorporates functions for automatically summarising (via sentence extraction) the documents returned to a query [Neff and Cooper, 1999]. Two levels of 'active' contextualisation are provided. First, the sentences of the (automatically generated) summary can be activated to display their occurrence in the full-text document. Then, the most salient terms (vocabulary items; including single- or multi-words (but without stop-words) that appear frequently in the document itself but not in the collection) are displayed in an upper frame of the results window and can be activated to query their related terms. Additional queries can then be made using these new (related) keywords.

Ontology-supported contextualisation

Contextualisation can also be supported by a domain ontology which not only highlights the concepts found in a document, but also offers the additional connections between the

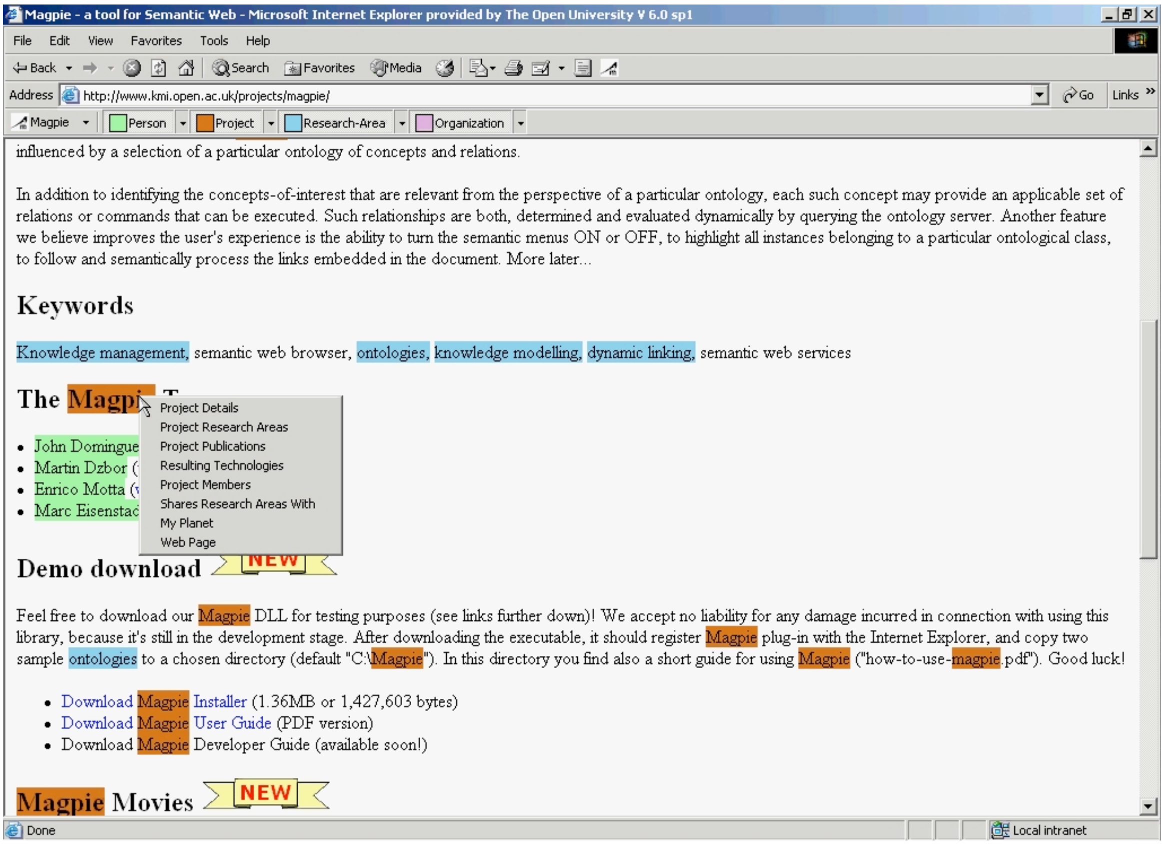


Figure 4.18: Magpie contextualises reading by highlighting the instances of concepts - organised in categories (*person*, *project*...), defined in an underlying ontology - which are matched in the document. (It also proposes additional services related to these instances.)

concepts it defines [5].

Magpie adds context to a Web page by highlighting the information found in the page that is defined in an underlying ontology [Domingue et al., 2004]. The contextualisation makes '*the document relevant in the context of the ontology*', and makes '*its ontology-related aspects (the concepts that might be found in its contents) salient*' via the addition of a semantic layer (c.f. figure 4.18.) For instance, given a Web page describing a researcher's activities, and an ontology describing the members, activities, projects and research areas of an organisation, Magpie can highlight the relevant extracts in the page, provided that this information is stored in the ontology. Additionally, a log of all the recognised entities (people, research areas, projects, and more) found in the current session is kept, enabling cross navigation between related pages. A contextual menu can be used to send different queries to the ontology server, based on the nature of the selected element: for instance (and because this information is available in some way in the ontology), a right click on a project name triggers a menu enabling to look for its detail, its associated research areas, its publications and its members. Figure 4.18 shows a Magpie contextual menu triggered by a right-click on a project name.

[5]We have given a brief overview of ontologies in section 2.1.2, c.f. page 14.

4.4.3 Integration of the annotation and reading processes

By presenting the document along the annotation interface, it becomes possible to tie reading with annotation again. This is the one crucial characteristic of our document-centric annotation environment. Existing applications already incorporate these two activities.

In SuperBook [Remde et al., 1987], comments can be added at any point in the document, by clicking on the desired location and typing the text in a small editor window. Their presence is signalled to future readers in the *Page of Text* window via an icon featuring the user's login and the date of her comment. A similar 'alert' mechanism can be found in YAWAS [Denoue and Vignollet, 2000], in which highlighted anchors can be clicked to reveal annotations in an external window. Annotea users can also annotate a whole Web page, the caret position or the current selection in the page (c.f. figure 4.4, page 44.) In the user interface of Haystack [Karger et al., 2003], an annotation panel is incorporated within the Web browser, letting readers add notes. Annotations written by others can also be accessed from the panel, in which they are organised in structures similar to threaded 'discussions.'

In D3E, the document being debated and the discussion space are displayed side by side (c.f. figure 4.5, page 45.) Notes (contributions to the discussion) can be added to its sections, and links to the discussion space are provided to comment any section of the document.

4.4.4 Discussion

As the annotators engaged in a ScholOnto sense-making process read a scholarly document and annotate it in parallel, possibilities to explore its content, to contextualise it with different sources of information (which are discussed further in the next section) and to seamlessly move from reading to annotation and vice-versa with as little interruption as possible are desirable.

Supporting reading by exploring the content of a document enables readers to focus on any aspect of the work being reported, by looking for the areas containing a particular keyword in a particular section for example. An approach such as 3Book is even more interesting to discover related keywords, and therefore additional areas that may not contain the exact query terms inputted by the user.

Contextualising reading with external information is another relevant possibility, as it shows the document through a lens. Ontology-based contextualisation *à la* Magpie seems to be, at a first glance, the most relevant contextualisation technique to our ontology-supported annotation task (c.f. section 1.1.2, page 2.) However, a major difference in the settings limits the underlying support we can provide (e.g. the different queries that can be sent to the ontology server.) Magpie relies indeed on a domain model to propose further links, based for instance on terms that are connected to the highlighted one via a relationship in the ontology. We do not have such a domain model, for only discourse relations are captured in the ScholOnto ontology and concepts are free in their expression. We could however make use of the annotators' submitted claims to propose, for any concept matched in the document, the concepts connected to it via claims or the list of documents in which this concept has been grounded.

Finally, integrating annotation with reading is also a desirable feature: displaying side-by-side a 'reading area' and an 'annotation area' facilitates the interaction between these two activities. It is also expected to diminish the cognitive burden on the user, as she does not have to refer to a printed version of the document or to another application displaying the document, as is currently the case in ClaiMaker and ClaiMapper (c.f. section 2.3, page 25.)

4.5 Concepts and claims extraction

Following our presentation of approaches to contextualise a document, we consider now the content of this contextualisation, in other words, the snippets that can be extracted and suggested.

We should restate again that we are not interested in annotating the content of a scholarly document, but rather an interpretation of its content. Extracting elements may thus seem at odds with our assumption that we cannot know in advance what will be modelled, as the annotation is influenced by factors such as one's research interests; the subtle difference in our approach is that this extracted information, if any, is only *suggested* to the annotator (and not recorded directly in a knowledge base.) The final decision (as to whether the suggestion makes a valid concept or claim) is left to the annotator.

4.5.1 Ontology population

Annotating a document in ScholOnto can be assimilated to populating an ontology, as we are instantiating concepts (i.e. discourse relations) in an ontology. It is however different from 'traditional' ontology population in the sense that our instances result from an interpretation of the content of the document (i.e. their formulation and their meaning may be different from the original author's formulation.)

Two ontology-related approaches can be considered to assist the creation of ScholOnto

Text	Template

1. Introducing ScholOnto
. . . However, there are few tools to track debate and analysis ideas in a domain. The Semantic Web [3] approach of augmenting Web documents with machine understandable information offers a potential means of addressing this need. The Scholarly Ontologies (ScholOnto) project [4, 5] takes this approach. . .
	`leftObject:` ScholOnto
	`leftObjectType:`
	`linkLabel:` supports
	`linkType:` Supports/Challenges
	`linkAuthor:`
	`rightObjectType:`
	`rightObject:` Semantic Web

Table 4.2: A ScholOnto claim as a populated template. Is it possible to fill such a template (right) automatically from a scholarly document (left) ?

triples. The first approach is to try to extract as much information as possible from a document, in an attempt to fill in a *template* representing the author's claims, and propose it to the annotator's consideration (who would be free to model it if she agrees, model a rebuttal if she disagrees or ignore it if it is not relevant enough to her.) The second approach is to start from a user-defined statement, expressed in free-text, and to *paraphrase* it to translate it in terms of the underlying ontology.

Template driven ontology population

ScholOnto claims could be assimilated to the blank skeleton given in table 4.2. Our goal would then be to discover as much information as possible to fill this skeleton. To achieve this, we would need to discover '*instances of a predefined class of events*' [Grishman, 1997]. Information extraction (IE) approaches seem to be a natural candidate to achieve our task:

> "*IE starts with a collection of (relevant) texts, then transforms them into information that is more readily digested and analysed. It isolates relevant text fragments, extracts relevant information from the fragments, and then pieces together the targeted information in a coherent framework.*" [Cowie and Lehnert, 1996].

Ontology-supported annotation tools are concerned with the marking up of information that *does exist* in a document to instantiate a class, resulting in the population of a pre-existing ontology. Although these applications are coined 'annotation tools' they are not similar to the ones we have seen earlier, such as Annotea [Kahan et al., 2001]. The term 'translation' is more appropriate here, as the expert's (a human or a software agent) task is to translate (in a formal language) information of a particular type that can already be found in the document. CREAM and MnM are two recent ontology-constrained annotation tools.

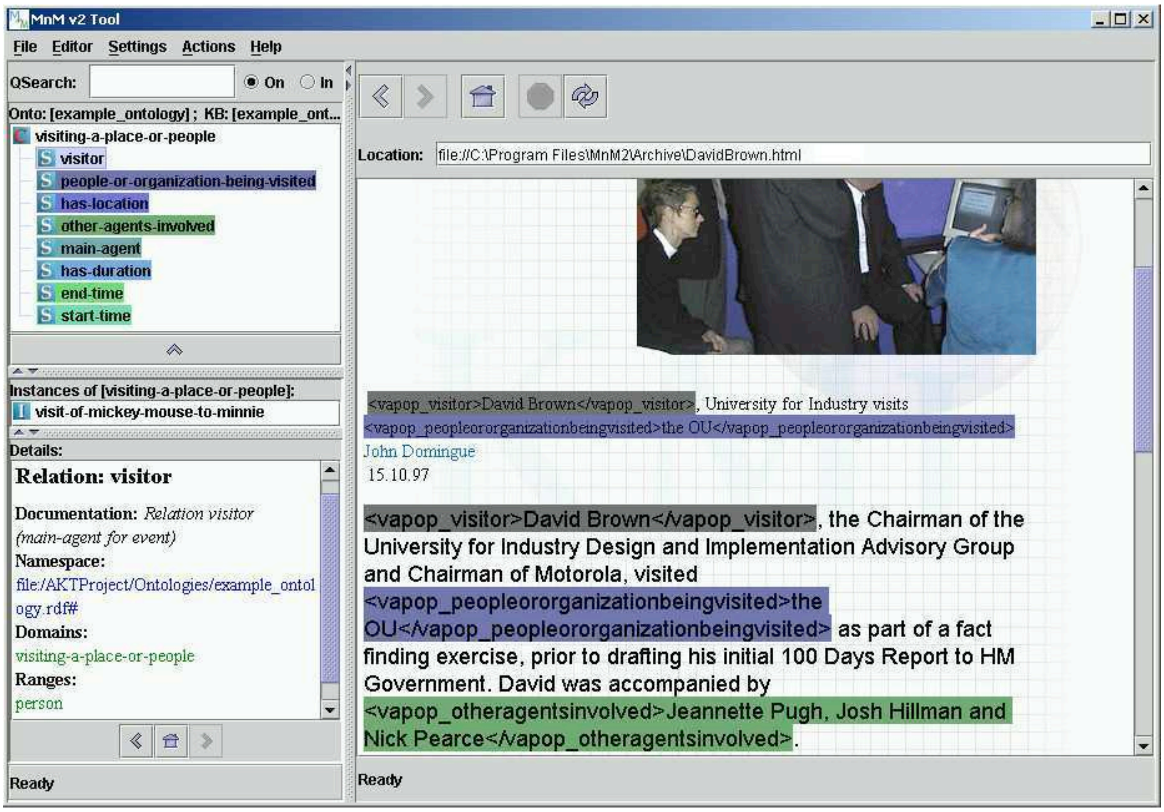


Figure 4.19: In MnM, a semi-automatic annotation mode provides support to identify instances of persons and organisations. For instance, the entity 'the OU' has been recognised as an 'organisation being visited' by the Amilcare IE engine [Ciravegna and Wilks, 2003].

CREAM In CREAM [Handschuh and Staab, 2002], an *annotation by markup* mode is provided, enabling the user to select any piece of relevant information from the page and drag and drop it to create or instantiate the selected concept instance (*researcher name*, *address*...) [6]. Text fragments are extracted from the page to foster a semi-automatic annotation: the knowledge expert agent only has to validate the proposed extracted elements.

MnM Annotation in MnM [Vargas-Vera et al., 2002] is supported by an IE engine that aims at delimiting areas of the document that can be instantiated as the concepts of an underlying ontology (including *visits*, *persons* and *organisations*.) A training corpus of annotated documents (i.e. containing instances of concepts) is created and sent to an information extraction engine that derives several rules, enabling the tool to find similar information in new documents.

For instance, sentences containing relevant information about a visit to an organisation are first identified, based on the presence of specific keywords. Within these sentences, the relevant objects (the person visiting and the organisation being visited) are extracted with a set of syntactic (syntactic cues) and semantic analysis (heuristic rules such as '*If 'something'*

[6] The Araucaria application that we have presented earlier (c.f. page 51) also enables users select text and drag and drop it, but in order to create an argument map.

is visiting a place, then this 'something' must be a person') and used to feed an instance of a class of the ontology.

When a new page is loaded, this support can be activated to provide (in addition to a fully manual annotation) a semi-automated population of the ontology (leaving the expert agent decide whether to include or not the elements which have been spotted and highlighted in the document.) Figure 4.19 gives an example of an annotation session within MnM.

In a ScholOnto scenario, provided that an IE engine could extract relevant classes of information, a similar scenario could be envisaged. An engine would extract relevant snippets of information and propose them to the annotator who could decide to keep or dismiss them. We have to assess whether this scenario is plausible or not.

Information Extraction

IE applications extract occurrences of a given fact in a document. To perform this extraction, a precise definition, or template, must be provided. A template is a structure of slots that will be filled with information, possibly picked from a list of predefined values, but more commonly extracted directly from the text, using patterns. As these facts are delimited in advance, it is clear that the domain of application has to be narrowly constrained.

Defining patterns to extract information (typically, regular expressions that will be matched against the text) makes IE systems difficult and time-consuming to build. Collections of manually built regular expressions have been progressively replaced with learning methods, in order to derive more or less automatically the rules specifying where to begin the extraction and where to end it. Such systems can for instance take as input a corpus of documents paired with their corresponding filled extracts to learn rules and fill the slots of a template.

Amilcare Amilcare is an IE system that learns rules to extract information, based on several examples annotated with the information to extract [Ciravegna and Wilks, 2003]. Starting from a collection of positive examples (i.e. containing an instance of the fact to extract) gathered in a training corpus, the system induces, for each of them, a rule specifying the preceding and following words surrounding the 'fact' to retain. This rule contains linguistic (lemma, lexical category, and case) and semantic (semantic category) constraints (c.f. table 4.3.) A generalisation process is then performed to derive a rule that covers as many positive examples as the initial rule, while being as generic as possible (by reducing the specificity of the constraints), in order to be applicable to more instances (c.f. table 4.4.)

Condition	Additional knowledge				Action
Word	Lemma	LexCat	Case	SemCat	Tag
the	the	det	low		
seminar	seminar	noun	low		
at	at	prep	low		
4		digit	low		<time>
pm		noun	low	timeid	
will	will	verb	low		

Table 4.3: An initial rule created by the AmilCare rule inducing algorithm to extract a 'time' in the phrase '*The seminar at 4pm will...*'

Condition	Additional knowledge				Action
Word	Lemma	LexCat	Case	SemCat	Tag
	at				
		digit			<time>
				timeid	

Table 4.4: A generalised rule created by the AmilCare rule inducing algorithm. Conditions on the occurrence of some specific words have been replaced by weaker ones, based on, for instance, their lexical or semantic category. Phrases such as '*The seminar at 6pm will...*' or '*The seminar at 9pm will...*' would be matched by this new rule, since the system only needs a `digit` to specify the time now, whereas it was requiring the explicit word `4` before (c.f. table 4.3.) This rule is a generalised version of the former one: it enables the capture of more instances.

Discussion

While this approach is successful to extract information that is described using a limited number of contexts (for each of these contexts has to be documented in the training corpus fed to the IE engine), we do not know how efficiently it would extract concepts and claims to fill a template similar to the one given in table 4.2.

Ontology-driven template population depend on the ability to spot elements to fill this template. This ability to spot elements relies on the fact that these elements are expressed with a limited number of contexts. In our situation, there is instead a very high variety of contexts with which a contribution, for instance, can be expressed (a contribution may even be mentioned implicitly, in words chosen not to hurt a colleague.) Providing an efficient training corpus would therefore be very difficult. Different knowledge experts (validating these annotated examples) may also disagree on what the nature of the contribution is, where its description in the text starts and where it ends. By comparison, it is much easier to discover rules to delimit a person name or a time slot, such as '4pm' in the example we have given for AmilCare.

4.5.2 Concepts extraction

While the population of a full template representing an author's argument may not be possible yet, the extraction of a few families of candidate concepts and relations may be envisaged.

Concept maps suggesters

Leake et al. design tools [Leake et al., 2003] to support analysts, based on the actual contents of their concept maps (c.f. page 46.) One of these tools is a concept suggester: it proposes new terms (candidate concepts) by crawling the Web for documents being related to the concepts currently on the map. Once these documents are found, the neighbourhoods of the existing concepts are extracted and analysed to find content words within a reasonable distance. Tentative concepts are ranked according to their frequency and proposed for consideration to the user.

In ScholOnto, related concepts could be found by proposing 'concepts' found in related documents (cited or citing documents.) However, concepts and claims have to be grounded in the current document. Therefore, any suggestion that is too 'disconnected' from the content of the current document would not be appropriate.

Index terms

Index terms can be used to identify the salient terms of a document. Similar to noun groups, they are syntactic units in which information about the noun is gathered, the noun being the central constituent determining the syntactic character of the phrase [Manning and Schütze, 1999].

LinkIT extracts index terms and sorts them by head (the noun) to assess their significance and thus identify candidate index terms, or significant topics, that can be used to represent a source document [Wacholder et al., 2001]. A study comparing different families of candidate concepts reveals that index terms are more representative of the content of a document [Wacholder et al., 2000].

Discussion

These two approaches to extract candidate concepts are valuable. Providing elements to reflect upon, such as concepts found in related documents or index terms, could bootstrap the annotation.

4.5.3 Claims extraction

Since ScholOnto discourse relations are described with natural language expressions (***uses***, ***addresses***, ***proves***...) that can be expected to be used in a consistent way (i.e. in their academic acceptation) by scholarly article authors, their instances can be spotted and proposed as a first approximation of 'author-made' ScholOnto claims: in other words, as an approximation of the areas where an author defends her position.

While verbs are not the most commonly used source of information in a document [Klavans and Kan, 1998], looking for instances of the relational type ***addresses*** is likely to identify a few positive sentences, including this sentence from [Gil and Ratnakar, 2002]:

> "*For example, if Joe Doe writes an article and publishes it claiming Henry Kissinger as the author, it should be possible to check the truth about the document's authorship.* ***Our work <u>addresses</u> a different issue on the Web of Trust regarding whether to trust the content of a Web resource depending on its source****. It seems that people reach some times informal consensus on how and when to trust what a source says.*"

This sentence clearly indicates a problem being addressed and as such, can be easily mapped to a *suggested* claim:

> {*Our work*, **addresses**, *a different issue on the Web of Trust regarding whether to trust the content of a Web resource depending on its source*}

The annotator would be free to edit it, to replace for instance the source concept *Our work* with *Trellis.* This is of course a simple approach. To make it (slightly) more robust, we could look at some selected synonyms for these verb expressions, extracted from WordNet [Miller, 1995]. WordNet is a thesaurus organising English words in a structure, with relations such as 'is a synonym of' or 'broader-than' (a brief comparison of thesauri with ontologies is given in section 2.1.2, page 14.) Not every candidate has an academic meaning, as the example in table 4.5 shows, and some filtering would have to be performed.

Discussion

We believe that an approach identifying instances of ScholOnto relations could help to identify the areas of a document in which the author defends her argument. While there would be restrictions similar to the ones we have mentioned for the IE approach (it would only match these very explicit contexts in which the authors decide to use one of these 'keywords'), it

address (v)	
1	speak to; ”He addressed the crowd outside the window” [syn: turn to]
2	give a speech to; ”The chairman addressed the board of trustees” [syn: speak]
3	put an address on (an envelope, for example) [syn: direct]
4	direct a question at someone
5	address or apply oneself to something, direct one’s efforts towards something, such as a question
6	greet, as with a prescribed form, title, or name; ”He always addresses me with ‘Sir”’; ”Call me Mister”; ”She calls him by first name” [syn: call]
7	access or locate by address
8	deal with verbally or in some form of artistic expression; ”This book deals with incest”; ”The course covered all of Western Civilization”; ”The new book treats the history of China” [syn: cover, treat, handle, plow, deal]
9	speak to someone [syn: accost, come up to]
10	adjust and aim (a golf ball) at in preparation of hitting [also: address]

Table 4.5: An overview of the senses associated to the verb ’to address’ given by WordNet 2.0. The fifth and eighth synset give several interesting synonyms to look for in a scholarly document.

could still spot potentially relevant areas. We present later (c.f. page 86) a more robust approach relying not only on a fixed list of given expressions but on a more comprehensive set of features.

4.5.4 Ontology paraphrasing

Paraphrasing is another approach that assists the population of an ontology. ACE (for Annotation Canonicalization through Expression synthesis) is an extension of Trellis supporting the incremental formalisation of statements (from natural language used by a human annotator, to a formalised version relying on the classes defined in an ontology) through a process called ontology paraphrasing [Blythe and Gil, 2004].

The ACE engine suggests replacements for the components of a free-text statement inputted by the user. Based on her answers, further replacements are generated and proposed to reach a formalised statement in which concepts and relations, defined in the Trellis language [Gil and Ratnakar, 2002] have replaced as much as possible the ones used originally in the free-text version. This process is separated in 4 steps:

- Starting from a user-defined statement such as ‘*WHU prefer forwards who play in the premier league*’ (this example is taken from [Blythe and Gil, 2004]), a term replacement step looks for any synonym found in the ontology (in this case, an ontology of football) and in a set of hand-coded synonyms augmented by the WordNet resource. If a substring can be replaced with a concept, the ontology is browsed to find the as-

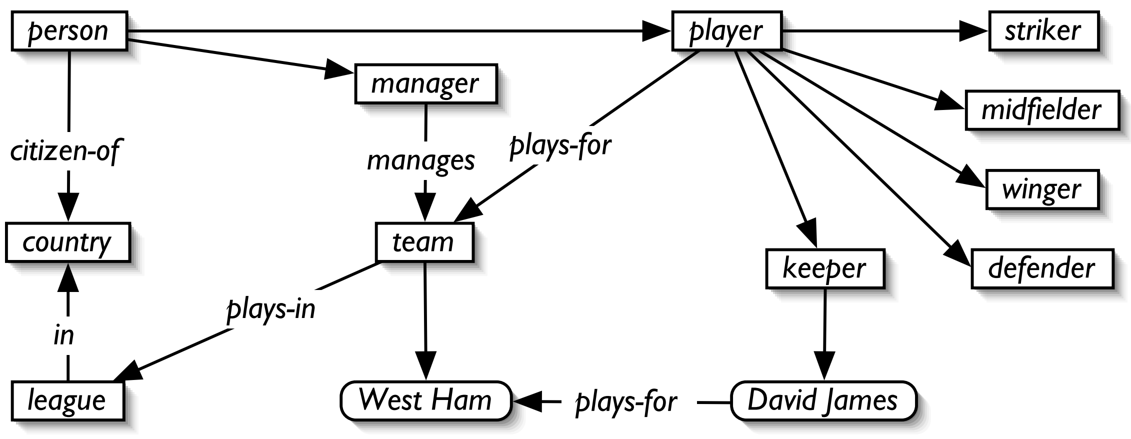


Figure 4.20: Browsing the ontology in [Blythe and Gil, 2004] enables the identification of concepts (using synonyms such as '*Striker*' for '*forward*' and '*West Ham*' for '*the hammers*' in this example) and relations (following the paths connecting the newly replaced concepts.) These can be suggested to the human annotator.

sociated relations, giving additional suggestions, which are proposed to the user who can accept or dismiss them. At the end of this step, the current statement has become '*West Ham prefer Striker who play in the Premiership*', (in which '*West Ham*' has been accepted as a replacement for '*WHU*', '*Striker*' has replaced '*forward*' ('*Striker*' is hand-coded synonym of '*forward*') and '*Premiership*' has replaced '*premier league*.')

- A sentence parsing module is also used to improve term substitutions. This module can remove words tagged as determiners, enabling substitution on groups of words. For instance, '*strikers*' and '*two strikers*' are considered equivalent in the sentence '*They want two strikers*.'

- An expression composer is then used to propose additional suggestions based on the concepts that have replaced text strings. This can result in a sentence '*West Ham prefer Striker plays-for a team that competes-in Premiership.*' Because the ontology knows that a '*Striker*' is a '*Player*', the expression composer module can look for relations associated to '*Player*': it finds a '*plays-for*' relation between a '*Player*' and a '*Team*', prompting further replacements to the user.

- The user can also define terms in the ontology at any moment (for instance, for a new team or a new player.)

User statements are thus progressively paraphrased and replaced with versions that can be matched against the domain ontology.

4.5.5 Discussion

This section has presented approaches that can be used to extract elements and propose them to the annotator. While the usefulness of IE and ontology-driven template population for our settings may be discussed, 'simpler' approaches such as the identification of candidate concepts and relations are worthy of consideration (and are most of all relatively straightforward to implement.)

Ontology paraphrasing is a potentially very interesting approach, as it lets annotators express themselves in the language they know best: their own natural language. By assisting the translation of their statements into a formal language, the machine removes the difficulties associated to formalisation. This approach relies on the presence of a domain model articulating its different concepts.

We do not have such a model (at least, not at present), for the formality is in the discourse relations only and the concepts are free in their expression. However, we have a collection of concepts connected in claims. A 'term replacement' step could look for WordNet synonyms of a concept submitted to the repository (for simple concepts.) A synonym could be proposed to the annotator. If she accepted it, the engine could then follow the claims defined between this synonym and other concepts, providing more suggestions. For instance, assuming that an annotator submitted a concept 'taxonomy', a concept suggester could propose 'ontology' and 'a specification of the concepts of a domain', provided that (i) 'taxonomy' was found to be a synonym of 'ontology' in WordNet or in a manually-created lexicon; and (ii) that a claim {*ontology*, ***is an example of***, *a specification of the concepts of a domain*} were defined. It could also help annotators reuse existing concepts, thus limiting the duplication of concepts expressing similar ideas and contributing to the creation of committed and fine-grained literature models (c.f. section 2.2.2, page 21.)

4.6 Coherent passages extraction

While the previous section has focussed on the identification of 'atomic' elements from the text, we now consider approaches to spot relevant passages [7] in a document. These two approaches can be combined, to extract first a relevant passage, and then focus on its content to extract atomic elements.

We consider four approaches to delimit coherent passages [8], in which coherence is as-

[7] We define passages as sets of one or several sentences.

[8] A passage is deemed coherent if it constitutes a self-contained unit, therefore if it does not contain any gap.

sessed with the structure of the (scholarly) document, with surface-based features, with a topical coherence measure and finally with a rhetorical coherence measure. Identifying passages is a way to reduce the amount of information faced by our end-users.

4.6.1 Structure-based relevance

The genre of the scholarly document influences annotators (consciously or not) in their concepts and claims formulation task. At a syntactic level, a scholarly document can be composed of a preface and a set of sections; a preface can be composed of a title, an author, an affiliation and an abstract; a section can be composed of a heading and a body [Sumita et al., 1993]. At a semantic level, the macrostructure of a scholarly document (such as the Introduction - Procedure - Discussion structure model [Hill 1982], cited in [Swales, 1990], page 133) indicates to readers/annotators (assuming they will be scholars) what to expect from each of its components.

We expect readers to make use of this structure in their concepts and claims formulation process in order to narrow a document to a set of passages [9]. Bishop reports how students and researchers approach scholarly documents by breaking them into components, such as abstract, figures and tables, bibliographies and author affiliations [Bishop, 1998]. Bishop notes how the '*nature of disaggregation (into components) is complex*', mentioning for instance how a component can be used for two purposes (an author's name used to assess the quality of a document or its relevance to a discipline) and how a component can be used in different ways to fit a similar purpose (figures used as a synopsis of a paper or as an assessment of whether the paper would hold one's interest.)

She also conducts interviews about the use of components to identify, read, and incorporate a document into one's work. Her four interviewees successively read the abstract and introduction (in order to ascertain key points), skim article headings (for a synopsis of the work), look for presentational features such as bulleted lists, summary statements, definition and pictures (to get the key points of a paper), focus on any particular section (that seems interesting), read the conclusion (to check one's understanding and identify other key points) and skim the references section. Readers (familiar with the genre) do make use of the specificities of these documents. We must be careful to preserve them as much as possible.

[9] We have shown already how annotators make use of signals such as citations or verbs to attribute reported work to external researchers for instance, c.f. chapter 3.

4.6.2 Surface-based relevance

Finding relevant passages is one of the main approaches in the field of automatic document summarisation [Sparck Jones, 1993]. Summaries are created by extracting and pasting sentences together (another approach relies on the compilation of sentences matching roles defined in a template, as we shall see later in [Paice and Jones, 1993].)

In this first approach, relevant sentences are identified with surface-based features. In Luhn's method, the score of a sentence [10] is based on its words (dismissing common words such as 'the' or 'and'), the words appearing frequently being given a higher score [Luhn, 1959]. [Edmundson, 1969] extends this approach by considering three additional sources of surface-based aspects to assess the score of each sentence:

- the *cue* method hypothesises that the importance of a sentence is influenced by 'strongly-indicative' cue words such as 'significant', 'impossible' or 'hardly.' A dictionary of cue words is used, and sentences containing them are given a higher score.
- the *key* method is similar to Luhn's, but it considers only the words that are not already part of the *cue* dictionary.
- the *title* method compiles for each document the words in the title, subtitle and headings of the document. Weights are given for each of these words, using assumptions such as 'the words an author has selected for the title of her document are more important than the words picked for the headings.' Sentences containing these words are given a higher score.
- the *location* method relies on two assumptions: sentences occurring under certain headings are more relevant than some others (sentences belonging to a section 'Results' may be more interesting than the ones belonging to a 'background information' section), and sentences appearing very early or very late in the document are more likely to discuss the topic of the document.

Edmundson reports on several cycles to determine the best weight (or relative importance) to give to each of these methods and finds out that the combination of the four methods performs better than Luhn's original method. Deciding on the best combination of these features (how important each of them should be) is not straightforward (as the cycles of the trial process have showed) and depends on the documents considered and on the type of

[10] The sentences with the higher scores are retained for inclusion in the generated summary.

summary expected. A more recent approach using a machine learning technique to adjust these weights automatically can be found in [Kupiec et al., 1995].

A trainable document summariser

Kupiec et al. implement a trainable document summariser to identify automatically (after a training period) the sentences of a document which are typically chosen by authors for their summaries [Kupiec et al., 1995] [11]. During a training phase, positive (i.e. reused in the author-written summary) and negative sentences of every document of a training corpus are described in terms of a set of features (presented below.) A set of correlations between feature values and their assigned category (i.e. 'positive' (for positive sentences) and 'negative' (for negative sentences) is obtained by a classifier algorithm [Sebastiani, 2002]. Using these correlations (or probabilities), a new sentence in a new, unseen, document can be classified as 'positive', based on its values for each of the features.

Description of the sentences of the training corpus The following features are considered relevant to decide whether a sentence is 'positive' or 'negative' (features return 'true or false', or an integer):

- Sentence length cut-off feature: short sentences are usually not included in a summary. This feature returns *true* for sentences containing more than a fixed number of words.
- Fixed-phrase feature: sentences containing an instance of a list of 26 indicator phrases (including expressions such as *In conclusion*, or occurring immediately after a section heading such as *Conclusions*) are assigned *true.*
- Paragraph feature: sentences membership to the initial, medial or final paragraph of a section is also used to decide of sentence inclusion
- Thematic word feature: every sentence is scored according to whether its words are in a list of thematic words (composed of the most frequent words of the document.) If the sentence has a high score, this feature returns *true*.
- Uppercase word feature: the presence of proper nouns in a sentence is also a sign of relevance. It is computed according to a method similar to the thematic word feature.

[11] This assumes that summaries are only composed of (whole or parts of) sentences which appear in the body of the document.

Learning A naive Bayes classifier is used to learn the correlation between the values of the features and the status of a sentence ('positive' or 'negative.') In this classifier, the probability for a sentence s to belong to a summary S [12], provided that it is described by the n features F_1, F_2,... F_n is given by the following formula:

$$P(s \in S|F_1, F_2, \dots F_n) = \frac{P(F_1, F_2, \dots F_n|s \in S)}{P(F_1, F_2, \dots F_n)} \tag{4.1}$$

An important assumption that is made in this classifier is that all the features are independent. This means that the value of a feature is not influenced by the value of any other feature. This is not necessarily true: there is a relation between the 'sentence length' and 'fixed-phrase' features for instance, as the probability that a sentence contains a fixed-phrase is much higher if it is longer. This naive Bayes assumption is generally applied however, as it simplifies greatly the computation of the probability estimates.

With the assumption, the formula becomes:

$$P(s \in S|F_1, F_2, \dots F_n) = \frac{\prod_{i=1}^{n} P(F_i|s \in S)P(s \in S)}{\prod_{i=1}^{n} P(F_i)} \tag{4.2}$$

with $P(s \in S)$, $P(F_i|s \in S)$ and $P(F_i)$ being estimated from the training set.

Classification The sentence s receives a score that can be used to assess whether it should be included in a summary.

Features impact Kupiec et al. report different experiments with different sets of features. The best combination found uses the *paragraph*, *fixed-phrase* and *sentence-length* features.

Discussion

This approach gives us several relatively simple features to consider to assess the relevance of a sentence (with relevance based on how 'worthy' to be included in a summary it is.) While this is different from our own settings (we do not want annotators to build their best possible model, but only a model that expresses their ideas in a satisfying manner), we can however assume that what is expected to be accepted as a '*condensation of the original document*' [Sparck Jones, 1993] can also provide valuable information for annotators skimming a document and looking for potentially relevant passages.

[12] i.e. to belong to the 'positive' category.

Pattern
effect on PROPERTY/SPECIES of SPECIES/AGENT
effect of INFLUENCE on PROPERTY of/in SPECIES
effect on PROPERTY/SPECIES
response of PROPERTY of/in SPECIES to INFLUENCE
yield(s) of SPECIES
SPECIES yield(s)
PEST is a ? pest of SPECIES

Table 4.6: Paice and Jones's contextual patterns are matched against the text and candidate strings are proposed as instances of concepts PROPERTY/SPECIES, SPECIES/AGENT or INFLUENCE.

4.6.3 Knowledge-based relevance

Another class of approaches to create summaries uses knowledge bases and templates [Paice and Jones, 1993]. It shares the same characteristics as IE techniques: restrictions to a limited and well-defined domain, and a need for domain-specific knowledge engineering.

Paice and Jones's automatic summarisation approach uses the regularities found in sentences describing concepts ('species', 'cultivar') of a particular domain (crop agriculture.) When the sentence '*The effect of mildew seed treatment and foliar sprays used along or in combination in 'early' and 'late' sown Golden Promise spring barley, Aberdeen, 1976 to 1982*' is encountered in a document, '*Golden Promise spring barley, Aberdeen, 1976*' is proposed as a candidate SPECIES filler because a pattern 'effect of ? in ? sown SPECIES' has been manually defined (c.f. table 4.6.)

Discussion

Our comment is similar to the one we have made for IE approaches: we do not know if a corpus of frequently used contexts (to state these concepts) can be created. A few contexts may still be modelled effectively however. A sentence such as '*The paper introduces an approach that organises retrieval results semantically*' (sentence #1 from the article we have used in our paper-based experiment, c.f. appendix A, page 241) could be used as a basis for a template 'The paper introduces an APPROACH', prompting '*an approach that organises retrieval results semantically*' as a candidate concept filler with a type APPROACH.

4.6.4 Topical coherence

Relevant passages can also be identified by the topics they discuss. While the segmentation in sections and paragraphs is helpful, it may not always be provided, and more importantly, it may not always reflect exactly the topical organisation of the document. For instance, a topic may be discussed in multiple 'units' (sections or paragraphs) which could advantageously be presented together to a reader. Conversely, a single unit may contain different topics and may therefore advantageously be split into smaller ones. Topical parsing goes beyond the initial structuration and proposes an alternative segmentation reflecting more accurately topical changes. Its goal is to identify a set of specific locations in the text where a discontinuity in the content of a document is detected. Discontinuities are good breaking points, as they imply that the segments they delimit a unique sub-topic (in other terms, that they are topically coherent.)

TextTiling

TextTiling is an algorithm that breaks a text into a set of multi-paragraph units reflecting its sub-topic structure [Hearst, 1994]. If a term appears several times in a set of adjacent paragraphs, it is deemed to be a good indicator of topicality; in contrast, a term which is distributed evenly along the document does not qualify (since a (sub-)topic is by definition not discussed throughout the document.) Locality is thus defined according to the terms' distribution: the vocabulary shifts in the text delimit units containing coherent sub-discussions.

The text is first subdivided into fixed-size token sequences, separated by gaps. The length of these sequences is adjustable, but is usually 3-5 sentences long (with sentences being blocks of 20 word tokens, rather than the syntactic unit.) Similarity between two blocks is calculated with the following cosine measure [13], in which $b1$ and $b2$ are two adjacent text blocks, t ranges over the n terms in the document, and $w_{t,b1}$ is a variant of the tf.idf weight (i.e. an estimation of the importance of term t in the block, compared with its importance in the document) assigned to term t in block $b1$:

$$cos(b1, b2) = \frac{\sum_{t=1}^{n} w_{t,b1} w_{t,b2}}{\sqrt{\sum_{t=1}^{n} w_{t,b1}^2 \sum_{t=1}^{n} w_{t,b2}^2}} \tag{4.3}$$

[13] In the Vector space model, documents are represented as vectors, with each word of the document being assigned a dimension and a weight in the vector space [Salton et al., 1996], [Manning and Schütze, 1999, page 539].

To assign these weights, or in other words to weigh terms, multiple approaches are possible. The simplest one is to count how many times a word appears in a document and consider this as the term weight.

The tf/idf (for term frequency $\times$ inverse document frequency) measure provides a more robust approach by comparing the number of times a term appears in the current document, with the number of times the same term appears in the remainder of the collection of documents considered. The idea is that terms that appear frequently in the current document, but not frequently in the remainder of the collection, must be good descriptors of the content of this document (by comparison, a term *the* may appear equally frequently in both the current document and the remainder of the collection, making it a bad descriptor of the content of this document.)

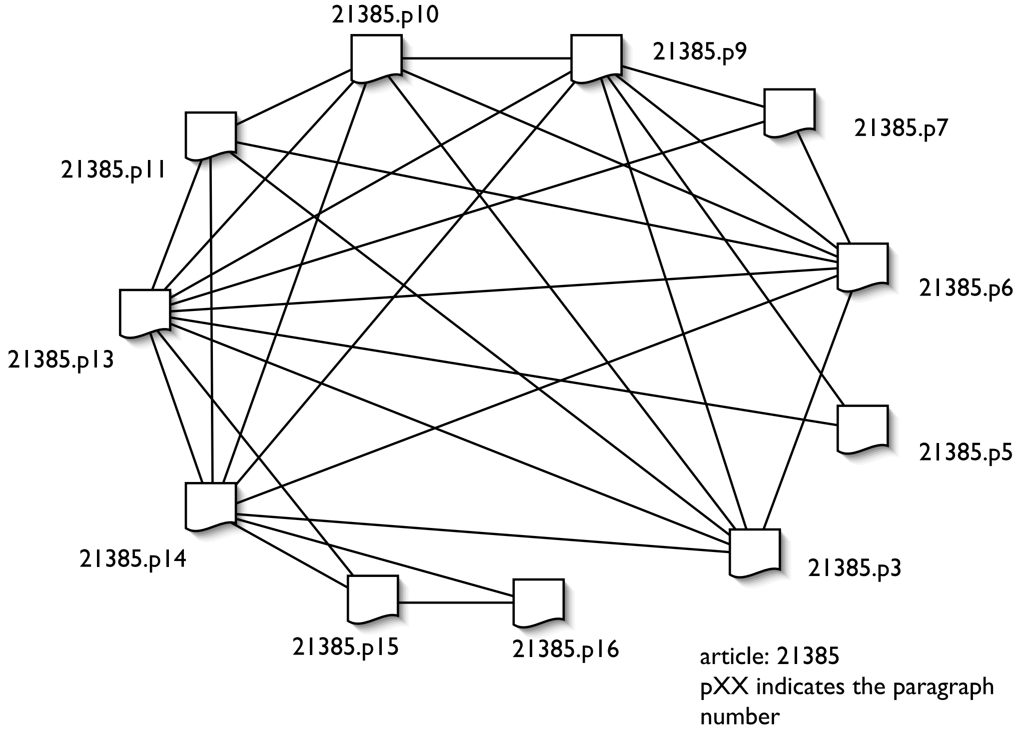


Figure 4.21: Drawing links between paragraphs for which similarity is higher than a given threshold (0.20 here) yields a map of their inter-relationships. The most 'central' paragraphs are the most connected ones (p3, p6, p9, p13 and p14.)

High similarity scores for blocks $b1$ and $b2$ indicate two properties: (1) they share terms in common; and (2) these terms are relatively rare in the block, with respect to the remainder of the document (because of the tf.idf factor), and thus do not indicate a shift on what is being discussed.

Introductory material and text themes

In [Salton et al., 1996], text units (paragraphs) are also described using the Vector Space model. A similarity measure is computed to generate pair-wise links between units, resulting in a graph structure that represents the degree of cohesiveness between the various parts of the text (c.f. figure 4.21.)

Two elements can be recognised from such a graph:

- Text units that are well linked internally but disconnected are recognised as text segments. They are usually associated to *introductory material* or to the exposition or development of a text. Paragraphs 5 and 16 in figure 4.21 are good candidates for introductory material.

- Semantically homogeneous text pieces (self-contained, with only a few connections to external units) are classified as *text themes*. Paragraphs 9, 13 and 14 appear to be text themes.

Discussion

Extracting coherent units may help annotators, by breaking a document not only in terms of its structure, but in terms of the content of this structure. It would give an additional fragmentation of the document and additional passages to focus on.

4.6.5 Rhetorical coherence

The final approach we consider to identify relevant passages relies on the rhetorical role sentences are playing. We have already mentioned that scholarly documents exhibit strong characteristics: they are chiselled to convince us of the relevance and validity of the research they report. Since a ScholOnto annotation is likely to contain annotators' reactions to these arguments, the ability to spot areas in which an author defends her position seems very important.

Roots: the Create A Research Space model

Swales' account of rhetorical moves in a corpus of introductions of physics research articles shows that (within this field, at least) authors present (and justify the existence of) their work by addressing three different needs: the need to re-establish the importance of the research field; the need to identify a niche where the contribution will be accepted; and the need to occupy and defend this niche [Swales, 1990]. His Create A Research Space (CARS) model presents for each of these needs (or moves) the different strategical steps which are involved (c.f. table 4.7.)

- Establishing a territory: to recall himself to the discourse community, an author may appeal to its members, asking them to accept that the reported research is part of a lively, significant or well established area ("*Recently, there have been a spate of interest in how to…*") Another strategy is to start with a topic generalisation, which is assumed to be more neutral, by making a statement about knowledge or practice, or statements about phenomena ("*There is now much evidence to support the hypothesis that…*", "*The … properties of … are still not completely understood.*") The final alternative to establish a territory is to review one or several relevant items of previous research.

move 1	**establishing a territory**
step 1	claiming centrality
and/or step 2	making topic generalisations
and/or step 3	reviewing items of previous research
move 2	**establishing a niche**
step 1A	counter-claiming
or step 1B	indicating a gap
or step 1C	question-raising
or step 1D	continuing a tradition
move 3	**occupying the niche**
step 1A	outlining purposes
or step 1B	announcing present research
step 2	announcing principal findings
step 3	indicating RA structure

Table 4.7: The CARS model provides an account of the strategies used by authors to convince their colleagues of the relevance of their work.

- To establish a niche, an author can use one of the following strategies, from 'risky' alternatives such as counter-claiming to gentler solutions such as raising a question ("*A question remains whether...*") or continuing a tradition. Alternatively, she can indicate a gap to signal limitations in previous work ("*However, the previously mentioned methods suffer from some limitations...*")

- Occupying the niche: to finally transform the delimited niche in a valid research space, an author can indicate her main purpose or describe the main features of her research ("*The aim of the present paper is to give...*", "*This study was designed to evaluate...*") She can also give a summary of her main findings, and finally an outline of the structure of the document ("*The remainder of this paper is divided into five sections. Section II describes...*")

Although the CARS model focusses on introductions, it provides evidence that certain areas of a research article (the moves and steps of CARS are illustrated with sentences extracted from the corpus) are more likely to contain valuable information, and moreover that the linguistic cues authors use to convey their message can be recognised by readers. '*Announcement of present research*' (move 3, step 1B, in table 4.7) or '*Review of previous items of research*' (move 1, step 3) can be of particular interest, as these are elements which are likely to be defended by authors and debated by readers, and thus modelled into ScholOnto concepts and claims.

Category	Function	*Examples*
Textual metadiscourse:		
Logical connectives	express semantic relation between main clauses	*in addition, but, therefore, thus, and*
Frame markers	explicitly refer to discourse acts or text stages	*finally, to repeat, Our aim here, we try*
Endophoric markers	refer to information in other parts of the text	*noted above, see Fig. 1, table2, below*
Evidentials	refer to source of information from other texts	*according to X, Y, 1990, Z states*
Code glosses	help readers grasp meaning of ideational material	*namely, eg, in other words, such as*
Interpersonal metadiscourse:		
Hedges	withhold writer's full commitment to statements	*might, perhaps, it is possible, about*
Emphatics	emphasise force or writer's certainty in message	*in fact, definitely, It is clear, obvious*
Attitude markers	express writer's attitude to propositional content	*surprisingly, I agree, X claims*
Relational markers	explicitly refer to or build relationship with reader	*frankly, note that, you can see*
Person makers	explicit reference to author(s)	*I, we, my, mine, our*

Table 4.8: Hyland's taxonomy of meta-discourse constructs organises them according to the role they play in the author's strategy.

Roots: Meta-Discourse

Hyland coins the term 'meta-discourse' [Hyland, 1998] to refer to these linguistic cues used by authors to organise their discourse, signal their attitude and influence readers' understanding of the text. Meta-discourse refers to '*the non-propositional aspects of discourse that help to organise prose as a coherent text*': it is adapted to a particular context to ensure that knowledge is passed on to the reader and interpreted accordingly. Hyland analyses a corpus of 28 research articles from four different disciplines and creates the taxonomy of meta-discourse constructs presented in table 4.8.

These particular expressions can also be used to identify areas in which an author defends her argument; the two approaches we present below make use of meta-discourse expressions to identify rhetorically-coherent passages.

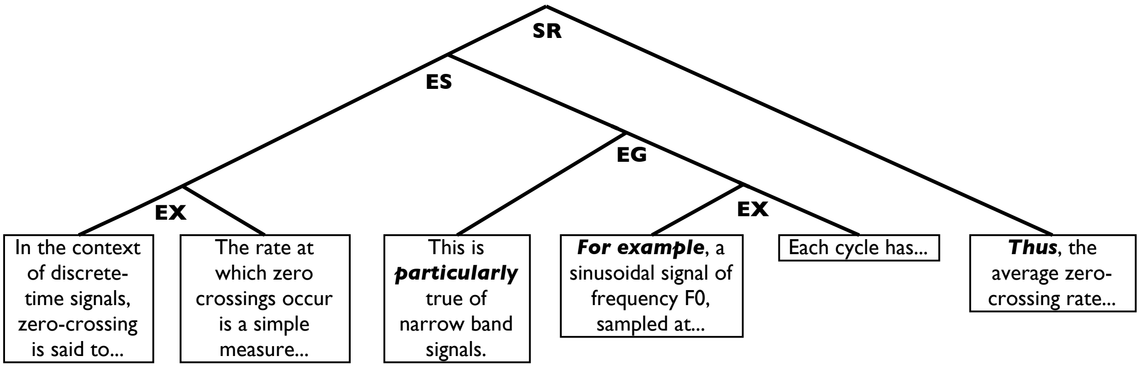


Figure 4.22: A discourse tree generated for a paragraph in Brevidoc. The matched **bold** expressions are used to assign a rhetorical relation between a sentence and its predecessor.

Discourse trees

In Brevidoc, discourse trees are used to tailor automatically-generated summaries [Miike et al., 1994], based on the Rhetorical Structure Theory [14]. Rhetorical relations are assigned (between a sentence and its predecessor) using a list of regular expressions; Miike et al. report on the use of a corpus of 1350 expressions, including connectives, anaphoric expressions and idiomatic expressions, some of which are given in table 4.9 with their corresponding rhetorical relation. This process results in the creation of a tree for each paragraph of a document, similar to the one given in figure 4.22. The process is repeated between paragraphs (with the tree root nodes of each paragraph considered as 'sentences') and results in a tree representing the full document.

Automatic approaches to derive these trees have been proposed in [Marcu, 1997].

Argumentative Zoning

Teufel and Moens propose an approach called argumentative zoning to identify the rhetorical roles played by sentences in a scholarly document and to create summaries [Teufel and Moens, 2002]. Roles are divided into two broad families: (1) the ones referring to the work being described, including BACKGROUND statements (attributed to the community in general), the AIM of the paper, its TEXTUAL structure, and other statements which do not fall in these categories (OWN); and (2) the ones indicating a relation to other researchers' work,

[14] The Rhetorical Structure Theory (RST) defines a coherent text (partly) in virtue of the relations that hold between its sub-parts (spans) [Mann and Thompson, 1987]. RST lists a catalogue of twenty-three relations, defined principally in terms of the likely effect of the two related spans on the reader. They include relations such as *background*, *elaboration*, *contrast* or *evidence*. Rhetorical relations can be applied between spans as small as clauses, or between larger spans such as paragraphs or groups of paragraphs, based on the assumption that intentions to produce small pieces of text are qualitatively similar to intentions to produce larger pieces. Based on this assumption, it is possible to describe a text recursively, resulting in a hierarchical structure of relations representing the relations holding between its spans.

Relation	Code	Expressions
serial	SR	thus, then
summarization	SM	after all, in summary
reason	RS	because, the reason is,
rephrase	RF	that is, in other words
negative	NG	but, although
example	EG	for example, and so on
background	BI	previously, until now
direction	DI	here... is described, this time
parallel	PA	at the same time, in addition
supplement	SP	besides, needless to say...
especial	ES	particularly...
contrast		however, on the contrary
repetition		in other words, it is...
topic shift		well, now
first		first
extension	EX	this is, such

Table 4.9: Linguistic cues used in Brevidoc to assign rhetorical relations. The relation 'Extension' is also assigned between two spans when no expression is matched.

including a mention of a CONTRAST with a previous work, a use of previous work as a BASIS in the work being reported, or an OTHER relation.

Rhetorical roles are assigned to (possibly adjacent) sentences, resulting in the identification of areas that are rhetorically coherent (i.e. playing a single rhetorical role.) The assignment of a role to a sentence, instead of relying on specific rules, is achieved via a sentence classification approach similar to the one we have presented earlier [Kupiec et al., 1995, Sebastiani, 2002]. By providing a sufficient amount of training data (composed of sentences and their estimated category), the goal is to find, for each sentence in a new document, its most probable category.

This training is created by three annotators who use a set of guidelines in specific situations. Because of this human intervention (whereas Kupiec et al.'s relevance is based on the abstract-similarity only, which can be performed automatically with string-matching functions), it is necessary to ensure that the human experts' analysis is both stable (i.e., an annotator classifies a sentence in the same category at different times) and reproducible (i.e., different annotators classify a given sentence in the same category) [Carletta, 1996].

Table 4.10 lists and explains the different features used to assign a role (OWN, BACKGROUND, AIM, TEXTUAL, BASIS, CONTRAST or OTHER) to a sentence. The list contains syntactic (as in [Kupiec et al., 1995]) and rhetorical features. A naive Bayes classifier is trained. It outputs sentences classified with their most likely rhetorical role (c.f. table 4.11.)

Teufel and Moens report on 'Location' being the single best feature to discriminate two

Type	Name	Feature description
Absolute location	Loc	Position of sentence in relation to 10 segments. sentences in the beginning are more likely to describe other researchers' work (like the identification of a shortcoming, c.f. move 2, step 1B in Swales [1990]' taxonomy), whereas the end of a document may contain limitations of the own work.
Explicit structure	Section struct	Relative and absolute position of sentence within section (e.g., first sentence in section or somewhere in second third) sentences at the beginning of a section are likely to summarise its contents. In a similar way, sentences at the peripheral of a paragraph summarise its contents. The headlines of the section a sentence belongs to is also a good indicator of its role: a set of 15 headlines are considered, including solution, related work,
	Para Struct	Relative position of sentence within a paragraph
	Headline	Type of headline of current section
Sentence Length	Length	Is the sentence longer than a certain threshold, measured in words? short sentences are typically not included in a summary
Content features	Tf/idf	Does the sentence contain "significant terms" as determined by the tf/idf measure? Sentences containing words from the title, and the ones containing words appearing frequently in the document, but not in the collection, might bring a good characterisation of its contents. These sentences are promoted.
	Title	Does the sentence contain words also occurring in the title or headlines?
Verb syntax	Voice	Voice (of first finite verb in sentence) POS tags are used to detect the tense and the voice of the verb. The phenomenon of hedging [Hyland, 1998] can be recognised through the detection of a modal auxiliary.
	Tense	Tense (of first finite verb in sentence)
	Modal	Is the first finite verb modified by modal auxiliary?
Citations	Cit	Does the sentence contain the name of an author contained in the reference list? If it contains a Citation, is it a self citation? Whereabouts in the sentence does the citation occur? the presence of a citation in a sentence is a strong indicator of membership to the OTHER category. However, if the citation is a self-citation, statements of continuation are more frequent than criticisms.
Formulaic	Formulaic	A list of indicator phrases is modelled by regular expressions and classified in 18 semantic classes, including for example GAP_INTRODUCTION (*to our knowledge*), COMPARISON (*when compare to our*), CONTINUATION (*following the argument in*) or POSITIVE_ADJECTIVE (*appealing*.)
Agentivity	Agent	Type of Agent I The agent of a sentence (US, THEM, OUR...)
	SegAgent	Type of Agent
	Action	Type of Action, with or without Negation. However, some statements (like the fact that a work is based on some previous work for example) do not follow a limited set of expressions, and do not look similar on surface (that was the reason for rejecting Information Extraction approaches.) The approach here is to 'break' the recognition of such sentences in two steps: recognise the agent type and the action type. Lexicons for these two categories are provided that they list the different roles agents can play (including US (*we*), THEM (*his approach*) and GENERAL (*traditional methods*) among others) and their different actions (ARGUMENTATION (*we argue against a model of*), BETTER_SOLUTION (*our system outperforms*)...) By combining these two elements, many more indicator phrases are detected, resulting in an improvement of the classification.
History	History	Previous category this feature keeps the value assigned to the previous sentence, to capture prototypical sequences of categories (like AIM following CONTRAST.)

Table 4.10: Overview of the features set considered in [Teufel and Moens, 2002] to estimate the rhetorical status of a sentence.

AIM: Our research addresses some of the same questions and uses similar raw data, but we investigate how to factor word association tendencies into associations of words to certain hidden senses classes and associations between the classes themselves. ...
BACKGROUND: Methods for automatically classifying words according to their contexts of use have both scientific and practical interest. ...
OWN (DETAILS OF SOLUTION): The first stage of an iteration is a maximum likelihood, or minimum distortion, estimation of the cluster centroids given fixed membership probabilities. ...
CONTRAST WITH OTHER APPROACHES/WEAKNESSES OF OTHER APPROACHES: His notion of similarity seems to agree with our intuitions in many cases, but it is not clear how it can be used directly to construct word classes and corresponding models of association. ...
BASIS (IMPORTED SOLUTION): The combined entropy maximisation entropy [sic] and distortion minimization is carried out by a two-stage iterative process similar to the EM method (Dempster et al., 1977.) ...

Table 4.11: Teufel and Moens's classifier outputs a set of rhetorically-coherent passages.

sentences. Good results are also given by the 'SegAgent', 'Citations', 'Headlines', 'Agent' and 'Formulaic' features. This emphasises the need to consider the specificities of the documents to properly assess the role played by each sentence.

4.6.6 Discussion

The different approaches presented in this section are all potentially relevant, as their goal is to reduce the amount of information being presented to the annotator. The ability to spot areas where an author states and defends her niche [Swales, 1990] would help an annotator position herself with respect to the research reported and model her interpretation if she deems it relevant enough.

In a ScholOnto scenario, the argumentative zoning approach could also extract sentences mentioning the AIM of a scholarly document and propose them to an annotator to make her think about the problem being addressed (resulting possibly in an ***addresses*** claim.) Similarly, sentences tagged with BASIS or CONTRAST could indicate to an annotator how the author assesses her own work in relation with the existing literature, prompting the annotator to model this assessment if, once again, she deems it relevant enough.

4.7 Document relatedness

The fourth activity of our ScholOnto sense-making process (c.f. figure 4.2, page 41) is concerned with the consideration of external documents to assist the construction of literature models. We present approaches to discover related documents based on their contents (via an implicit or an explicit mention of a shared topic), to automatically add hypertext links to a corpus of scholarly documents and to identify citations in a document and optionally assess their underlying motivation.

4.7.1 Content-based relatedness

Related documents can be discovered through an analysis of the similarity of their contents.

Information retrieval

The approach proposed by Salton et al. (c.f. page 82) to assess relatedness between paragraphs can be applied at the document level: each vector describes a document and the vector similarity operation indicates which documents are the closest (in terms of the words (or topics) they share.)

Concept-based relatedness

Concept-based relatedness is another approach to identify related documents. It is more 'expensive' for it requires an explicit mention of the concepts a document contains, compared to the implicit (and thus requiring less effort) one involved in the aforementioned vector space model approach.

[Cleary and Bareiss, 1996] report on a method called *simple concept linking* to automatically create hypertext links between documents that share a concept (added manually by an indexer to each document.) Their experimentation shows that this method has a high recall rate (the proportion of document pairs created by this automatic method, compared to the pairs created manually by indexers) of 70%. It is however also noted that the method generates a high number of links, many of them being potentially not relevant.

In the COHSE system [Carr et al., 2001], a conceptual hypermedia is developed, specifying the structure of the hypertext as metadata (encoded in an ontology.) The metadata can be used to instantiate links between pages: pages 'tagged' with a similar concept (or containing concepts which are connected via a relation in the ontology) can be connected automatically. The linking process between concept-related documents is summarised in figure 4.23. The

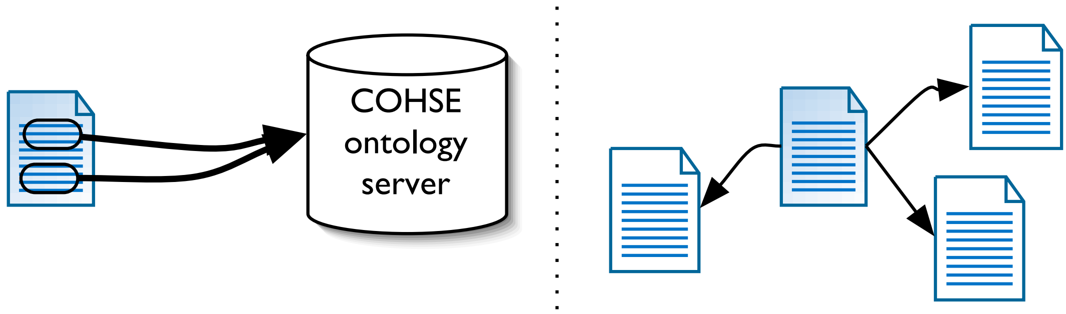


Figure 4.23: In COHSE, documents sharing a concept can be identified via the meta-data encoded in the ontology and linked from the current document.

COHSE concept has been recently developed into Magpie, which we have presented earlier: while the former provides links between documents based on the concepts they both contain (including '*narrower-than*' links from *page1* to *page2*, if *concept2* found in *page2* is found to be connected with a '*narrower-than*' relation to a *concept1* found in *page1*), the latter proposes additional services based on the relationships defined for this concept (c.f. page 64.)

The e-Scholar Knowledge Inference Model (ESKIMO) project is another example of automatically-generated links added to a collection of documents [Kampa, 2002], in order to create a consistent hypertext and issue queries such as '*On what projects has X worked?*' or '*What are the seminal papers on X?*' ESKIMO defines two ontologies: (i) a *scholarly community* ontology, defining concepts such as *organisation*, *person*, *team* and *activity*, organised with relations such as *works at* or *represented in*; and (ii) a *research themes* ontology which acts as a domain ontology, defining concepts such as *hypertext*, *theory* or *architecture*. It then acquires data (to populate these ontologies) from a repository of documents using a semi-automatic approach: journal details, person names and citations are extracted automatically from manually edited text versions of PDF conference proceedings.

Such systems provide potentially interesting services. The ability to bring in a literature model related documents sharing a concept is desirable. While ESKIMO lies at a different level from ScholOnto (it is concerned with factual knowledge that does not suffer contestation; this is to be contrasted with our interpretations that are open to debate and discussion), it is nevertheless a potentially very useful tool, as it is designed to efficiently support scholars (or in this case, 'e-Scholars') in their activities, by helping them to answer critical questions.

We imagine that a combination of ESKIMO and ScholOnto would provide a very powerful toolkit to e-Scholars.

4.7.2 Hypertext links

Explicit hypertext links can be added between documents. Trigg and Weiser propose a taxonomy of link types to connect parts of documents, as part of a hypertext system for scientific writing [Trigg and Weiser, 1986]. Links are separated into *normal* links, connecting parts of a single scientific work, and *commentary* links providing comments and critique to that work.

Normal links contain elements such as *background*, *future work*, *refutation*, *methodology*, *citation*, *data*... The *citation* link is itself subdivided into *source*, *pioneer*, *credit*, *leads* and *eponym*. Commentary links are organised in comment (*critical*, *supportive*), related work (*misrepresents*, *isSupportedBy* and *isRefutedBy*), problem-posing (*trivial* and *solved*), thesis (*trivial*, *contradict*, *dubious* and *counter-example*), argumentation (including *invalid*, *alternative*), data (*inadequate* and *ignores*) and style (*boring* and *incoherent*) links. 80 links in total are proposed, several of those can be found in the ScholOnto ontology.

Blustein evaluates the usefulness of structural links (enabling direct access to a particular section), definition links (connecting specific terms to their definitions) and semantic links (connecting related passages in the original document; such connections can tie for instance an element of the abstract to its related passage in the full text, or two passages which are deemed to be semantically coherent) [Blustein, 2000]. He reports that definition links are the most followed ones in his experiment.

4.7.3 Citations

Citations associated to a scholarly document provide a valuable list of related documents an annotator may be interested in. We can envisage to bring cited documents in the claim space automatically, using existing technologies such as CiteSeer or OpCit.

CiteSeer [Lawrence et al., 1999] uses an autonomous citation indexing approach to build networks of scholarly papers by extracting their citations and matching references to identical papers (CiteSeer also proposes related documents based on their content.) OpCit [Hitchcock et al., 2000] is a similar approach to dynamically add citation links to the documents retrieved from an archive. Pivotal points can also be proposed to filter a network of documents and retain the most important ones only [Chen, 2005].

Reasons for citing a document:
Paying homage to pioneers
Giving credit for related work
Identifying methodology, equipment, etc.
Providing background reading
Correcting one's own work
Correcting the work of others
Criticizing previous work
Substantiating claims
Alerting researchers of forthcoming work
Providing leads to poorly disseminated, poorly indexed, or uncited work
Authenticating data and classes of fact - physical constraints, etc.
Identifying original publications in which an idea or concept was discussed
Identifying the original publication describing an eponymic concept or term
Disclaiming work or ideas of others
Disputing priority claims or others

Table 4.12: Weinstock's taxonomy of reasons to motivate a citation to a work.

Motivated citation maps

Further support can be provided via an estimation of the motivation underlying a citation. Scholarly work can be cited for various reasons, and an assessment of these reasons can help to navigate a network of documents. Weinstock proposes a taxonomy of motivations to cite an external work that we present in table 4.12 [Weinstock, 1971].

Nanba and Okumura propose - in their multi-paper summarisation strategy to assist the writings of surveys of a particular domain [Nanba and Okumura, 1999] - an approach to both identify reference areas and the role played by these areas. They consider the following roles: references indicating other researchers' theories or methods used as a basis (type B), references to related works to mention a contrast or a problem (type C) and other references (type O.) Reference areas are sets of adjacent sentences spread around the sentence containing the citation signal and include the necessary information to understand the role it plays (B, C or O.)

To decide whether a sentence must be included in a reference area, 6 families of cue words (anaphors (*in this*, *on this*, *such*); negative expressions (*but*, *however*, *although*); first person pronouns (*we*, *our*, *us*, *I*); third person pronouns (*they*, *their*, *them*); adverbs (*furthermore*, *additionally*, *still*); and others (*in particular*, *follow*)) are identified in an analysis phase (in which a corpus of annotated examples of reference areas is given to an algorithm.)

Roles are then assigned using a list of cue words and heuristics such as '*If a negative expression appears at the beginning of the sentence following the sentence containing the citation signal, assign type C to the area*' are used to assign types to areas. Good candidate cue words for roles C and B include *although*, *however* and *instead of* for the former, and *used by*, *we adopt*, *we use*, *We can* or *We use* for the latter. Success ratios of 75, 78.1 and 88.5% for the automatic discovery of reference types C, B and O in their corpus are reported.

Argumentative zoning

Teufel and Moens's work on rhetorically-oriented document summarisation (c.f. page 86) classifies sentences referring to cited work in three categories BASIS, CONTRAST and OTHER, which can help understand why a particular work is cited. A new project, CitRez, is currently extending work in this direction [Teufel, 2004].

4.7.4 Discussion

These approaches can be used to provide additional resources while making sense of a scholarly document. Although the product of a ScholOnto annotation is not a citation map, proposing a list of the citations contained in a document, and an assessment of the motivation underlying them (c.f. table 4.12), can help the annotator: she can confirm, correct or refine the citation links offered by the tool if she agrees with the author, bring in a new document which the original author has forgotten or model a rebuttal if she disagrees with the assessment of the cited work.

4.8 Participatory argumentation

Sharing annotations is the final step of our ScholOnto sense-making task (c.f. figure 4.2, page 41.) By sharing their concepts, claims, and the documents on which they are based, annotators contribute to the creation of a knowledge artefact (whether they agree on its content or not) and can benefit from their peers' points of view.

In this final section, we look at possibilities to support the participatory aspect of the annotation process, through (i) the share (and retrieval) of the notes added to a document; (ii) the notification of an annotator when relevant content is added; (iii) the provision of outlines of discussions (showing the positions of multiple annotators); and (iv) the enforcement of methodologies to promote more effective participatory modelling (for example, to enforce the consultation of the arguments submitted by multiple annotators.)

4.8.1 Annotations share and retrieval

Sharing (and retrieving) notes offers possibilities to consult and re-interpret peers' arguments. In Annotea [Kahan et al., 2001], annotations associated with a document can be browsed and filtered. Different filters can be applied, based on their author, their type or the annotation server. It becomes possible to focus only on the annotations created by a particular person or on the *example* notes added. Shared annotations are also available in YAWAS [Denoue and Vignollet, 2000]: a search mechanism enables users to browse and retrieve annotations based on document topics (the documents they are associated to), annotation categories, text selection types (person's name, software...) or stances (agreement with the document, disagreement...) In CritLink [Yee, 2002], annotations can themselves be annotated to facilitate sharing and discussion. In D3E and SpeakEasy, notes added by an annotator's peers are obviously visible, being a part of the discussion. They can also be commented upon. Nodes in gIBIS and Compendium can be browsed and retrieved, according to several criteria like their type, their creator or the keywords associated to them.

Sharing and retrieving notes (concepts and claims) is an important part of our sense-making process. Annotators can benefit from their peers' knowledge and either reuse concepts and claims or take position with them.

4.8.2 Notification

Retrieving notes still requires an intervention from the user. Proactive support (performed by the system, on behalf of the annotator) offers support automatically, without user intervention.

The FXPAL Bar is an example of a recent proactive information system [Billsus et al., 2005]. The bar, sitting on the main window of the Web browser, supports users by providing two sources of recommendations: contact recommendations based on matches (in the current pages) on the names of known contacts, and content recommendations based on the similarity between the current page and documents in the database of the system. Notifications of a found contact or related document can be displayed either as a set of entries in a menu of the toolbar (although this has been found to be less efficient, as people tend to forget about its presence), or as translucent windows popping up in the bottom-right corner of the screen (an approach which works better, as it is both explicitly stating a recommendation has been made and non-intrusive.)

Three techniques to support awareness are proposed in [Brush et al., 2002] to notify the user of the additions, discussions, replies, or changes in annotations. The first of these techniques indicates in a graphical way (via multiple colours) the addition (or modification) of an annotation when the user browses the page. The second is based on subscription, a list of all the modifications being sent on a regular basis via email. The third mechanism uses a sidebar in the user's workspace: modifications are signalled in this sidebar, which is refreshed at a customisable rate.

In gIBIS, a notification system (for collaboratively-built maps) informs users of any change, such as the addition of a new node to the map.

Notification techniques in ScholOnto could also be implemented, for instance to alert an annotator when a new concept is added to a document or one of her claims is challenged.

4.8.3 Discussion outlines

Discussion outlines provide overviews of an argument and represent the multiple perspectives and positions of a community of annotators. Comments (and their author) are shown as a threaded discussion in D3E [Sumner and Buckingham Shum, 1998], with the ability to initiate several 'sub-discussions' about particular points. In SpeakEasy, a discussion object, in which the multiple perspectives on a problem are compiled, can be analysed to identify which items generate questions (comments labelled with a '*?*' type), areas of general agreement (consecutive comments categorised with '*and*' or '*i.e.* ') and areas of controversy (consecutive comments labelled with *but*.) Annotations can also be displayed as a threaded discussion in Haystack [Karger et al., 2003].

While ScholOnto does not provide a 'real' discussion space, views summarising the different stances taken on a problem could help annotators by indicating different perspectives and areas where controversy is more important.

4.8.4 Methodologies

Methodologies can finally be proposed to ensure users contribute to the argument. Enforcing participation is a first approach to make each participant contribute. In SpeakEasy, authors note a "*high level of reflection spurred because of two characteristics of the discussion space: the fact that students have to state their opinion before joining the discussion space (they* *have to formulate their idea on the initial topic of the debate on logging in), and the fact that they have to assign one of the semantic labels to their contribution.*" In CSILE [Scardamalia, 2004], quotes are represented in a different graphical style, making them stand out from the main contribution of a participant. Participants are therefore encouraged to contribute fresh ideas to the debate, instead of quoting ideas that are already present.

The SECAI methodology

A methodology to gradually share annotations can also help and may be of particular value in pedagogical contexts. In CLARE [Wan and Johnson, 1994], a complete methodology, SECAI, is provided to guide learners towards a discussion of the arguments they have built. SECAI (for 'Summarization, Evaluation, Comparison, Argumentation and Integration') defines a process model to combine the different knowledge models that have been constructed. Learners (each modelling their own document) progressively move from a private area (containing their own representation) to a common area in which representations are shared:

- The first step of the process is called *Exploration* and uses the notions of primitive nodes and canonical forms which we have introduced earlier (c.f. section 4.3.2, page 55.) It is composed of a *summarization* and an *evaluation* steps. Learners build their own model without being aware of what other learners do.

- The second step is called *Consolidation* and consists of a *comparison*, an *argumentation* and an *integration* steps. Learners become aware of their peers'

  – The first sub-step, comparison, is performed at the level of the artefact (scholarly paper) and at the three levels defined in RESRA (c.f. page 54): the canonical form level, the link primitive level and the node primitive level. A first comparison at the artefact level is performed to find out if the artefact has been classified in a similar way (for instance, as a concept paper.) A comparison at the canonical level is performed to find out if learners' representations differ from the standard one (associated to the canonical form of the paper.) Comparisons at the link primitive level involve the different interpretations (links) that have been drawn from a particular *evidence* node for instance. Finally, a comparison at the node primitive level involves the formulation of the elements identified in the artefact.

  – Once the comparison is finished, the different learners can start their argumentation about the contents of the models which are revealed. A *Critique* node can for instance be generated in a peer learner's own map to note the fact that she has just missed an important point in her representation.

– In the final argumentation step, learners debate and critique each other's models. This debate can result in a learner agreeing that one of the critiques being made is justified. In this case, this learner can model the *Critiques* being formulated by a peer. Links between multiple representations can also be created.

The SECAI process is designed to enable learners to gradually share their models. The phased approach is also designed to increase their confidence and knowledge of the formalism.

4.8.5 Discussion

ScholOnto shares similar goals: bringing annotations together, making them visible to a group of annotators and opening them up to discussion and debate. Possibilities to let annotators share their annotations and benefit from them are important. ScholOnto differs however by introducing a formalism to support large-scale annotation.

A methodology to gradually introduce peers' concepts and claims can also help annotators, giving them additional elements to react to. For instance, they could be asked to first model the elements they have retained from a document. In a second step, peers' concepts and claims could be presented, and annotators could reuse them or take position with them.

4.9 Final analysis

This section concludes our design phase. We begin with an overview of the different techniques we have presented to design a document-centric annotation environment supporting annotators in their sense-making task. We then transform this knowledge into a plan supporting our understanding of the motivation and expectations of our end-users and present our final set of requirements.

4.9.1 Overview

Our document-centric sense-making environment considers the document as a source of support, makes it visible and integrates it with the annotation process, following one of the principles defining good annotation environments identified in [Marshall, 1997]. Figure 4.24

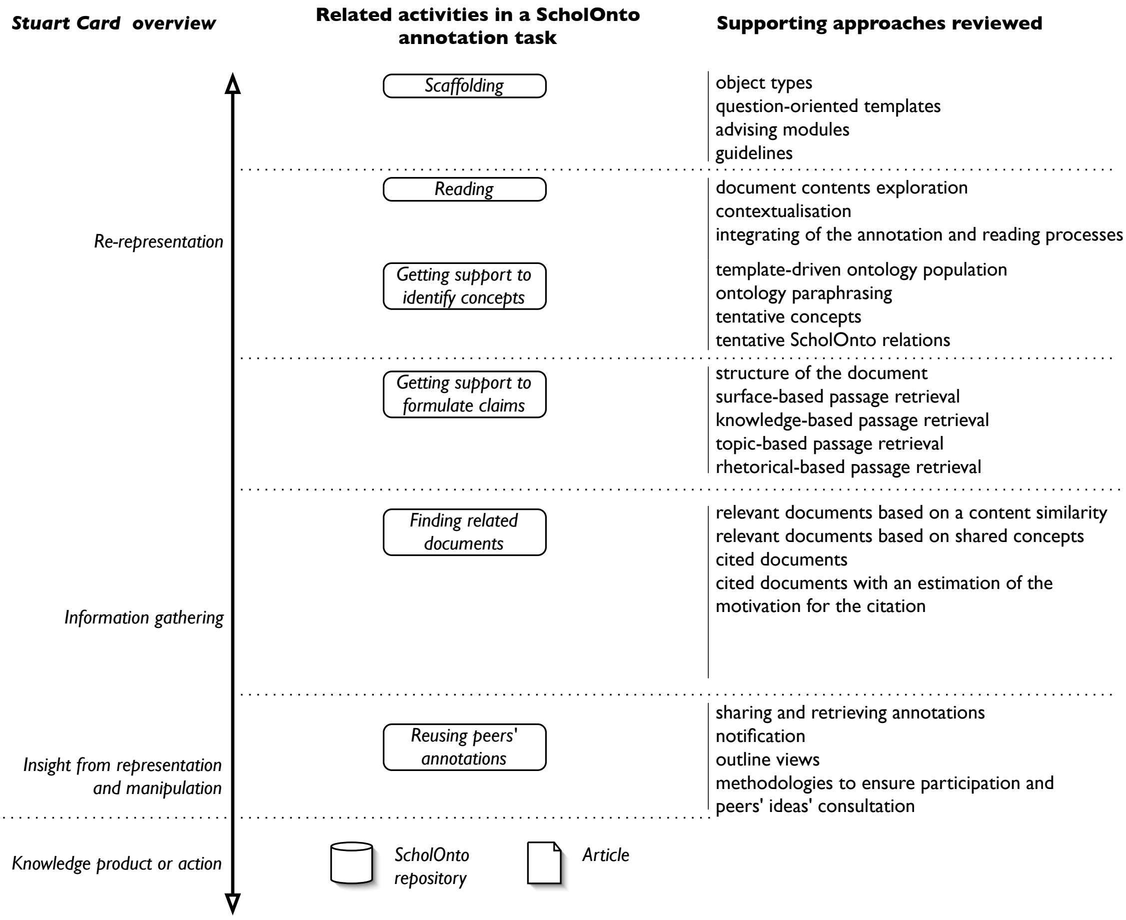


Figure 4.24: The approaches we have reviewed in this chapter can be used to support the different tasks of a sense-making process (c.f. figure 4.2, page 41.)

associates the techniques we have presented to assist each activity of the ScholOnto sense-making process.

Using the analysis of this chapter, the goals taxonomy presented in chapter 2 and the tasks and activities analysis presented in chapter 3, we can extend our preliminary activity diagram (c.f. section 3.3.1, page 36) to incorporate our improved analysis of the sense-making process (c.f. figure 4.25.)

4.9.2 Plan for a document-centric approach

We can also use this knowledge to devise a plan reflecting the motivations and expectations of our end-users, using a preliminary usage scenario.

In our active environment, the document is contextualised (c.f. page 59) with elements spotted from its content (c.f. pages 66 and 75), acting as suggestions and giving readers possible elements to react against. Additional documents are also proposed to consideration (c.f. page 90.) Scaffolds (c.f. page 52) are provided by the ScholOnto formalism, in the form of concept types and fixed relationships.

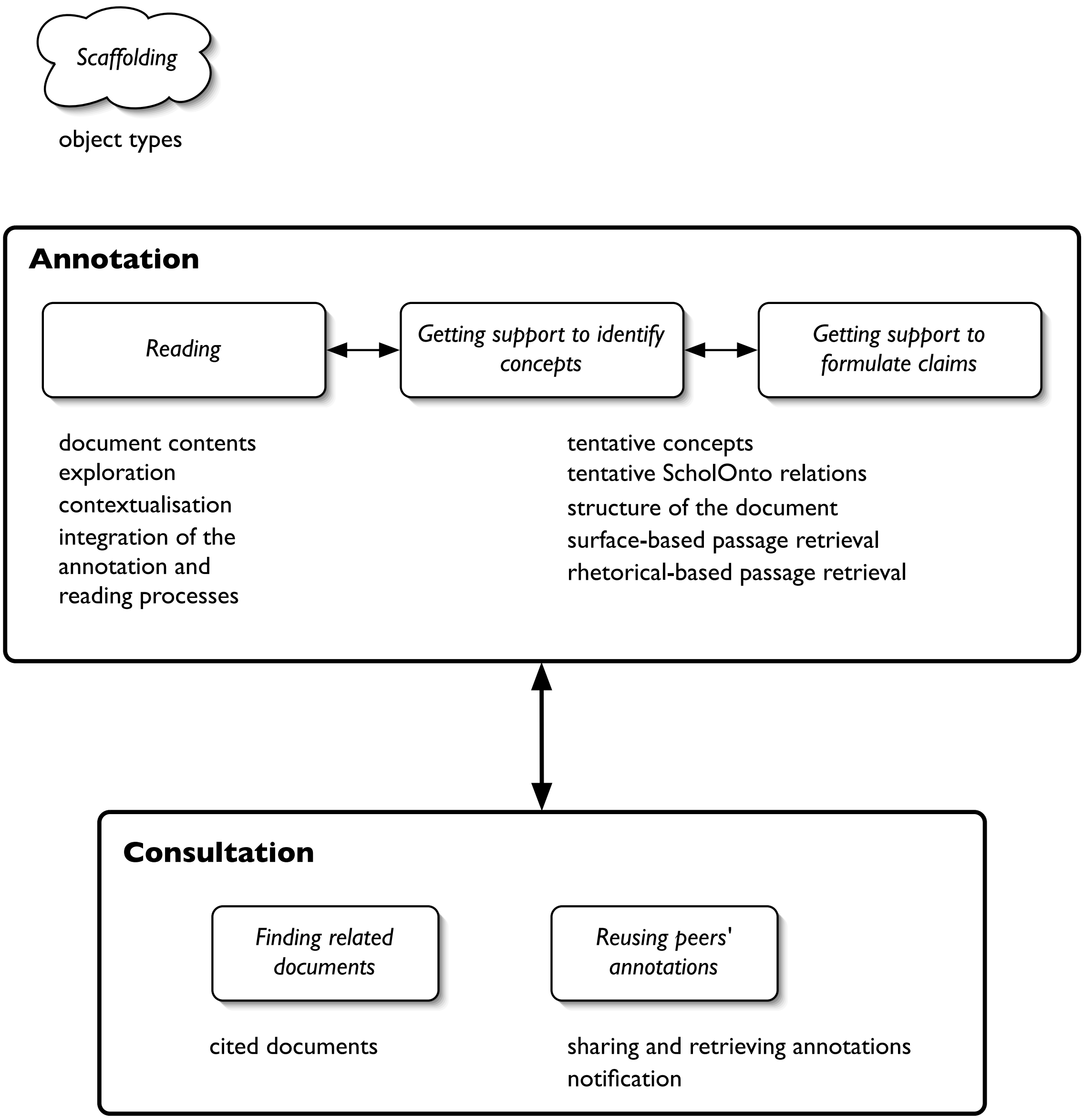


Figure 4.25: A revised version of our activity diagram (c.f. figure 3.2, page 36 for the initial version.) Each task (identified from the paper-based study and revised in this chapter) can be assigned one or several supporting approaches.

Peers' annotations are accessible to let annotators reuse or discuss a position expressed previously (c.f. page 94.) A notification mechanism informs annotators of the modifications brought to an annotation or to the arguments modelled about a particular concept they are interested in.

4.9.3 Final set of requirements

Figure 4.26 reflects our improved understanding of the transition from a document to a networked structure of arguments. We identify the following requirements for our annotation environment:

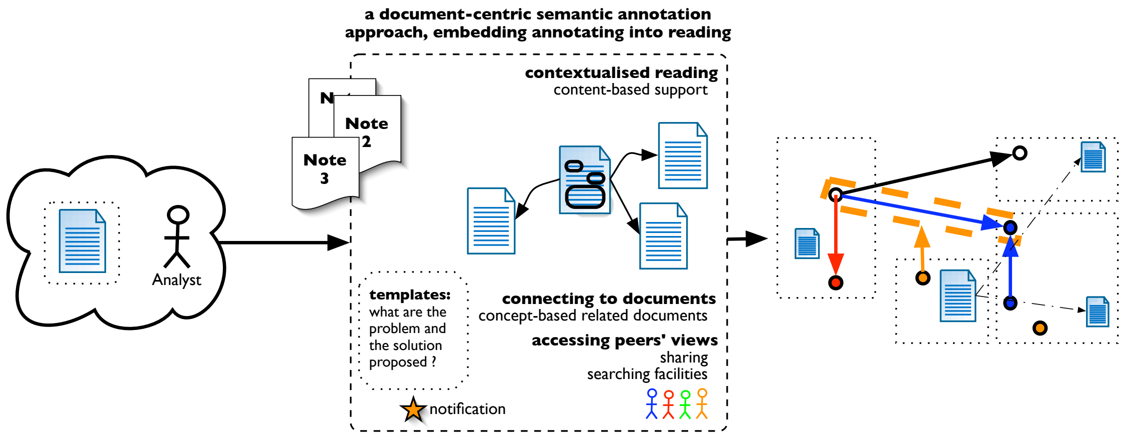


Figure 4.26: A document-centric approach to support sense-making integrates the various sources of support we have identified in this chapter, to support the transition from a scholarly document to a rich structure of concepts and claims (c.f. figure 4.1, page 40.)

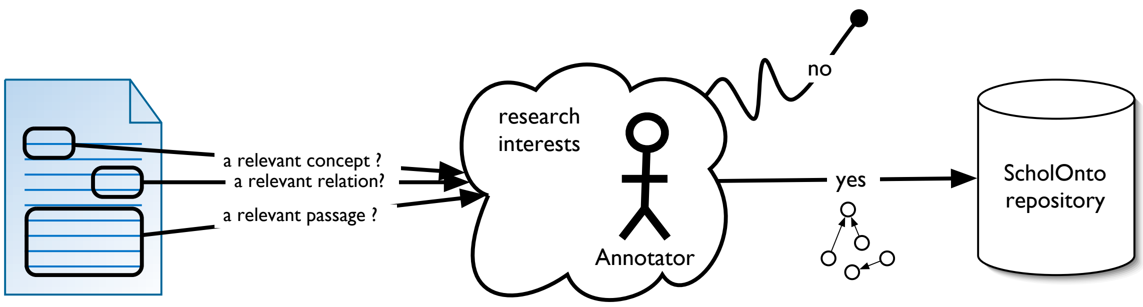


Figure 4.27: To support annotation, suggestions spotted in the text are proposed to the annotator's consideration. She is free to model them if they are important and/or relevant enough to her or to dismiss them.

- The environment must display the document and the product of its annotation side by side, to facilitate the interaction between reading and annotation (c.f. section 4.4.)

- It must provide a suite of tools offering content-based support, by spotting *potentially* valuable information and overlaying it onto the original document, to draw attention to components which may be particularly significant. These spotting filters are expected to reduce the information load created by a cognitively intensive sense-making task. We aim to propose a 'toolbox' to experiment with, that is, a set of possibilities that annotators will be able to adapt to their own needs. This toolbox shall realise the 'set of strategies' we have identified at the end of the previous chapter.

- These tools must offer additional facilities to navigate the content of a document. Figure 4.25 also contains the approaches we have selected for our initial prototype.

- The environment must provide facilities to record new semantic annotations and reuse existing ones.

- It must provide pointers to related documents, based on the concepts and claims made by an annotator's peers.

- It must rely on an open architecture. This openness is needed to ensure the environment is as flexible as possible, either to update a spotting filter with a more robust version or to add new features.

Chapter 5

The ClaimSpotter prototype

We present in this chapter our semantic annotation prototype, ClaimSpotter [1]. We detail its architecture and discuss a set of spotting components (c.f. section 5.2.) This is followed by a description of its user interface, including a brief presentation of the different phases of its evolution (c.f. section 5.3.) We give a virtual tour of the environment illustrating the interactions of a fictional user with ClaimSpotter and the dialogue instigated between them. A few technical considerations are finally given.

5.1 Architecture

Figure 5.1 presents the current architecture of ClaimSpotter. Its main part is a set of components spotting information from the original document (presented in the following section.) This architecture is open: spotting filters can be replaced by other filters, provided they use the underlying data representation format (also presented in the following section.)

5.2 Spotting filters

We present in this section the different filters available in the current prototype of ClaimSpotter that we have either developed or integrated.

5.2.1 Data representation

Spotting filters rely on a rich (structured) representation of a scholarly article to extract information. We have used an XML specification, and more precisely, the DTD developed

[1] This chapter and the following one are extended versions of [Sereno et al., 2004a,b, 2005].

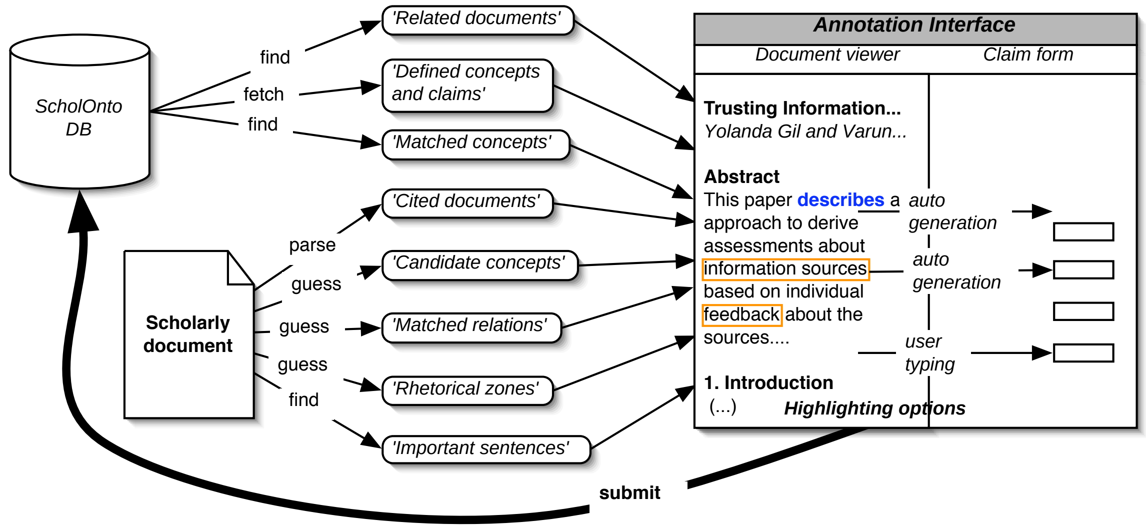


Figure 5.1: The architecture of our current annotation environment includes a suite of spotting filters and a user interface. The output of these filters can be accessed and displayed in the interface, *in situ* if applicable, via several '*highlighting options*' controlling the '*document viewer*' (left frame.) Annotations are added in the '*input form*' (right frame) from scratch or via the proposed suggestions.

by Teufel and Moens to represent scholarly documents in their rhetorical parsing application [Teufel and Moens, 2002]. This DTD represents articles as highly structured units [2]. Table B.1 (c.f. page 248) gives an extract of a scholarly article represented as a structured XML file.

5.2.2 Filters

Our focus is not the provision of state-of-the-art spotting components; we are more interested in an assessment of their impact on the annotation process, i.e. on the dialogue they foster between end-users and the interface [3]. The openness of the architecture is a direct consequence of this decision: it gives us the opportunity to replace our preliminary implementations with more robust ones.

Several filters use the Natural Language processing ToolKit, or NLTK [Bird and Loper, 2004], a comprehensive collection of tools written in the Python programming language (including tokenizers, taggers, chunkers and more) and resources (including annotated corpora to train the taggers) to facilitate the development of natural language processing systems [4].

[2] In ClaimSpotter, these files are created manually, including the segmentation into sentences and paragraphs, attribution of consequent `ids` to sentences and the detection of reference signals. This could be (at least partly) automated.

[3] We do acknowledge that the quality of the filters may directly influence the annotation process: filters that are 'more robust' may increase the confidence of the annotator in the tool for instance. For this prototype, we have focussed on preliminary versions of the filters.

[4] NLTK is available at `http://nltk.sourceforge.net`

The last five filters of our architecture take as input the repository of annotations only (i.e. concepts and claims submitted by annotators.) The last three (the 'important sentences', 'candidate concepts' and 'rhetorical areas' filters) require additional processing; they are therefore executed before the ClaimSpotter environment itself. They store their output in XML-formatted results files that are parsed when needed by ClaimSpotter.

'Matched concepts' filter

This filter spots words or groups of words matching an existing concept in the ScholOnto repository. It looks at concepts created by the current user (in any document she may have annotated before) and at concepts created by any annotator in any document. The former option provides a first step towards a personalised view of the document (assuming one's concepts represent one's areas of interest.)

Since it looks for *matched* concepts, it only highlights text strings that are matched in the content of the document.

'Matched relations' filter

ScholOnto relations (verb expressions) are spotted to draw annotators' attention to areas where an author uses one of these explicit verbs to defend her arguments. To enhance the number of matches, we look for additional synonyms in WordNet [Miller, 1995]. We have pruned several incorrect synonyms and retained the ones given in table 5.1.

A more robust filter could help by omitting instances of verbs in the ontology that do not correspond to academic usage: since every instance of these verbs is highlighted, several false positives are also retained by the filter. We hope though that this number remains quite low, as we deal with scholarly papers only: instances of 'to address' in contexts other than the ones given in synsets 5 and 8 (c.f. table 4.5, page 73) should not occur too frequently.

User-defined filter

This filter takes as input a user-defined term or expression and spots sentences containing it. It provides a possibility to personalise the view of the document, enabling readers to highlight the sentences containing an expression they are interested in. This filter can be applied successively: each iteration highlights the additional sentences containing the newly queried expression.

Category	Relations and **selected synonyms from WordNet**
general	is about, uses, applies, is enabled by, improves on, impairs; **utilise, employs, implements, betters, amends, ameliorates**
problem	addresses, solves; **covers, deals**
supports/-challenges	proves, refutes, is evidence for, is evidence against, agrees with, disagrees with, is consistent with, is inconsistent with; **demonstrates, establishes, shows, rebuts, controverts, holds, concords, dissents, disagrees**
taxonomic	is part of, is an example of, is a subclass of
similarity	is identical to, is similar to, is different to, is the opposite of, shares issue with, has nothing to do with, is analogous to
causality	predicts, envisages, causes, is capable of causing, is prerequisite of, is unlikely to affect, prevents

Table 5.1: List of relations matched by our 'matched relations' spotting filter. It contains relations defined in the ScholOnto ontology (c.f. figure 2.2, page 16) and the WordNet synonyms we have retained (in **bold**.)

'Defined concepts and claims' filter

This filter retrieves (from the ScholOnto repository) existing (i.e. previously defined, by the current annotator or by one of her peers) concepts and claims associated to the current document. The set of defined concepts is different from the aforementioned set of 'matched concepts,' which is composed of the concepts defined in *any* document, but *matched* in the content of the current one. This set is instead composed of all the concepts associated to this document, whether they are *matched* in the current document or not. Defined concepts can be reused in a new claim or integrated in an existing claim. Defined claims can be duplicated and/or challenged.

'Cited documents' filter

Cited documents that are mentioned (manually) in the XML representation of the document (c.f. listing B.1, page 248) can be accessed.

'Related documents' filter

Two documents become *related* if there is a concept reused in both, or if a claim connects a concept defined in each of them. This filter provides links to related documents automatically.

Id	Sentence	Score
S-8	Our work addresses a different issue on the Web of Trust regarding whether to trust the content of a Web resource depending on its *source*.	7
S-20	This paper describes our initial work on *TRELLIS* to enable *users* to express their trust on a *source* and the statements made by it, and to combine individual views into an overall *assessment* of each source of **information**.	10
S-37	Our goal is to enable *users* to create annotations of their *analysis* of alternative **sources** of **information** as they make a decision or reach a conclusion based on their *analysis*.	10
S-202	We have shown an approach to capturing *assessments* of *users* about their trust on individual **information sources** as they are deciding whether and how to use **information** from each *source* in a specific *analysis* or decision making process.	12

Table 5.2: 'Important sentences' of the article '*Trusting Information Sources One Citizen at a Time*.' **Bold** words appear in the document title and words in *italics* appear in the headers.

'Important sentences' filter

This filter spots the most important sentences in the document [5]. We have seen in the previous chapter different surface-based approaches to assess the relevance of a sentence, by considering for instance its words or its location in the document. As an initial step, we assess the importance of a sentence by looking at its words and checking, for each of them (common stop words being ignored), if they can be found in the title, in the section headers, or in the abstract. A sentence gets three points for each word present in the title, two for each word in the section headers and one for each word found in the abstract. Table 5.2 gives an overview of the score associated to four sentences of the article '*Trusting Information Sources One Citizen at a Time*.'

'Candidate concepts' filter

This component identifies candidate ScholOnto concepts:

Keywords The keywords of the document are proposed as candidate concepts. They are extracted manually and stored in an XML file.

[5]The following three filters require extra-processing and therefore cannot be run within the environment. This should not have a major impact, since the results they generate can be considered immutable, a property that is not true for the components we presented earlier: for instance the 'matched concepts' filter should be run 'live' to match additional concepts as soon as annotators add them to a document. These filters use a standardised output format to enable their integration in the architecture (c.f. appendix B for examples of these files.

Sequences of determiners, adjectives and nouns We also retain noun groups as candidate concepts [6]. A part-of-speech (POS) tagger [7] assigns to each word of the text its most likely POS tag (*noun*, *adjective*, *determiner*, *verb*, *auxiliary*...)

We have followed the strategy defined in the NLTK tutorial [8] and created a chain of taggers (a backoff tagger), composed of a bigram tagger, a unigram tagger and finally a regular expression tagger. The interest of this approach is that every word is (1) tagged first by the bigram tagger, then (2) by the unigram tagger if the bigram tagger does not find a tag for that word, and finally (3) by the regular expression tagger if the two previous ones fail.

Regular expression tagger The regular expression tagger is the simplest of these taggers. It uses a regular expression to decide which tag to assign to a word token. If the expression is matched, it assigns one tag; if it is not matched, it assigns another tag. For our purpose, the regular expression tagger tries to discover if a word token is a number. If it is the case, the token is assigned the tag `CD` (for cardinal); if not, it is assigned the *noun* tag (`NN`.)

Unigram tagger A unigram tagger provides more accurate tagging, using the annotated corpus to assign to each word the tag that has the most frequently been associated to it in this corpus. For instance, if the word *work* has been found 10 times as a noun (as in *"My work relates to..."*) and 5 times as a verb (as in *"They work very hard..."*), each instance of *work* in the current document is assigned the `NN` tag.

Bigram tagger Although an improvement, the unigram tagger still assigns an incorrect tag in several cases. With the previous example, all the occurrences of *work* would be tagged as nouns, whether they actually are nouns or not. A bigram tagger refines the tagging by looking at both the word to tag and the tag of the word preceding it to make its decision. The assumption is that considering the preceding tag gives more context to make a sound decision. In our example, training the tagger yields several contexts for the word '*work*' [9]. When the tagger meets *work* in the sentence *"They work very hard..."*, it looks at the tag of the preceding word (a subject pronoun) to decide which tag to assign to '*work*'. If, in the training corpus, most occurrences of *'work', preceded by a subject pronoun* are tagged as a *verb* (`VB`), the classifier can correctly choose `VB` for this instance.

[6] We define a noun group as a sequence of determiners, adjectives and nouns.
[7] instantiated and trained with NLTK.
[8] `http://nltk.sourceforge.net/tutorial/tagging/index.html`
[9] A context is composed of both the word to tag and the tag of its preceding word

Combining taggers Combining these three taggers gives a more accurate coverage. Appendix B contains the output of this filter on the abstract of the document '*Trusting Information Sources One Citizen at a Time*.' Results are organised by frequency, with the most frequent noun groups presented first. This is a preliminary version, and further processing can be applied, for instance to remove erroneous propositions and to merge the singular and plural forms of a noun group.

'Rhetorical zones' filter

This filter can be considered an extension of the 'matched relations' filter. While the latter relies on specific keywords (ScholOnto discourse relations) to identify areas of a document in which an author defends her argument, this filter relies on a greater number of features to spot relevant sentences.

We have implemented a simpler version of the rhetorical parsing approach described in [Teufel and Moens, 2002], using a three-category scheme. This scheme focuses on the notion of scientific attribution: we are interested in sentences describing research work carried out by the author (OWN), work attributed to external (to the document) authors (OTHER) and work (or ideas, assumptions, hypotheses...) attributed to a research community in general (i.e. in which no explicit mention of a person's name is given) (BACKGROUND.) The goal is to find, for each sentence of the document, its most likely rhetorical category [10].

Description of the sentences of the training corpus A training corpus is built to learn correlations between the values of a set of features and the categories they are more likely to belong to. 10 research articles' introductions taken from the KMi technical reports collection are used to create this mini-corpus (their annotation consisted in determining, for each sentence of each paper, its category.) The final corpus contains 230 sentences for the OWN category, 135 sentences for the OTHER category, and 244 sentences for the BACKGROUND category. Our naive Bayes classifier uses a 'bag of words' approach in which every word appearing in any sentence of any training document is considered a feature.

Annotation at this stage is made by the author of this dissertation only. Phenomena such as the stability and reproducibility of the annotation are not considered.

Learning A naive Bayes category is trained on these annotated sentences to learn, for each word of each sentence, its probability to belong to each category. The formulas used to

[10] In a recent private communication, Simone Teufel has confirmed that these three roles are the key ones in her classification set.

Id	Sentence	Category	Confidence
...			
S-25	The paper begins with an overview of TRELLIS as an information analysis tool.	OWN	0.799
S-26	Then we describe how users can specify source descriptions and qualifications in TRELLIS.	OWN	0.990
S-27	We show how TRELLIS derives ratings for each source, averaged over many users and many analyses.	OWN	0.572
S-28	We discuss how these ratings can be presented in useful ways to users to help them assess sources in subsequent analysis with TRELLIS.	OWN	0.442
S-29	We conclude with related work and a discussion of future directions.	OWN	0.580
S-30	TRELLIS is an interactive tool that helps users annotate the rationale for their decisions, hypotheses, and opinions as they analyze information from various sources.	BACKGROUND	0.462
...			

Table 5.3: Output from our re-implementation of Teufel and Moens's rhetorical classifier on the article '*Trusting Information Sources One Citizen at a Time*.' As shown in forthcoming table 5.4, the presence of *We* and *paper* in a sentence strongly influences the decision of the classifier to put a sentence in the OWN category.

compute these estimates are presented in the literature review, c.f. page 78.

Classification Table 5.3 gives an extract of the classification performed by this component. Sentences are assigned a category and a degree of confidence in the assignment. The closer to 1, the higher the confidence of the classifier is.

Features impact Our 'bag of words' assumption potentially results in an extremely high number of (term, document) probability estimates to compute (this may not hold in our example using a limited training corpus, but the problem exists.) To reduce this cost, a feature space reduction process can be applied (the Bayes independence assumption already eliminates a part of the computations by assuming that every term appears independently of the others.) This also gives us a chance to identify the features (in our case, words) which have the highest impact on the classification process, i.e. the terms that contribute the most to the decision of putting a sentence in a category rather than in another.

Feature set reduction To reduce the feature space, we apply a χ^2-based filtering operation [Yang and Pedersen, 1997]. Yang and Pedersen define the χ^2 metric as '*a measure of the lack of independence between a term t and a category c*.' The metric is computed with

Rank	Own	Other	Background
1	[citation]	[citation]	we
2	we	specification	university
3	section	we	visualisation
4	paper	eg	students
5	our	tools	useful
6	or	architectural	paper
7	how	protégé	not
8	two	his	remain
9	this	words	interests
10	followed	spark	amount
...	introducing, describe, present, first, study, ...	includes...	...

Table 5.4: The 10 most contributing terms for each category of our classification.

the following formula:

$$\chi^2 = \frac{N \times (AD - CB)^2}{(A + C) \times (B + D) \times (A + B) \times (C + D)} \quad (5.1)$$

where N is the number of training documents, A is the number of times a term t is found in a category c, B the number of times a term t is found without c, C the number of times a category c does not contain the term t, and D the number of times t nor c appear.

Results The results, presented in table 5.4, appear to be very mixed, which is not surprising given the small size of our corpus. Several terms considered to be highly correlated with their category are very good nevertheless, especially in the OWN category. In particular, *section*, *paper*, *our*, *describe*, *present* and *study* are good indicators of a sentence referring to the work being presented by the author. Similarly interesting candidates from the OTHER list are *his* and `[citation]` (a meta-expression which we have used in our preparation phase to replace every instance of a citation, such as '[4]', '(Smith 2005)' or '(Smith *et al* 05).')

On a less positive note, a few domain-related terms such as *protégé* appear, probably because of the limited size of the corpus. As there are not enough instances of 'good terms' (compared to domain terms), these unwanted domain terms are not considered 'unimportant' enough. More surprising, the presence of *we* in different categories, at a high rank, is interesting. This can be explained by the fact that the expert preparing the corpus has found some instances of *we* used to designate a community as a whole, instead of referring to the authors of a paper. Consider the following instances:

- OTHER: '*We say different things to different people varying the level of detail emphasis perspective and so forth.*'

- OWN: '*We are currently conducting this research in the context of a new first-level course on object-oriented computing.*'

The former use of *We* refers to a community as a whole, while the latter is referring explicitly to the authors of the paper. Discovering which words contribute the most to the decision of putting a word in a category instead of in an other one is very interesting, as it can at times challenge our own expectations. Additionally, it gives results that can be tailored to match the jargon of a given community.

5.3 User interface

The second major part of the ClaimSpotter environment is its user interface. It is composed of the following components:

- a *document viewer*, in which where the working document can be browsed and its presentation modified by the application of the aforementioned spotting filters;
- an *input form* to record concepts and claims;
- additional windows to display further information.

5.3.1 Navigation design

Figure 5.2 presents the revised version of the navigation design diagram. This extended version of the schemas we have used in our design phase (c.f. figure 3.2, page 36, and 4.25, page 100.) includes the main screens of our active semantic annotation environment and their interconnection [Dix et al., 2004, page 204].

We have incorporated the various tasks that can be performed within ClaimSpotter. Tasks, identified from our paper-based study and our literature review, are organised according to the window they 'belong' to. The five windows available in ClaimSpotter are: (i) a main *annotation* window, in which annotators can read the document, manipulate its presentation (via the activation of spotting filters that act as lenses over the content of the document) and record their concepts and claims (our 'annotation' activity in earlier versions of this diagram); (ii) a *history* window that regroups all the operations related to the ScholOnto repository, including searches (our 'consultation' activity); (iii) a *more ideas* window that displays the candidate concepts identified from the document; (iv) an *add a document* window in which annotators can add a document to the repository; and (v) a *help* window displaying the help screen.

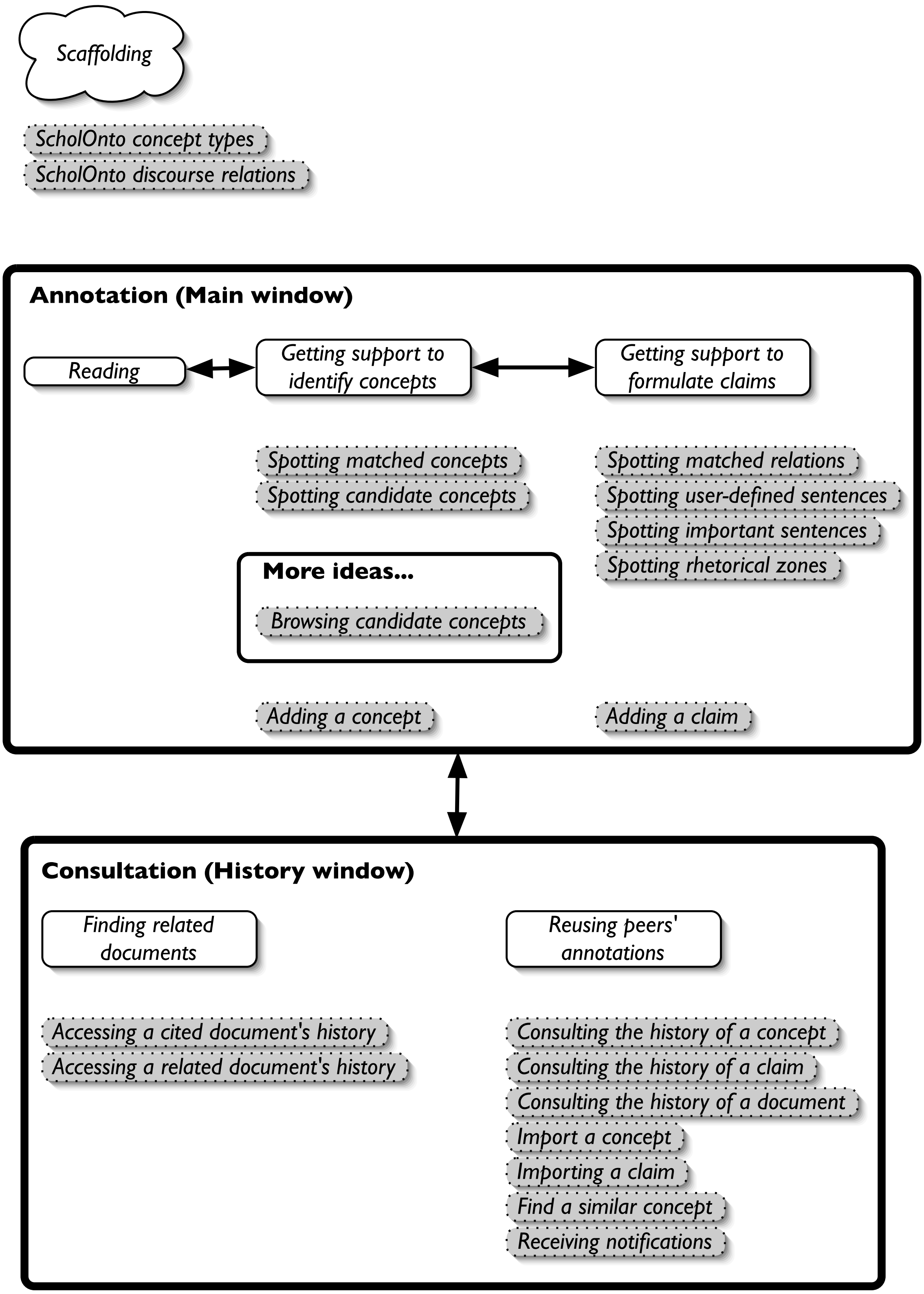


Figure 5.2: Navigation design diagram for the current version of ClaimSpotter. It lists the different operations an annotator can perform, organised by the window they belong to, and their interrelations. The annotation environment is composed of two main windows (*annotation* and *history*) and three secondary windows (*more ideas*, *add a document* (not shown) and *help* (not shown.))

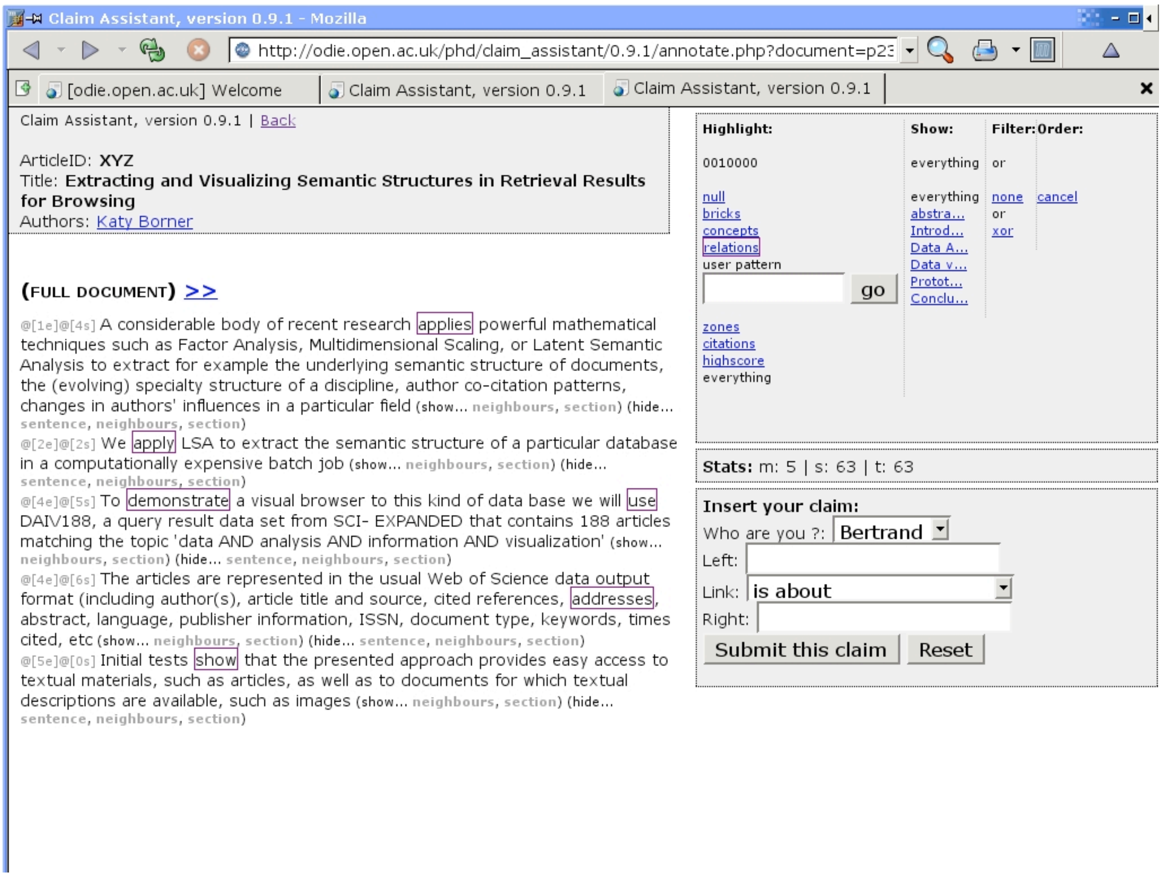


Figure 5.3: A first prototype of ClaimSpotter, ClaimAssistant (2002 version.)

5.3.2 Design evolution

The environment has evolved over the last 3 years, resulting in several versions fixing bugs and adding new features. These improvements have resulted from internal testing and discussions with the project supervision team. Figure 5.3 is a screenshot of the early days of ClaimSpotter (named ClaimAssistant at the time), in 2002, while figure 5.4 is a screenshot of ClaimSpotter in 2003. We present below a few salient aspects which have evolved in these successive versions.

Consistency in terminology

Consistency in the interface enables users to transfer their knowledge and skills from one application to another [Apple, 2005]. We can extend this definition: it also enables users to transfer their knowledge from one part of the application to another part of the same application. To improve consistency within the interface, we have corrected instances of vocabulary mismatches in distinct parts of the interface (for instance, 'relations' and 'claims' used to designate a similar object.)

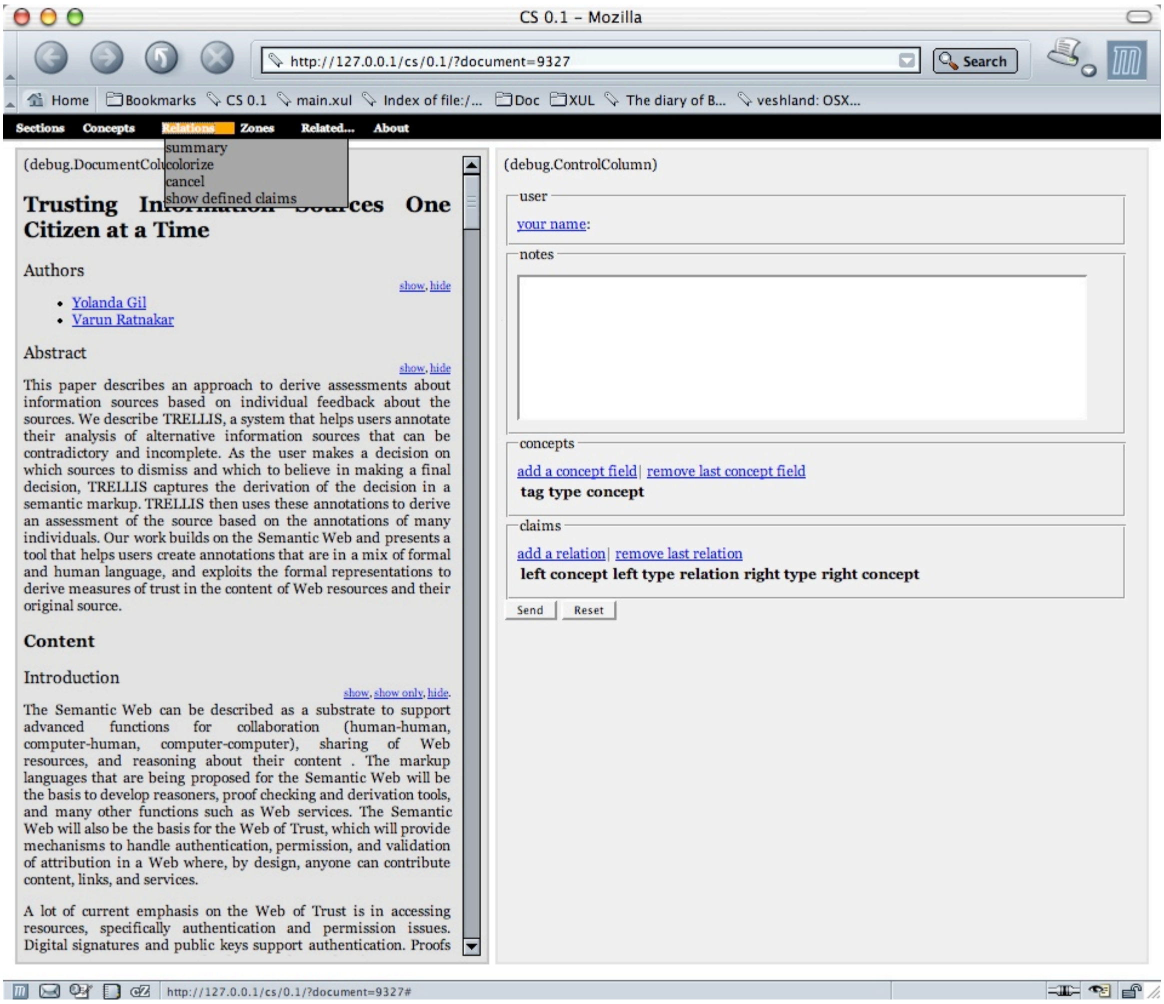


Figure 5.4: ClaimSpotter (2003 version.)

Navigation and access to information

Navigation within a document and access to additional resources is one of the aspects which has the most evolved.

Space fragmentation With multiple sources of information to display on the screen, we have tried to find the best way to fragment the available space into coherent zones, shown in the right part of the architecture diagram (c.f. figure 5.1.) We have opted for a light colour scheme that separates the main window into three main components: a document viewer, an input form and several command toolbars.

Table of contents The table of contents on the left side of the interface has also been added. We have begun the development of this environment with short papers (2 pages only): the need to access a particular area directly was not crucial. However, as soon as we have started to include longer papers, the need to jump from, for instance, the results section back to the introduction has become more acute. We have added this feature to enable annotators to access any section or subsection directly. This is a fixed component: it is always displayed on the screen.

Toolbar The toolbar has also been improved. Options were originally gathered on the right side of the interface, in a list-like presentation (c.f. figure 5.3.) One major inconvenience was that these options were not 'fixed' in the interface, and scrolling down a document was hiding them. We have replaced this list with a toolbar sitting at the top of the annotation window (c.f. figure 5.4.)

The toolbar also featured several operations that were conceptually unrelated (such as going back to the login screen, highlighting a particular class of information in the text and accessing the history window.) It was also getting longer and longer as we were adding options to it. We have first adopted several background colours to regroup conceptually-related operations (for instance, the family of spotting filters.) This has been successful up to a point [11].

We have then reorganised these operations into two toolbars: the upper one gathers the 'main' operations, including 'going back to the login screen' or 'activating the history window.' The second toolbar, located below the first one, gathers all the 'spotting' operations associated to the personalisation of the document. Both toolbars are 'fixed' in the interface: they remain visible at any time.

A history window This additional window has been created to display information queried from the repository of annotations (concepts available, claims, documents…) We used to have multiple pop-up windows, such as one for existing concepts, another one for existing relations and yet another one for cited documents. Using a single window has improved the interface, by providing a single area to look for for these conceptually-related sources of information.

Visual feedback

Visual feedback has also been improved over time. We have already mentioned the adoption of a colour scheme to separate the different areas of the interface. This scheme breaks the available space into:

- a *document viewer* in the middle containing the document, in which annotators can read and interact with the suggestions proposed.

[11] We present screenshots of the various elements of the interface in the next section.

- a *table of contents* on the left side
- an *input form* on the right side in which annotators can *input* their concepts and claims.
- two *toolbars* at the top of the screen, which list the different operations which can be applied.
- a *footer*, which mentions the name of the current annotator and the document she is working on.

Colour scheme A range of signals is used to emphasise the spotted suggestions, making use of variations on colour, font weight and font size [12]. We have ensured that any combination of highlighting options remains legible:

- Existing concepts matched in the content of the document are displayed with a yellow background. An additional border indicates the presence of a coloured area.
- Existing instances of the relations defined in the ScholOnto ontology are displayed with a blue background. An additional border indicates the presence of a coloured area.
- Important sentences are displayed with a larger font, emphasising their relative 'importance.'
- Rhetorical relations are highlighted using a colour scheme, with a colour for each rhetorical category.
- Finally, sentences matched against a user-defined query are emphasised with a specific colour.

The 'Help' screen provides a key for the variations used. It is accessible from the top toolbar. Although this is not currently possible, an interesting option would be to let users define their own schemes to highlight the different categories of information.

Alerts Alerts (in the form of pop-up windows) have also been added. Initially, some operations were performed by the environment (typically, highlighting one class of information) without letting the user know whether the operation was completed or not. In most cases, this was not causing too much of a problem as at least one instance of the desired class of information could be found in the first 'screen-height' of the document (a matched existing

[12] Rendering is implemented as CSS stylesheets.

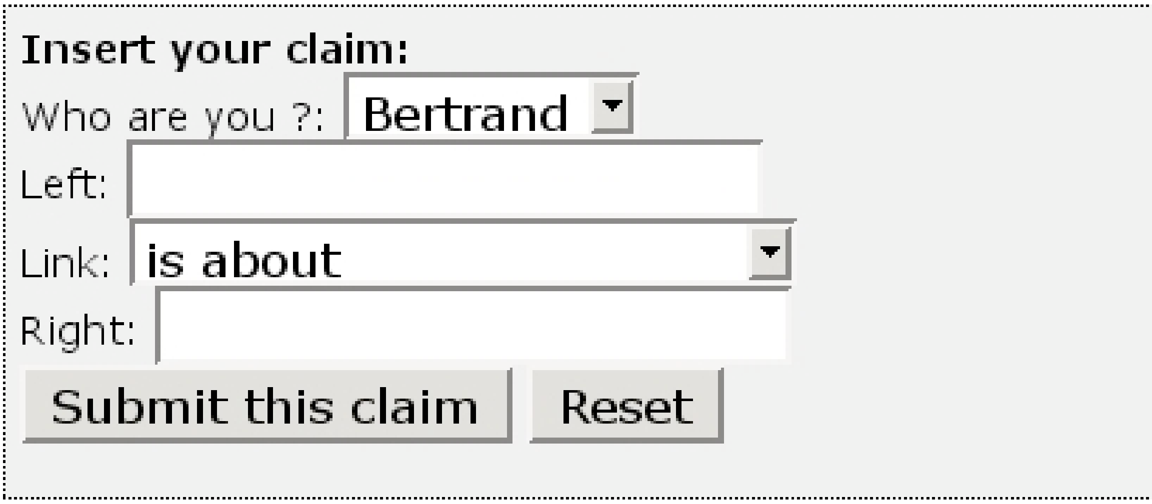


Figure 5.5: Input form (2002 version.) It displays components of a claim using a vertical layout, creating additional confusion.

concept, for instance.) However, we have realised over time how important a visual alert was, in order to let the user know that her choice had been taken into account. We have added pop-up windows for these operations for which feedback is not immediately given on the screen.

Tooltips Tooltips have also been added to most operations. Hovering on a button gives an indication of the operation associated to it.

Input form

The input form has also gone through several revisions.

From a vertical to an horizontal layout For space-saving reasons, each claim was initially rendered as a three-row area, with the top row containing the source concept (the left part of the claim), the middle row containing a drop-down menu listing the different relations available, organised by their categories, and the bottom row containing the destination concept (the right part of the claim) (c.f. figure 5.5.) While this layout was efficient in the sense that it was giving large input (typing) areas for concepts, it was at odds with what annotators would expect. The source and destination parts of a claim are indeed implying a notion of flow, a flow that is best approached if the claim is arranged with the *source concept* on the left, the ***relation type*** in the middle, and the *destination concept* on the right (as in the notation we have used so far to represent claims on this document, {*source concept*, ***relation type***, *destination concept*}.)

Relation types A pull-down menu organises the different relations available by their category (c.f. figure 5.6.)

Shortcuts Several shortcuts to facilitate the manipulation of the components of a claim have been added.

Personalisation

Possibilities to combine spotting filters as lenses over the document can result in highly-personalised views, tailoring the document to let it show as much or as little as desired. We expect this feature to be essential. Figures 5.7 onward show several examples of personalised views.

5.3.3 Current version

The interface of the current ClaimSpotter prototype is Web-based. It acts as both a document reader [Graham, 1999] - the appearance of this document can be modified by the activation of different spotting plug-ins - and a tool to record concepts and claims. Figure 5.7 gives an overview of the current, 2005, version of this environment.

In this section, we give a short account of the features of ClaimSpotter, starting with this screenshot taken in the middle of an annotation process (a more detailed presentation is given later.) An annotator has combined several filters from the 'spotting' toolbar ④ to get suggestions to consider and/or react against ①. The structure of the document is presented in the table of contents panel ② to facilitate navigation. Elements found in the text include existing concepts, matched relations and sentences matching a user-defined query. The input panel ③ enables the insertion of notes ⑤, concepts ⑥ and claims ⑦. Concepts can be combined in triples and connected with a relation ⑧. A footer mentions the name of the current annotator and the document she is annotating ⑨.

5.3.4 Controls

The interface of the current version of ClaimSpotter is composed of two toolbars, a main area split in three panels and a footer.

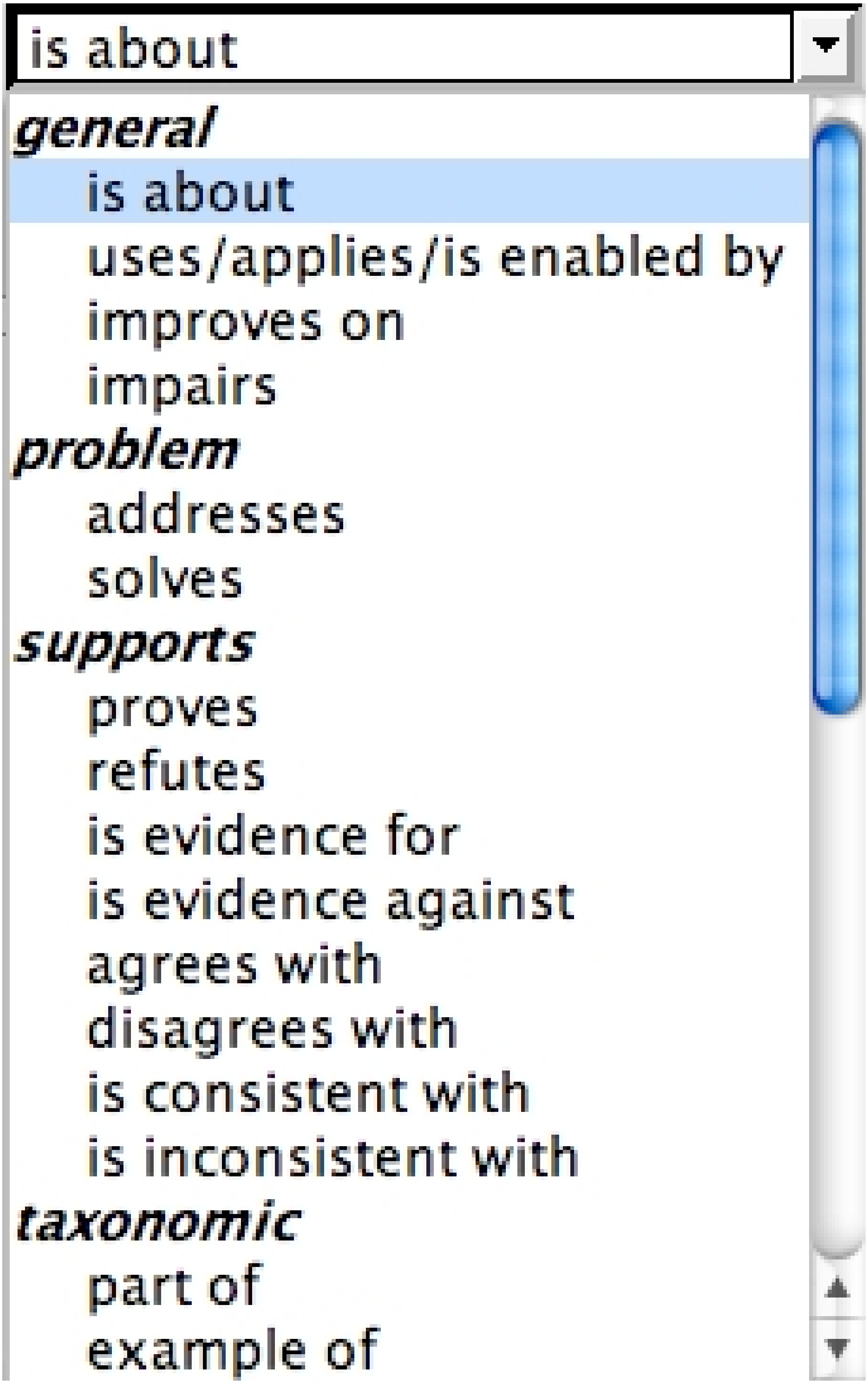


Figure 5.6: Picking a discourse relation for a claim in ClaimSpotter.

'Main' toolbar

On the upper side, a first toolbar regroups the main operations that can be performed in the system. It offers possibilities to go back to the login screen, to access the history window, to add a document to the repository, to switch between the two views associated to the interface and to export the current literature model towards a graph file. An 'about' button gives basic information about the environment, and a 'help' button activates a help screen summarising the basic functions of the environment and the principles of the ScholOnto model. Figure 5.8 provides a zoomed view of this toolbar.

'Spotting' toolbar

The second toolbar (the element marked ④ in the previous screenshot) regroups the different spotting filters that can be activated. By using and combining these suggestions, an annotator can build her own representation of a document, by highlighting for instance the concepts she has created earlier (in another document) and the sentences matching a particular expression. Figure 5.9 provides a zoomed view of this toolbar.

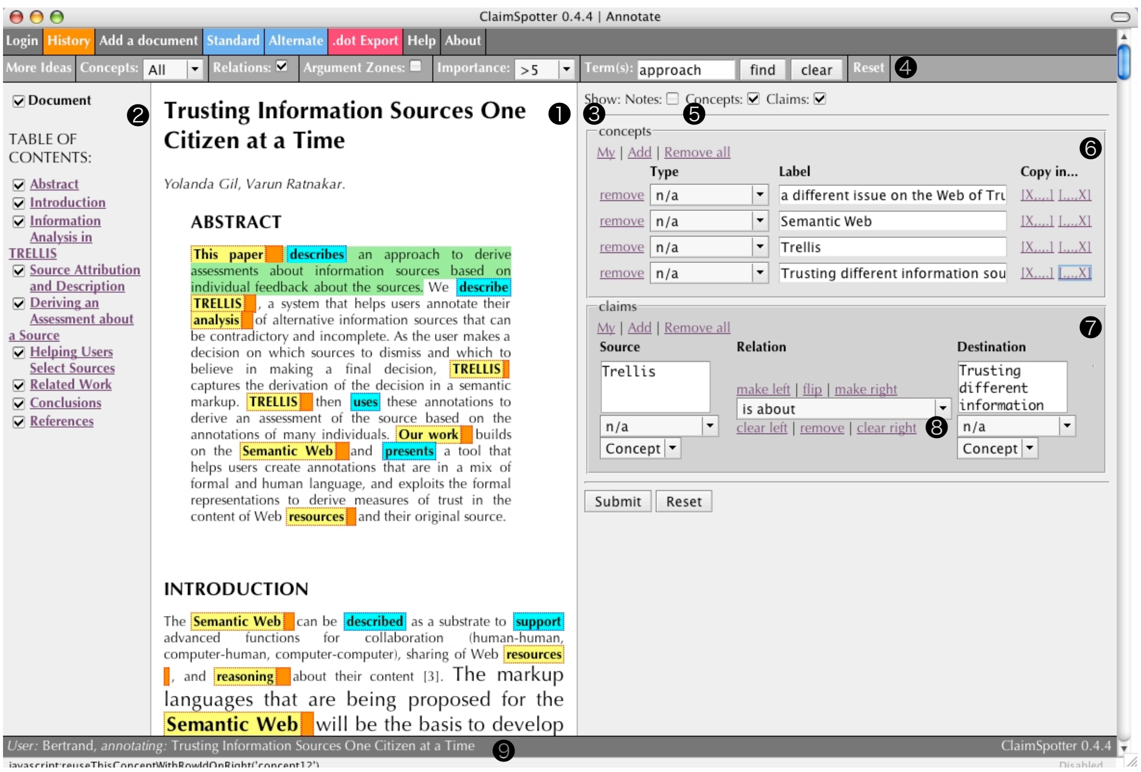


Figure 5.7: ClaimSpotter (2005 version.)

Figure 5.8: 'Main' toolbar. Colours are used to group related options.

Figure 5.9: 'Spotting' toolbar.

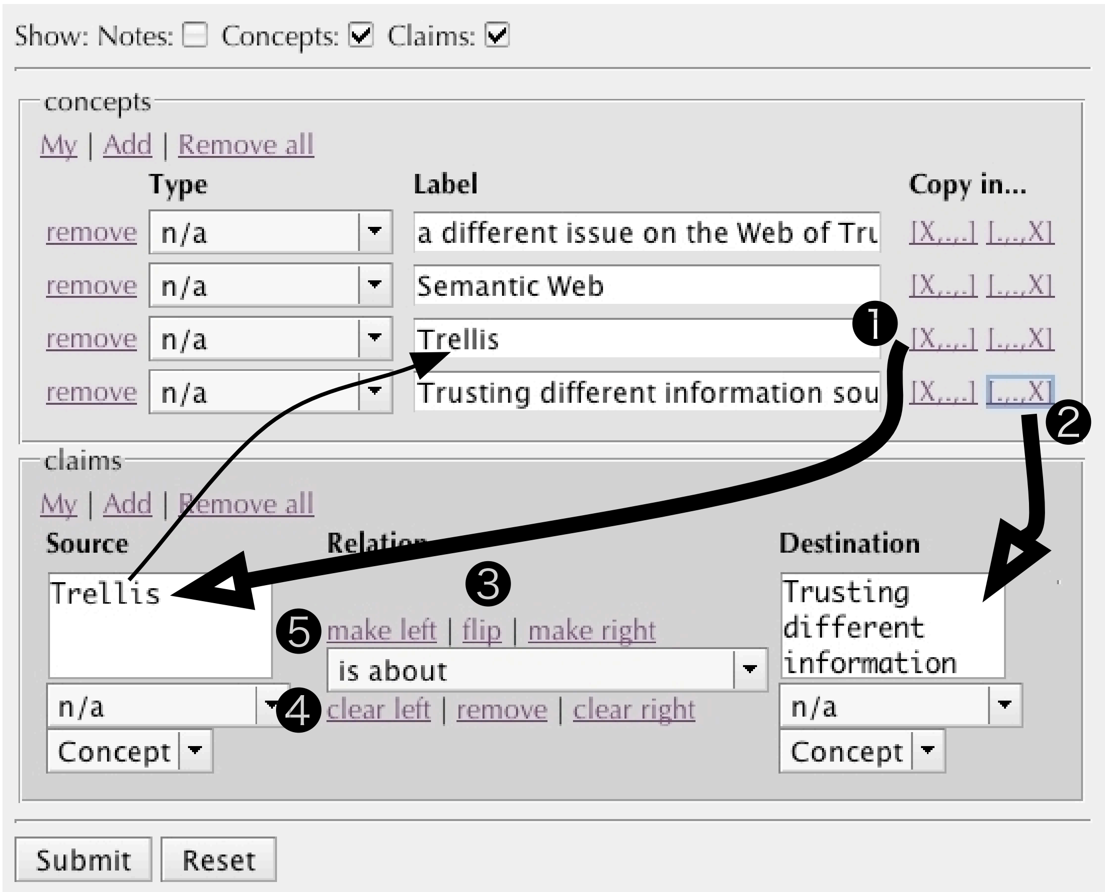


Figure 5.10: 'Input form' panel.

Main area

Table of contents On the left side of the main area, a 'table of contents' panel offers possibilities to quickly navigate within the document. Options to show and hide any component are provided (c.f. figure 5.11.)

Document viewer This panel features the document currently annotated.

Input form This panel is used to record concepts and claims (c.f. figure 5.10.) Concepts can be typed in directly, dragged and dropped from the content of the document, or imported, by clicking on their occurrences in the document (examples are given later.)

A notepad is also incorporated: annotators can type or drag and drop portions of the text

Figure 5.11: ‘Table of contents’ panel. Showing/hiding a section shows/hides its subsections. A shortcut to show/hide the whole document is provided.

to refer to at a later stage. It can be used to copy a paragraph one wishes to model with claim triples, for instance. These notes can be parsed to look for instances of ScholOnto relations and selected WordNet synonyms. If a relation is found, the content of the note is split around it and a claim is created.

Shortcut links The input panel also provides several shortcuts to facilitate the formulation of claims. Two buttons [X,.,.] and [.,.,X] ① ② (c.f. figure 5.10) copy the concept they are associated to to the left or right side of the last claim, if that space is empty and ready to receive a concept; or to a new, blank, claim triple if that space is already occupied by a concept. In the example, the user has clicked on the [X,.,.] button of the concept *Trellis* to reuse this concept in the left part of a new claim, then on the [.,.,X] button of the concept *Trusting different information sources* to reuse this concept on the right part of the same claim.

The left and right sides of a claim can also be flipped ③, transforming {*Trellis*, ***is about***, *Trusting different information sources*} into {*Trusting different information sources*, ***is about***, *Trellis*}. The user can clear either side of the claim to try a different concept ④. Finally, if she creates a claim by typing its constituents (concepts), she can copy these elements back to the concept list. This gives her the possibility to reuse them at a later stage (using the[X,.,.] and [.,.,X] buttons) ⑤.

These facilities have been added after personal extensive experience with ClaiMaker forms and our own experimentation.

Footer

It contains the name of the current annotator and the document she is annotating.

5.4 Virtual tour

Having presented the different parts of the ClaimSpotter environment, we can now illustrate its usage with a virtual tour, featuring an annotator making sense of a document.

5.4.1 Log in (1/20)

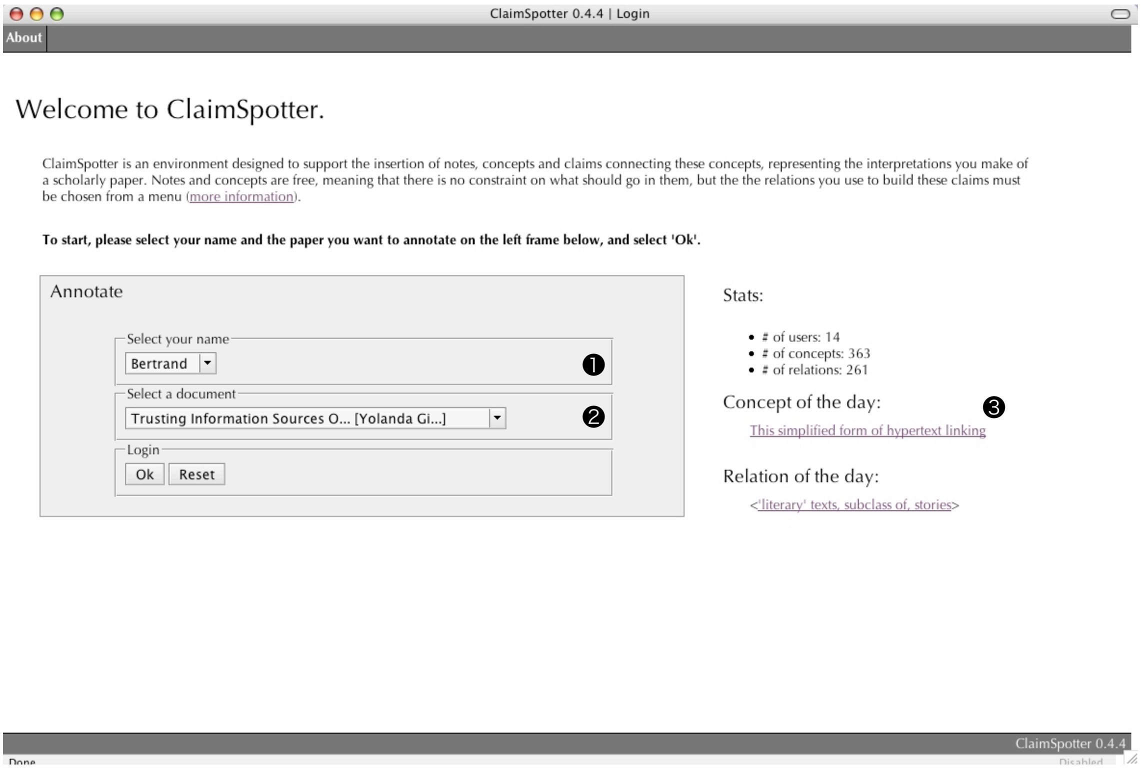


Figure 5.12: Virtual tour | Log in. (1/20)

To log in ClaimSpotter, our annotator selects his name ① and the document (available in XML format) he wishes to annotate ②. On the right side of this screen, a *gestalt* view gives him a few statistics and the concept and claim of the day ③, a humorous addition to bring life to the environment, ensuring that this screen is different from one day to another.

5.4.2 Annotation (2/20)

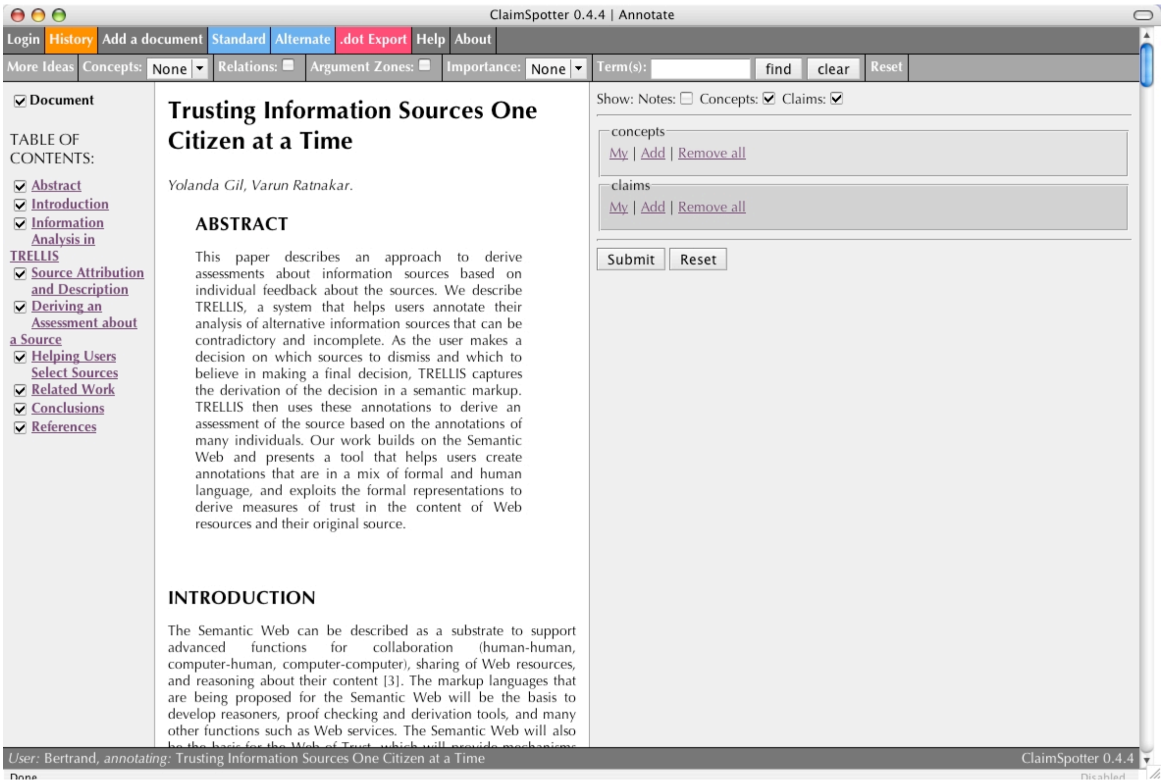


Figure 5.13: Virtual tour | Annotation. (2/20)

The initial view of the interface presents a pristine view of the document.

5.4.3 Annotation (3/20)

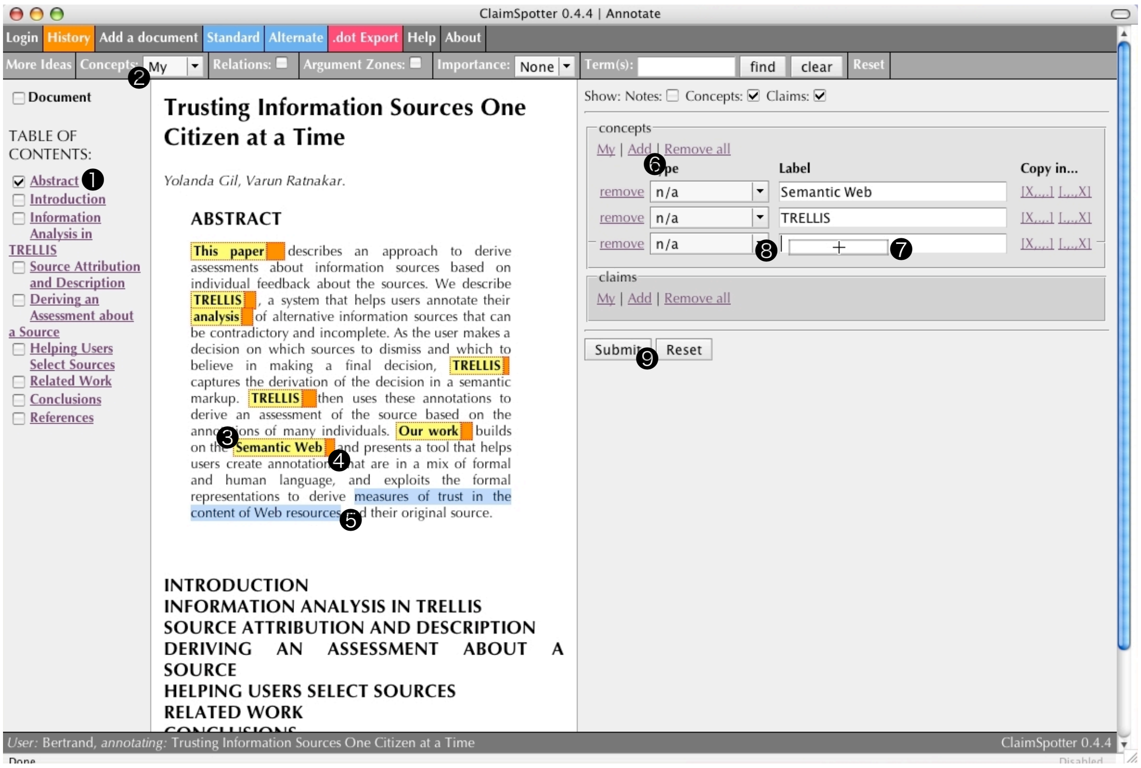


Figure 5.14: Virtual tour | Annotation. (3/20)

Our virtual annotator starts by focusing on the abstract. To this effect, he has hidden the entire document but this part, using the table of contents ①. He has also activated his previously added concepts (whether grounded in this document or not), matched in this document ②. Each matched concept can be clicked ③ and imported in the input form. On the right side of each matched concept, there is a small orange button: clicking it launches the 'History' window for that particular concept ④. As the annotator comes across the group of words '*measures of trust in the content of Web resources*' ⑤, he decides to record them as a concept. He can either click on the 'add' button of the concepts area of the form ⑥ and type it in; or select words and drag and drop them into an empty concept box ⑦. Once the concept is dragged and dropped, and classified with an (optional) type ⑧, he submits his annotation ⑨. Concepts already defined for this document and this annotator (e.g., *Semantic Web*, *TRELLIS*) are not duplicated. Only *measures of trust in the content of Web resources* is added to the repository.

5.4.4 Annotation (4/20)

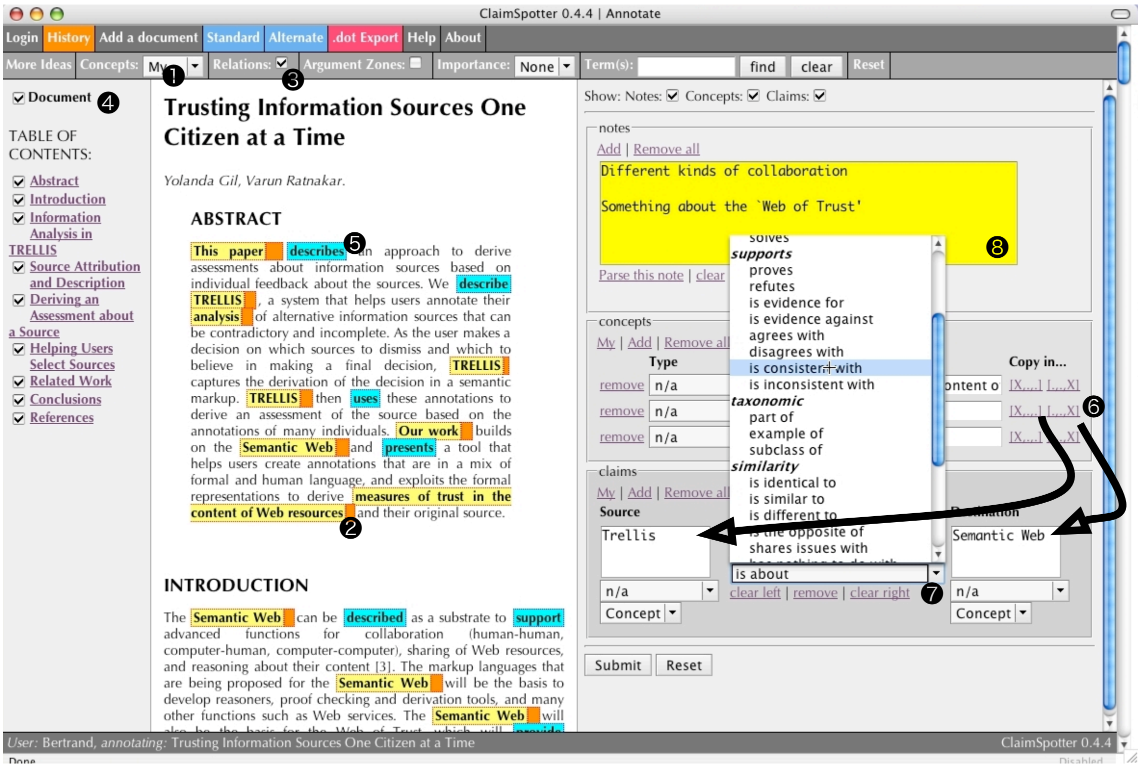


Figure 5.15: Virtual tour | Annotation. (4/20)

After the submission, the document is rendered in its original view, without any filters applied. Our annotator activates his concepts again, to see if there is any new matched concepts in the document ①: the concept *measures of trust in the content of Web resources* is now picked by the 'matched concepts' spotting filter ②. He decides to change the view of the document to look for different clues: he activates the 'matched relation' filter ③ in the entire document ④. Relations, as concepts, are clickable: we will see an example of this in the next step of this tour ⑤. For now, our annotator decides to create a claim. He can either type the claim triple from scratch in the form, or combine the concepts he wants to use with the [X,.,.] and [.,.,X] shortcut links ⑥ and select the relation to use ⑦. He finally decides to write a note ⑧ that is saved as a single concept *Different kinds of collaboration. . . Trust*, since it does not contain any ScholOnto relation.

5.4.5 Annotation (5/20)

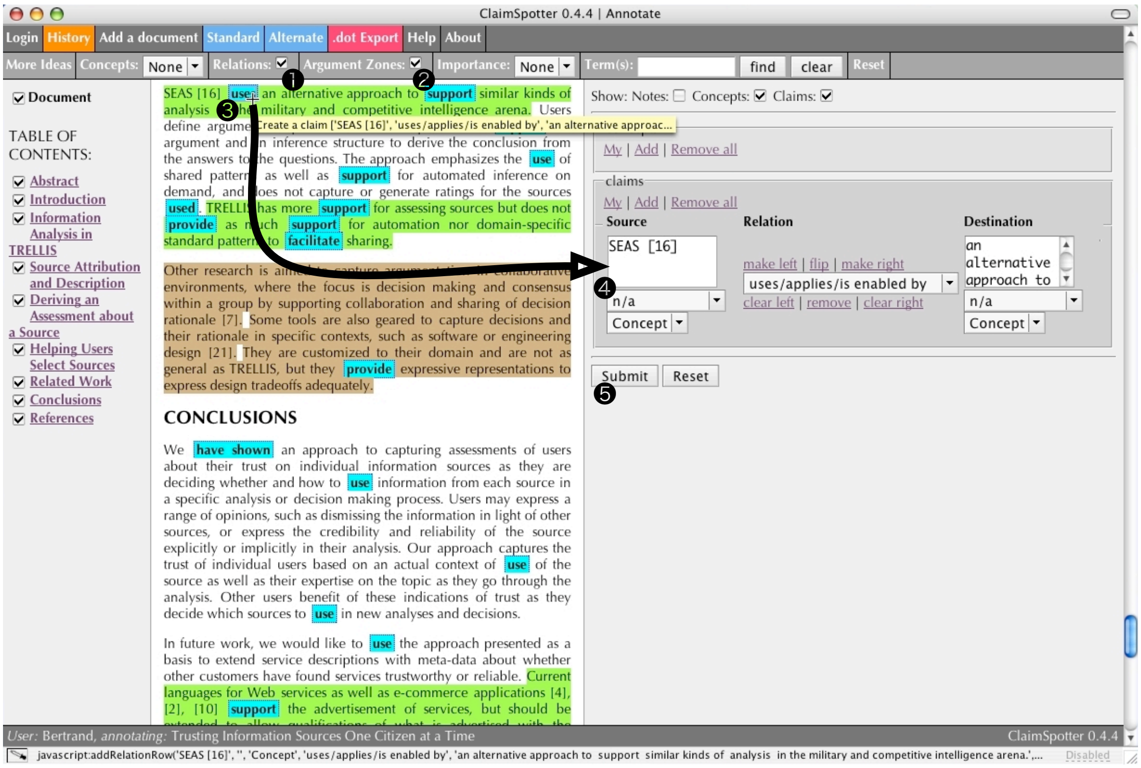


Figure 5.16: Virtual tour | Annotation. (5/20)

At this stage, our annotator creates another view of the document, activating again the 'matched relations' filter ① and adding the rhetorical zones filter (red and light green sentences) ②. Clicking on an instance of a ScholOnto relation in the document ③ creates a claim by splitting the sentence into a triple ④ centred on the detected relation: the sentence up to the relation goes on the left side, the sentence from the relation on the right side, and the relation itself, the verb (or a synonym if it is one of the WordNet elements considered (c.f. table 5.1, page 106)), goes in the middle part of the claim. This screenshot shows that not every instance of a ScholOnto relation matched in the document makes an interesting claim triple. In this example, our annotator finds the generated claim triple - {*SEAS*, ***uses/applies/is enabled by***, *an alternative approach to support similar kinds of analysis in the military and competitive intelligence arena*} - significant and decides to submit it ⑤.

5.4.6 Annotation (6/20)

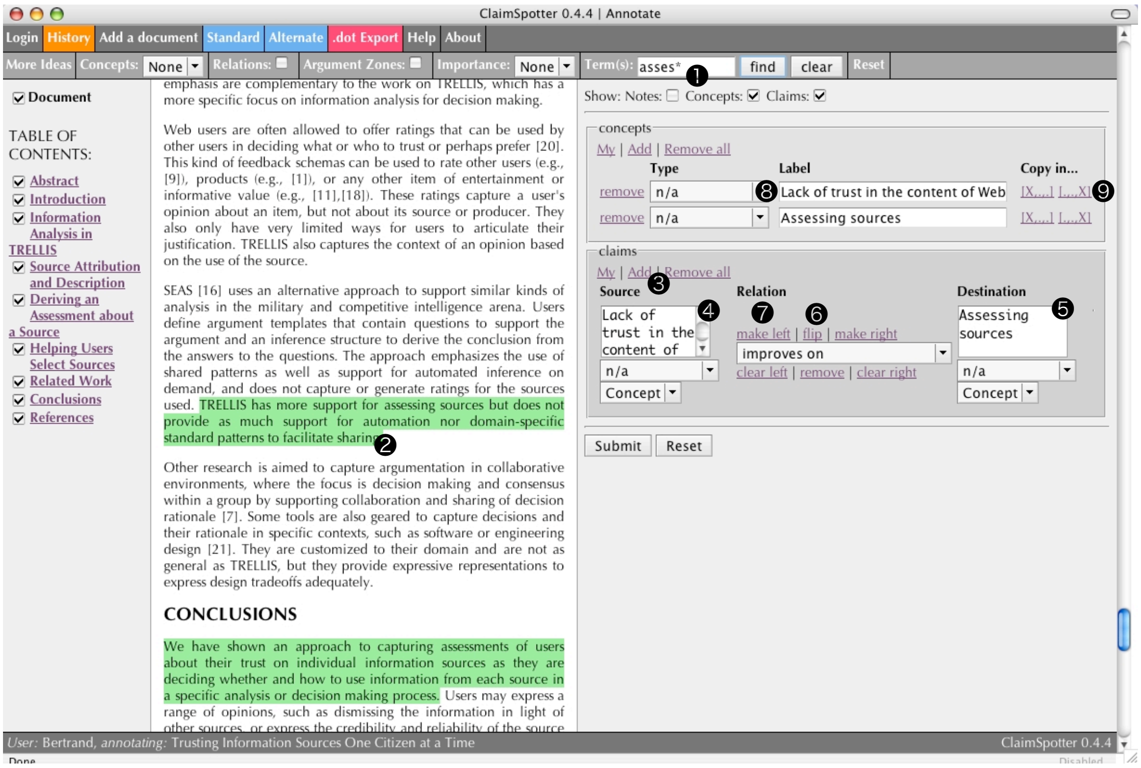


Figure 5.17: Virtual tour | Annotation. (6/20)

Our annotator wants to focus on the notion of 'assessment.' He activates the 'user-defined' filter ①. This filter highlights ② the sentences containing the terms matched by this expression (including 'assess', 'assessment' ...) He then models a claim by adding a blank claim to the form ③ and typing its concepts ④ ⑤. He realises that 'improves on' is the right relation to use but the order in which the concepts are arranged is not satisfying. The 'flip' button ⑥ enables him to switch the left and right concepts: {*Lack of trust*, ***improves on***, *Assessing sources*} is replaced with a more meaningful {*Assessing sources*, ***improves on***, *Lack of trust*}. The left and right objects of a claim can be duplicated in the concepts part of the form ⑧ with the 'make concept' shortcut links ⑦, enabling them to be combined in new claims without having to re-type them (using the [X,.,.] and [.,.,X] shortcut links ⑨.)

5.4.7 Annotation (7/20)

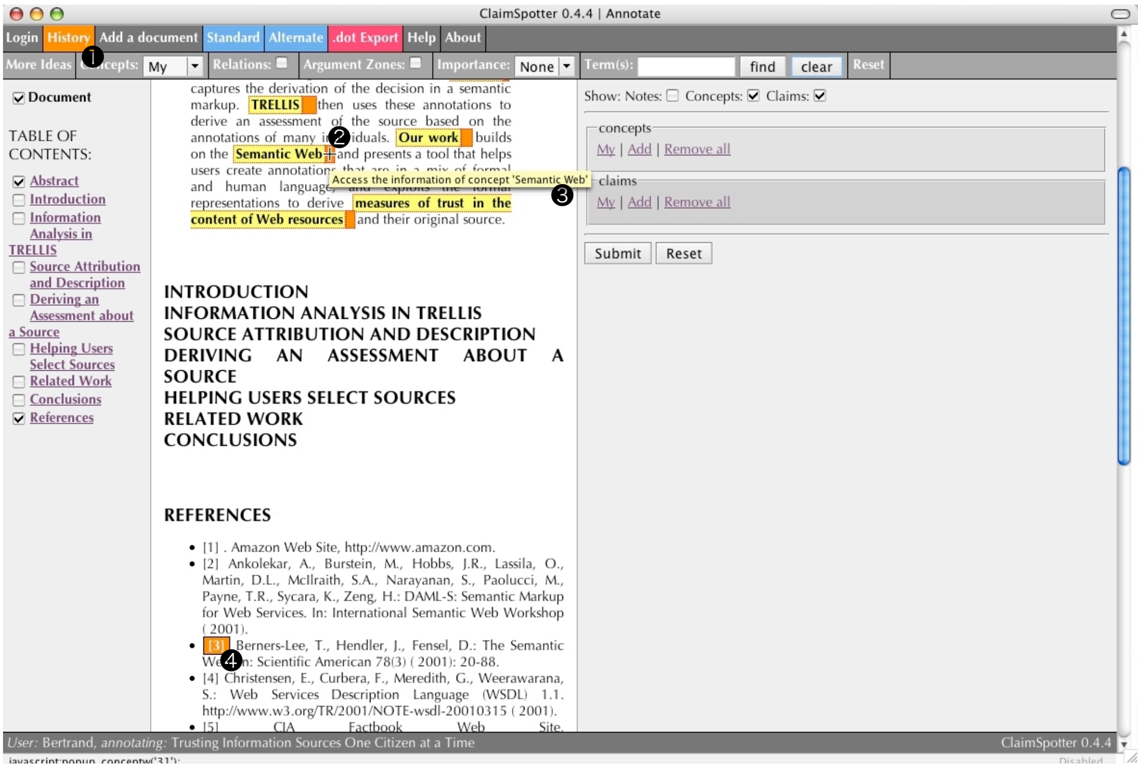


Figure 5.18: Virtual tour | Annotation. (7/20)

The history window is accessible from the main toolbar ①. It contains information associated to the current document: its concepts, its claims and the documents it cites (using the manually created citations XML file listing the cited documents existing in the ScholOnto repository.) It also shows the information associated to any concept, claim, or document cited in the current document. Shortcut links to the relevant history page for the cited documents are provided for each existing concept found in the document ②, with a tool-tip indicating the nature of the information available ③. Links are provided for cited documents ④, enabling our annotator to access their concepts and claims and to reuse them (for instance, by creating a claim between a concept in the current document and a concept in a cited document; more about this later.)

5.4.8 History (8/20)

We now turn to additional screens that are not part of the main window but that are launched via buttons in the general toolbar.

The History window displays a summary view of any concept, claim or document existing in the ScholOnto repository. Figure 5.19 lists the different options available in the toolbar of this window: our annotator can access his profile page, a summary view for the document he is currently annotating, a search page to look for any concept, claim or document title matching a query term, and a page to discover the most similar existing concepts to a query expression. These options are presented in the following pages.

Figure 5.19: History toolbar.

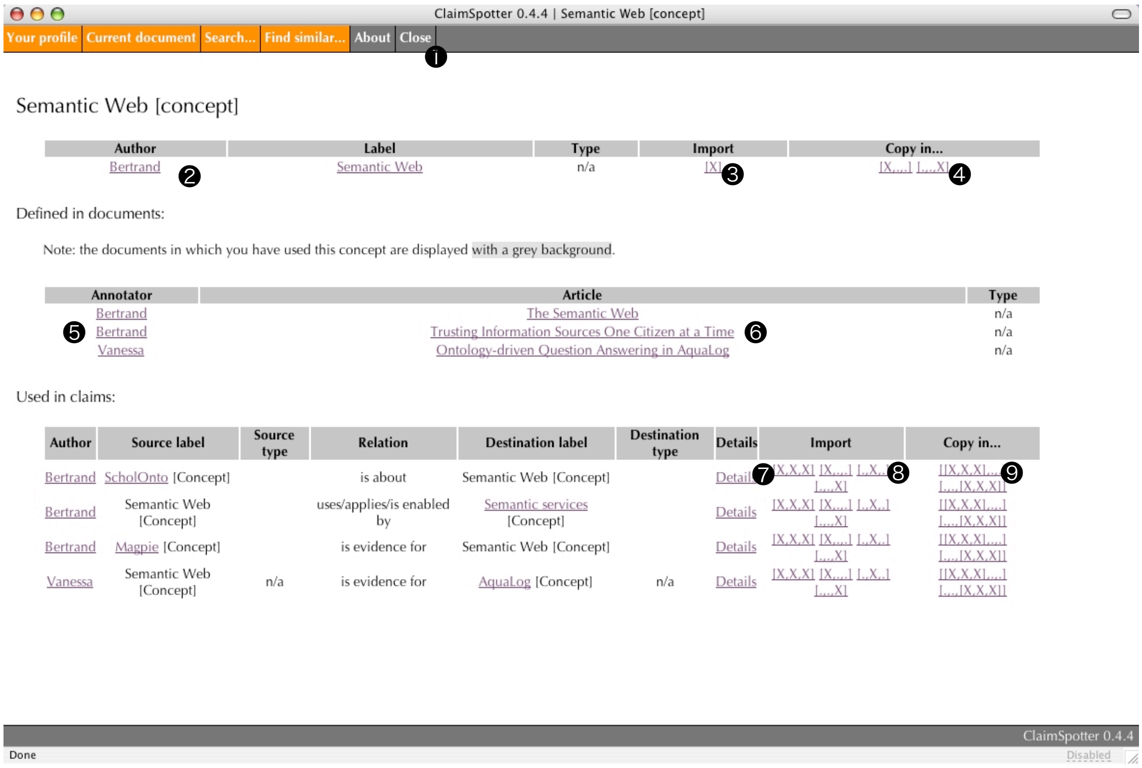


Figure 5.20: Virtual tour | History. (8/20)

Our annotator decides to learn more about the concept *Semantic Web* (this window is shown after a click on button ② in the previous screenshot: it aims at answering questions such as '*Who has created this concept?*', '*Which document has it been used in, and by who?*', and '*Which claims is it involved in?*') The window features the creator of this concept at the top ② and facilities to import it in the current document as a concept ③ or as a left or right part of the last claim (if empty) or of a new claim ④. Imported concepts are added to the input form of the main annotation window.

It also shows the different contexts this concept has been used in: it has been used already by our annotator in another document and by one of his peers in another document ⑤ ⑥, consciously or not: his peer may have checked if this concept was already existing and if so decided to reuse it, or he may have created his own version without checking beforehand. Since the system does not allow duplicates, both have in effect reused an existing concept. Clicking on the name of a document launches the History window for that document ⑥.

Finally, the claims in which this concept is used are also displayed. Detail of each claim can be accessed ⑦. 'Import' and 'Copy in' links are also provided ⑧ ⑨.

5.4.9 History (9/20)

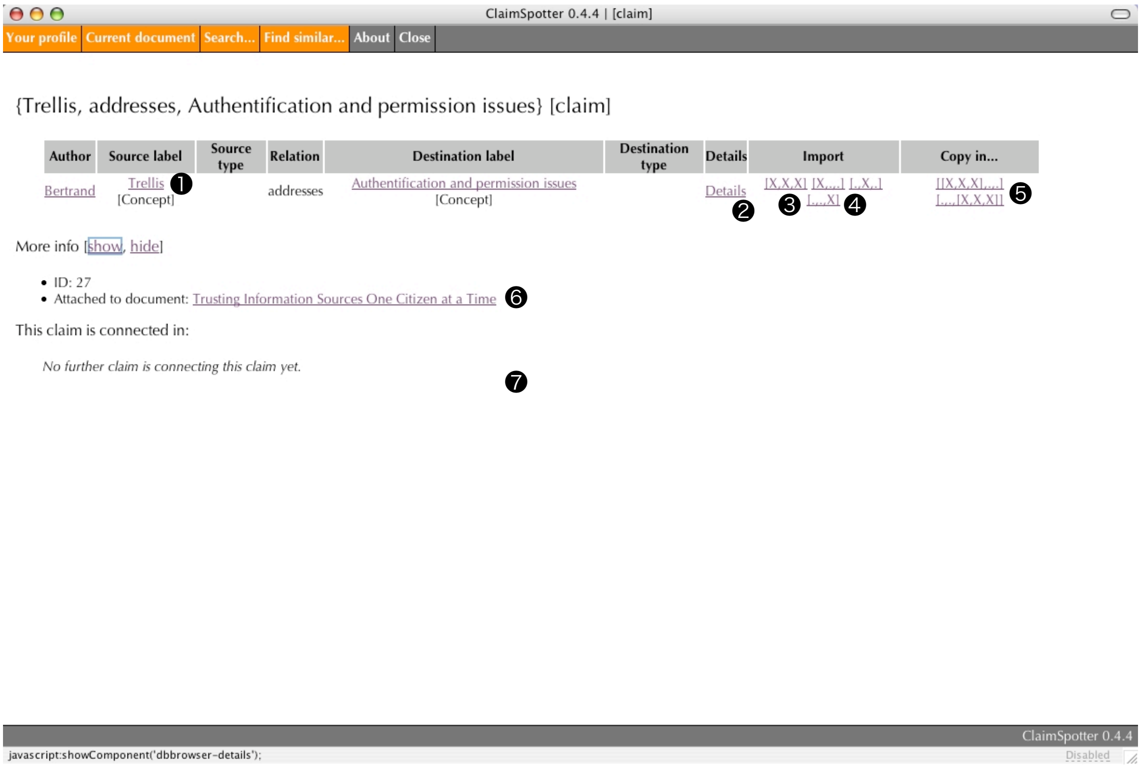


Figure 5.21: Virtual tour | History. (9/20)

This history window displays the detail of a claim. It can be used to answer questions such as ‘*Which concepts are involved in this claim?*’ and ‘*Has it been discussed in other claims?*’

Source and destination concepts history pages ① and additional detail ② for the claim can be accessed. Import buttons offer quick ways to duplicate the claim in its totality ([X,X,X]) if our annotator wants to express the same interpretation ③ and to create new claims with either the same source concept ([X,.,.]), relation type ([.,X,.]) or destination concept ([.,.,X]) ④ (leaving the other parts blank.)

The ScholOnto data model also permits a whole claim triple to be linked from another claim or to another claim [13]. The final set of copying buttons ([[X,X,X],.,.] and [.,.,[X,X,X]]) ⑤ can be used to copy an entire claim as a left or a right part of another claim.

[13]Examples of chained claims are given in section 2.1.5, page 18.

5.4.10 History (10/20)

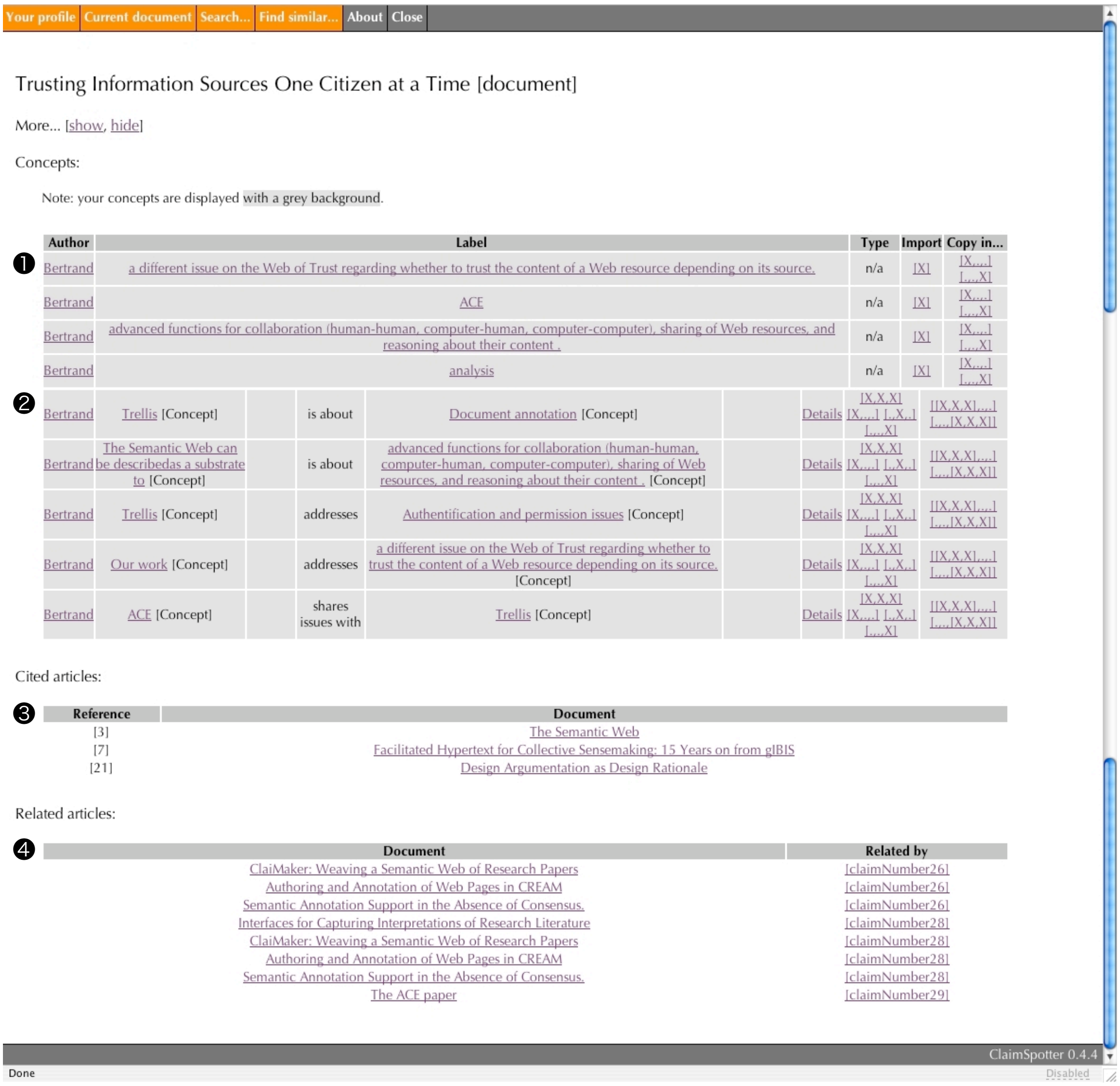


Figure 5.22: Virtual tour | History. (10/20)

Our annotator activates this history page to reveal a summary view [14] of this document. It includes the concepts ① and claims grounded in the document ②, the (manually specified) cited documents ③ and related documents ④ (updated automatically.)

[14]This screenshot is a composite view showing the top and bottom only of a larger window (hence the double scrollbar.)

5.4.11 Search (11/20)

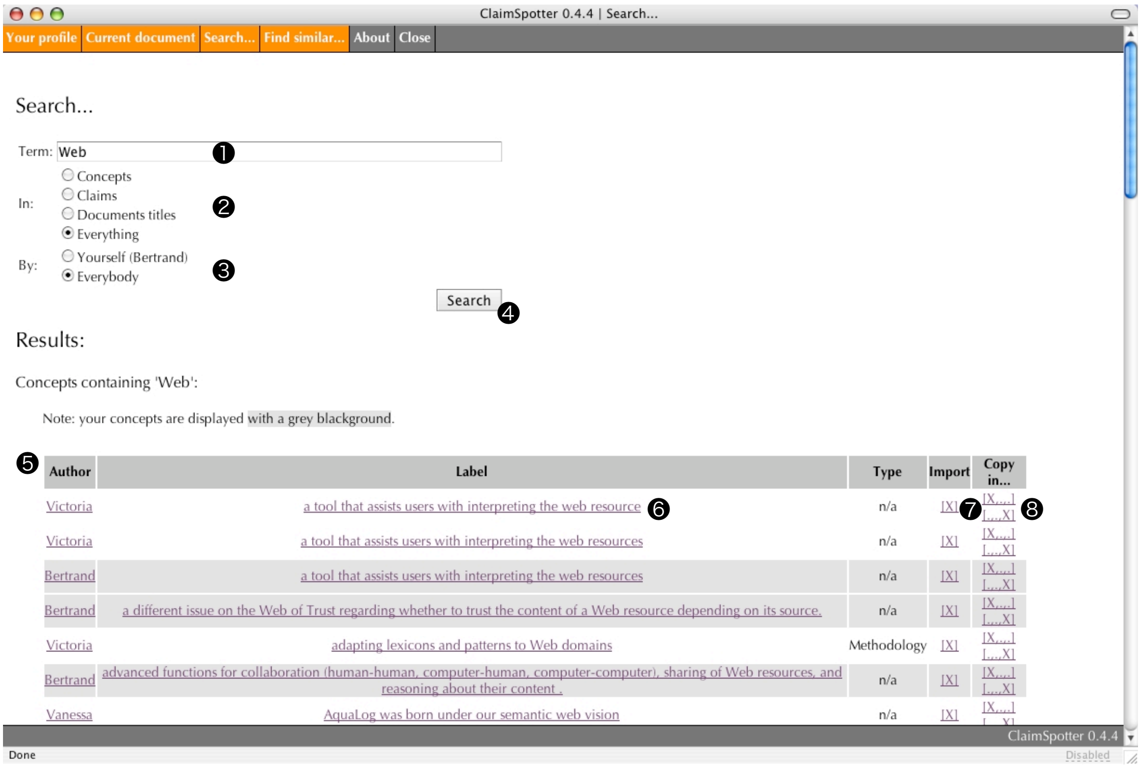


Figure 5.23: Virtual tour | Search. (11/20)

Our annotator can use a search window to look for ④ any combination of concept, claim, document title ②, defined by himself or by any of his peers ③, matching the given query expression ①. Results are displayed ⑤ with the usual layout that includes additional links to the history page of each object returned ⑥ and possibilities to import or copy the object in the current document input form (in the main window) ⑦ ⑧.

5.4.12 User profile (12/20)

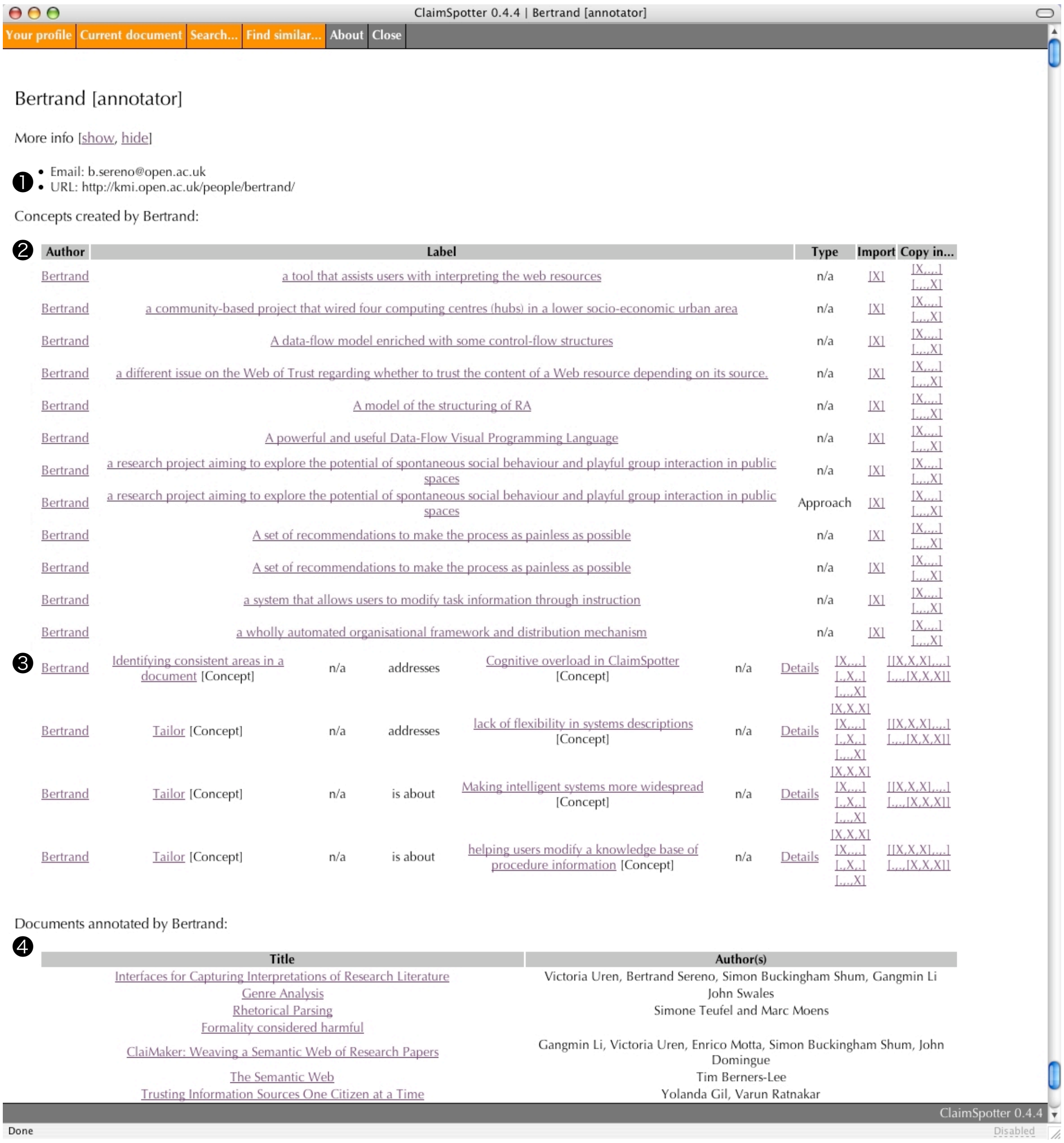


Figure 5.24: Virtual tour | User profile. (12/20)

The user profile reveals to our annotator his information ①, the concepts he has created ②, his claims ③, and the documents he has annotated (for which he has at least created a concept and/or a claim) ④.

5.4.13 Find similar. . . (13/20)

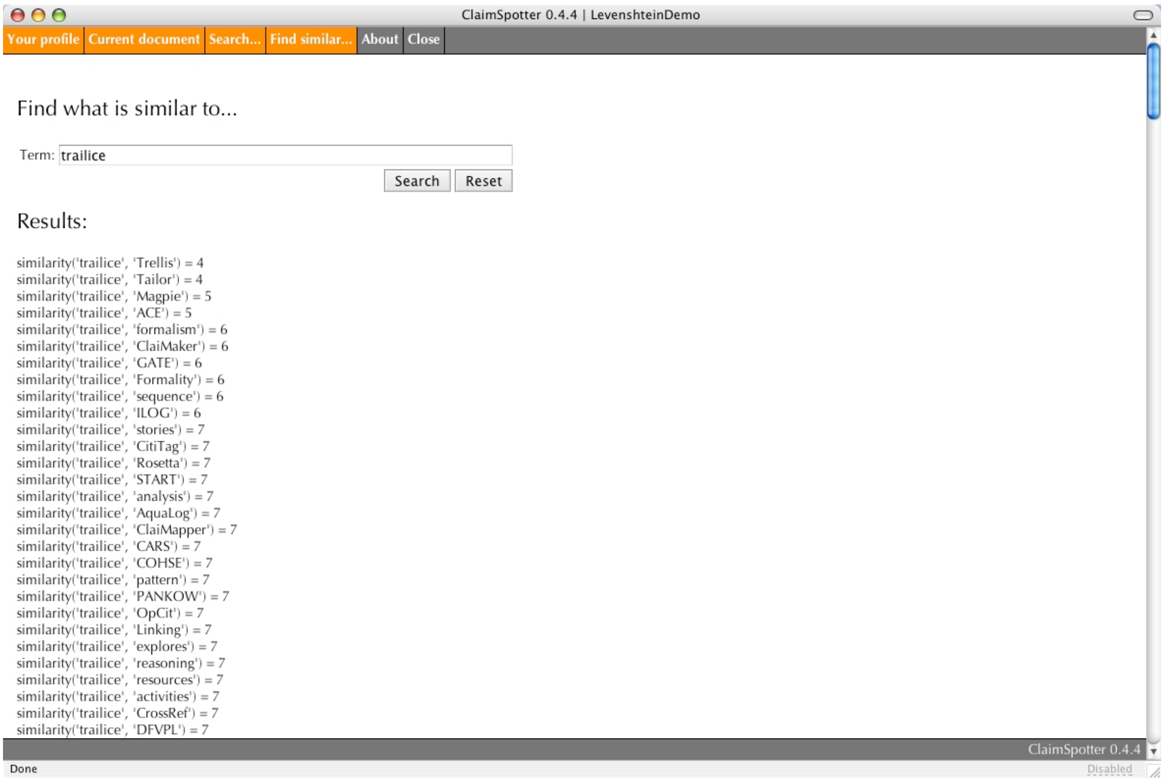


Figure 5.25: Virtual tour | Find similar. . . (13/20)

The system does not store duplicate concepts: if a concept already exists, it is reused and an instance of it is created for this particular combination of user and document. However, there may be times when one creates a new concept for which there is already a very similar concept (that could be advantageously be reused.) For instance, our annotator may want to create a concept ‘ontologies’ when there is already ‘ontology’ in the repository, or he may misspell the name of an existing project. In such cases, comparing the user input with existing concepts can be helpful. We have developed a module to perform this comparison, using an implementation of the Levenshtein algorithm [15].

[15] The Levenshtein algorithm compares two strings by computing the cost of transforming the first into the second one. The cost is expressed in terms of transforming operations applied to the original string to make it match the second (including operations like adding, removing or changing a character.) More information can be found at `http://www.levenshtein.net/`

5.4.14 More Ideas (14/20)

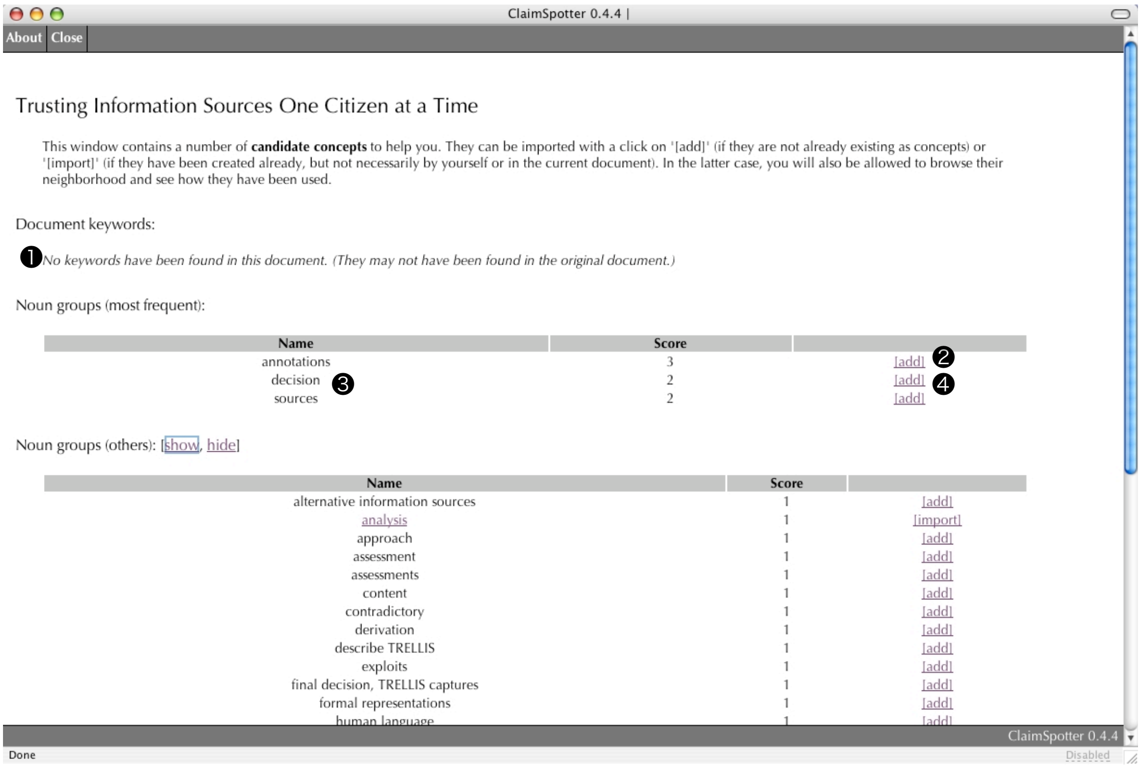


Figure 5.26: Virtual tour | More ideas. (14/20)

The candidate concepts suggested by ClaimSpotter can be accessed via the ‘More Ideas’ button in the main interface. The ‘More Ideas’ window lists the document keywords (manually stored in an XML file) and the most frequently found noun groups in the document. Each of these items can be imported by our annotator in the main annotation window with a click on the ‘[Add]’ shortcut button (and edited if desired.) If any of these expressions happens to be a concept already, the ‘[Add]’ button is replaced with ‘[Import]’ and the ‘History’ for the concept can be accessed (in the example, *analysis* already exists as a concept.) The aim of this module is to give additional ideas to annotators by proposing elements they may have overlooked in their interpretation.

5.4.15 Connect documents (15/20)

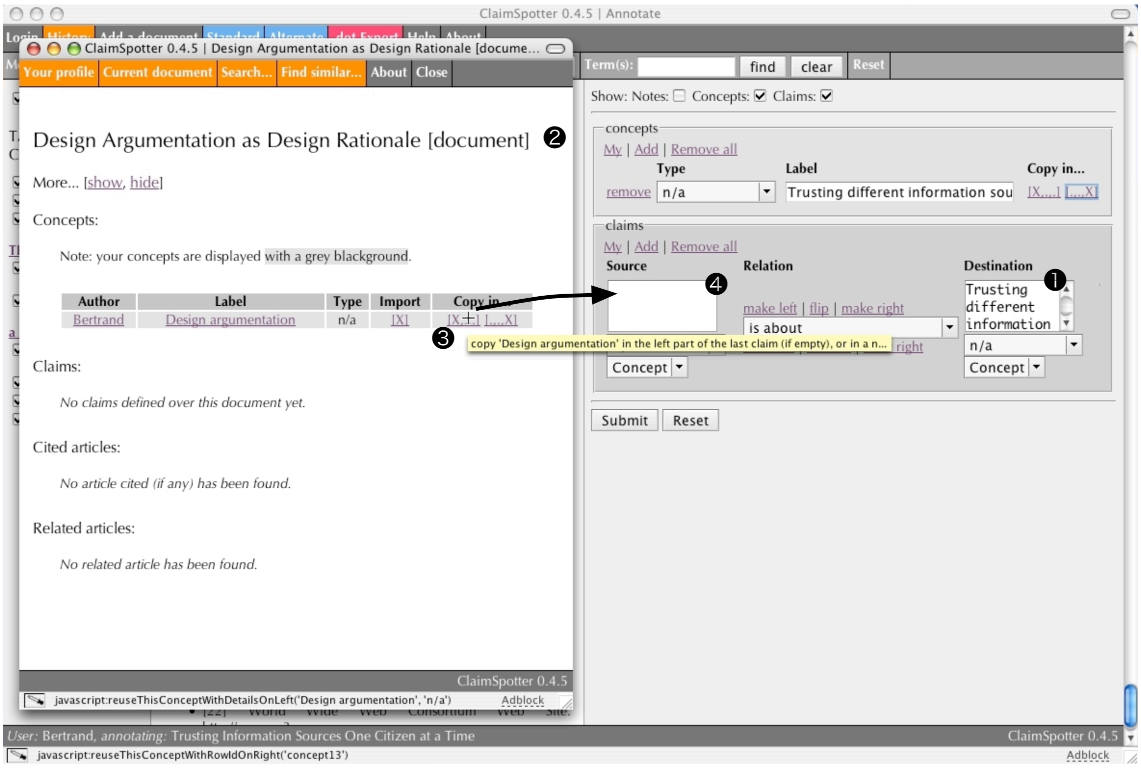


Figure 5.27: Virtual tour | 'Connect' documents. (15/20)

Documents can be 'connected' in ScholOnto via (1) shared concepts or (2) the definition of claims relating concepts defined in different documents. In this example, our annotator creates a claim relating a concept he has defined earlier, '*Trusting different information sources*' ①, and a concept that has been defined in another document ②. Using the shortcuts associated to the remote concept ③, he can combine concepts in a claim ④ and submit it. Upon submission, the current document and the document '*Design Argumentation as Design Rationale*' (from which the distant concept is imported) become 'connected', or 'semantically related,' in the ScholOnto repository.

5.4.16 Add a document (16/20)

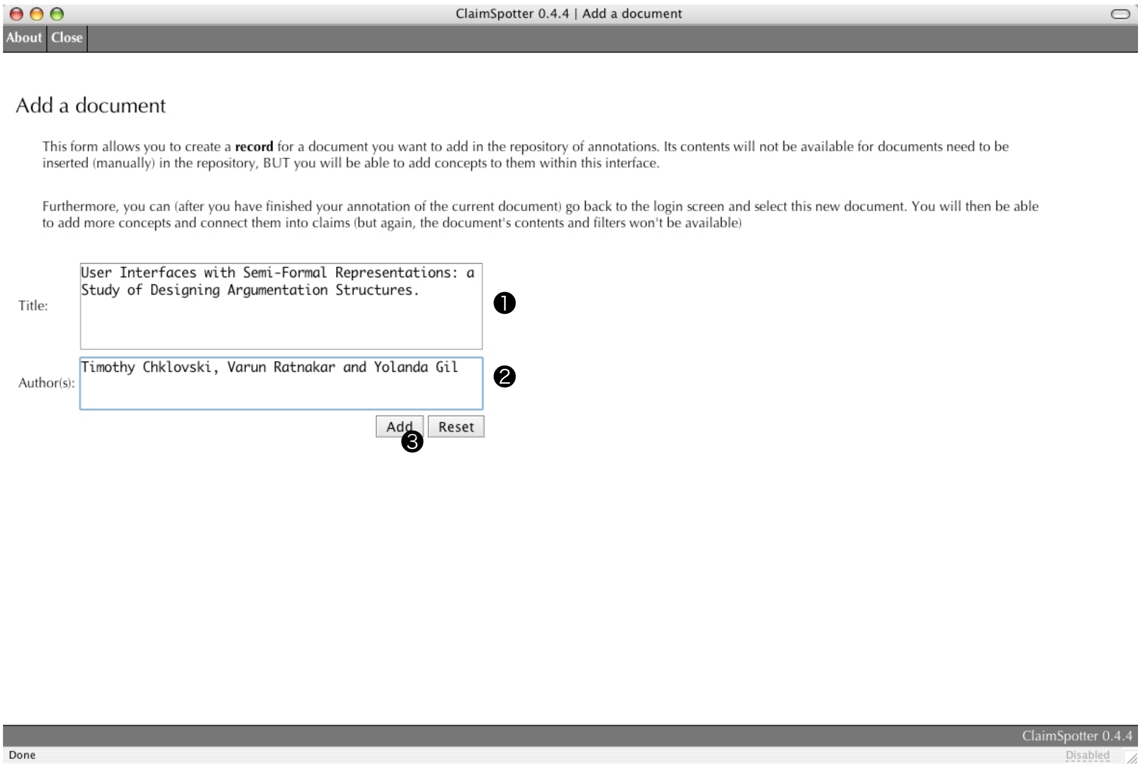


Figure 5.28: Virtual tour | Add a document. (16-1/20)

As there is no way in the current interface to upload an XML file representing the content of the document, a stub can be created to represent it [16].

[16]The ClaiMaker environment offers possibilities to upload personal bibliography databases, written with BibTEX for instance.

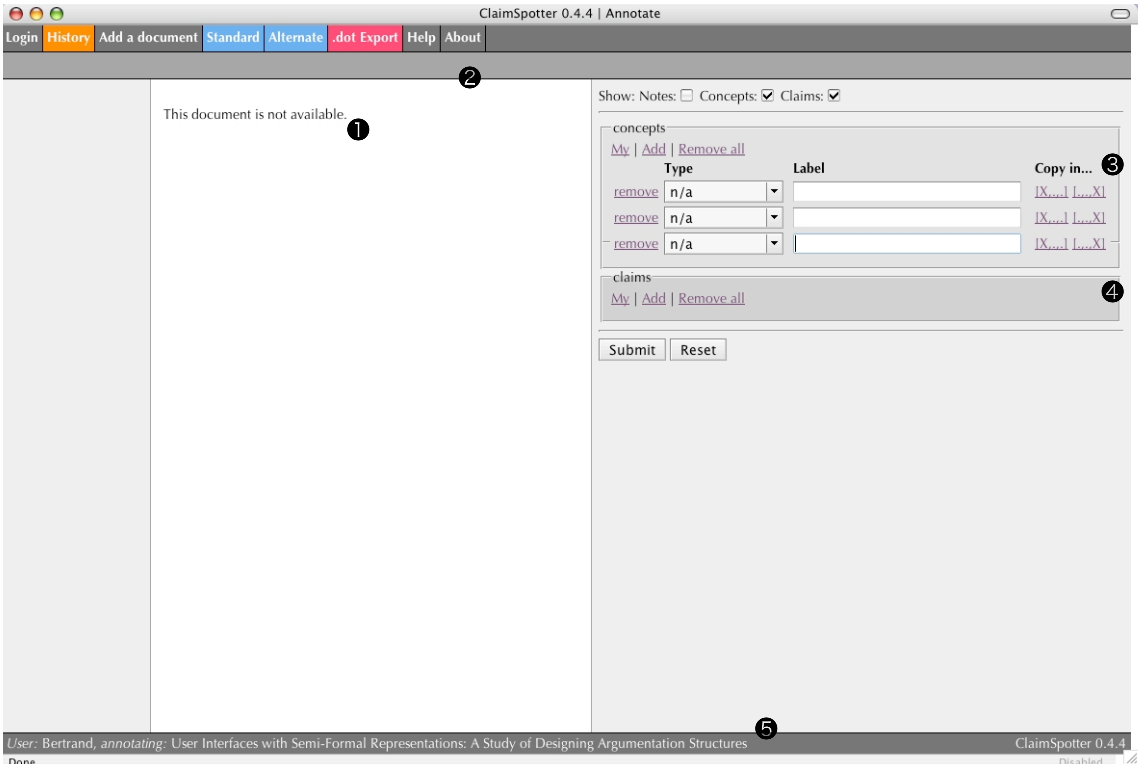


Figure 5.29: Virtual tour | Add a document. (16-2/20)

When only a stub is available, the 'spotting' toolbar is disabled ②, but the annotator can still define and combine concepts ③ and claims ④ for this document ① ⑤, connect it to external documents. . .

5.4.17 Views (17/20)

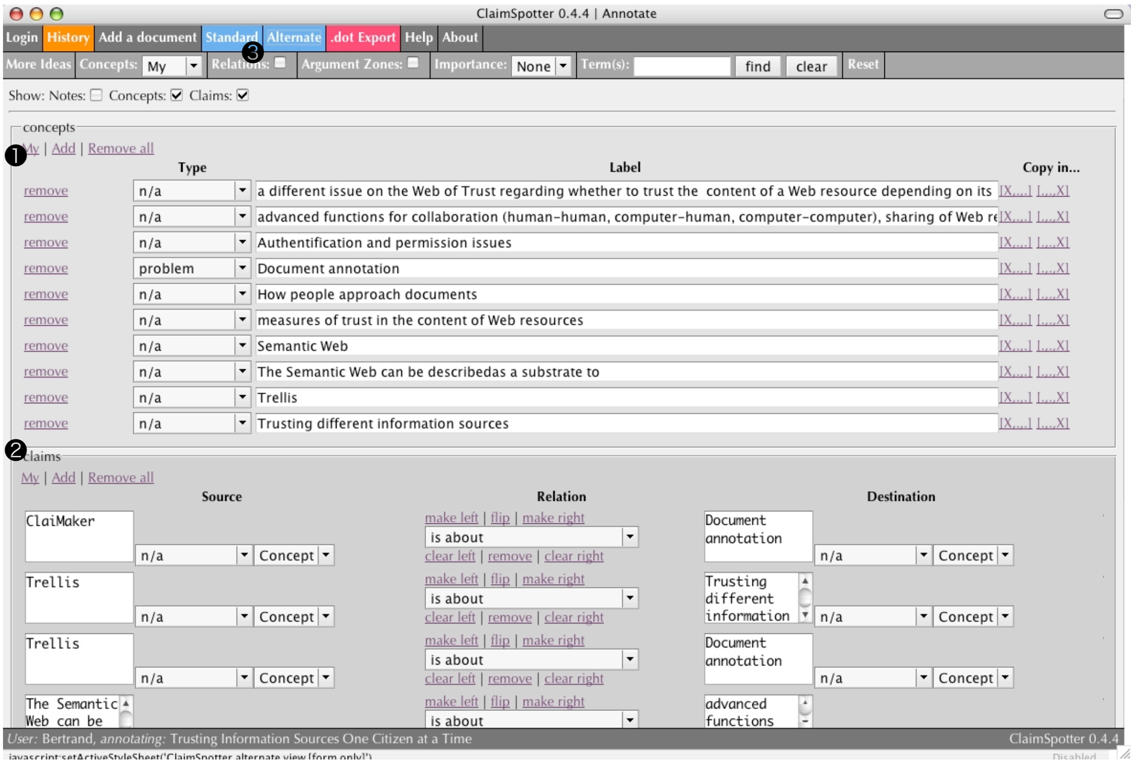


Figure 5.30: Virtual tour | Views. (17/20)

The view mechanism is a recent addition to ClaimSpotter. It experiments with the idea of displaying information in multiple ways, to suit different end-users. This alternate view hides the document and provides more space to create concepts ① and articulate them into claims ②. This view, focussing more on the annotation aspect and less on the interaction with the document and the 'discovery' of ideas, may be more suitable to reflective work, when annotators know what they want to record and look for the best way to express it in the formalism. It is possible to switch to the standard view at any time using the buttons in the general toolbar ③. This view is essentially the form-based view of ClaiMaker (c.f. page 25), but with multiple usability improvements.

5.4.18 Help (18/20)

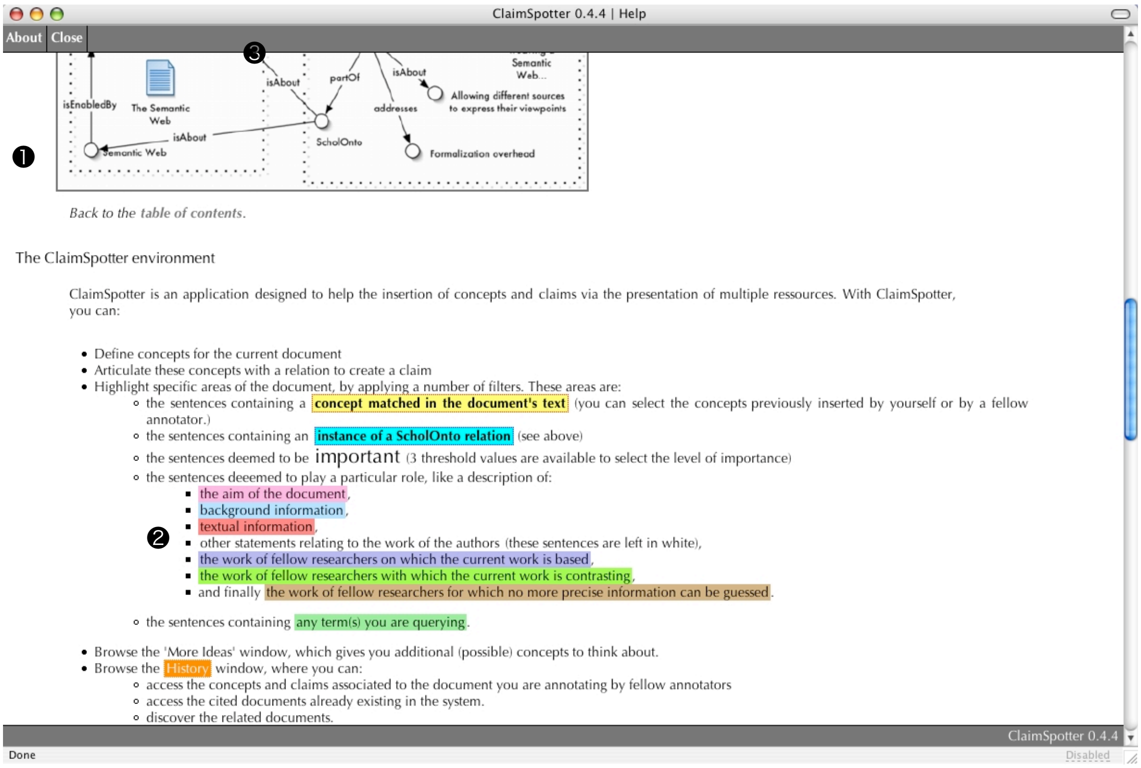
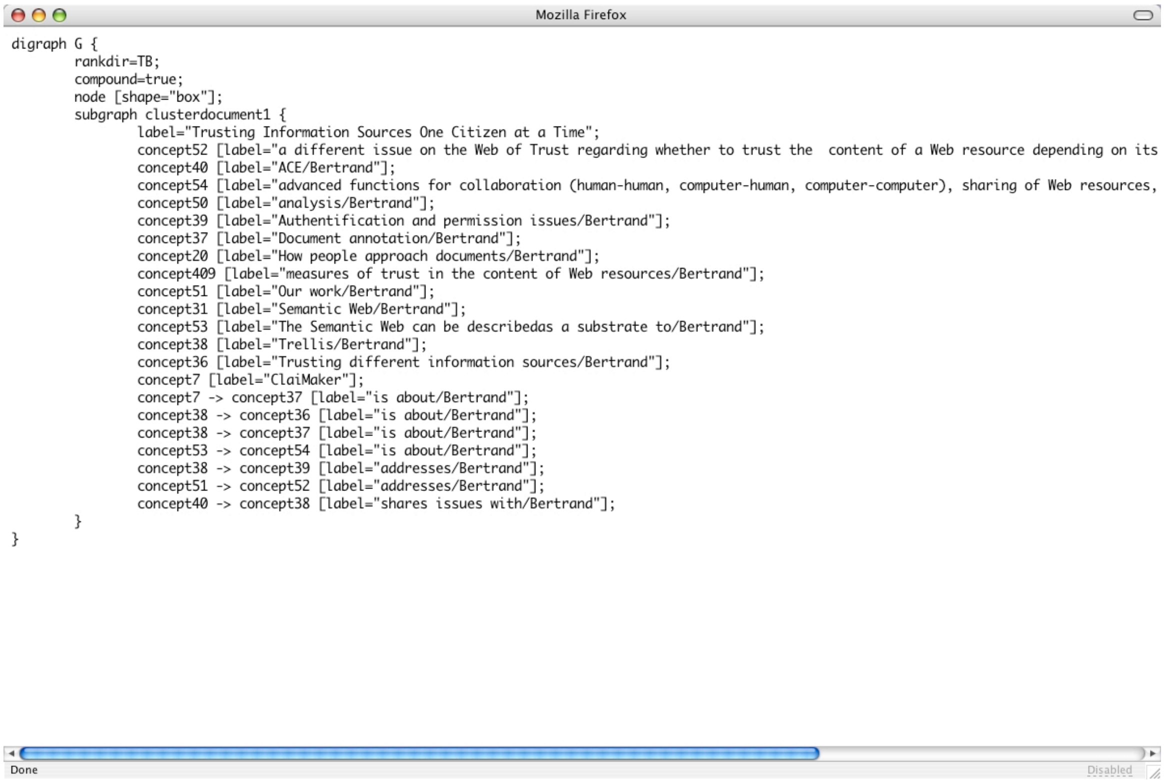


Figure 5.31: Virtual tour | Help. (18/20)

A 'Help' screen is provided, containing information about the ScholOnto language ①, the key for the different colours used by the spotting filters ② and instructions on how to perform common tasks such as inserting a concept or a claim.



```
digraph G {
        rankdir=TB;
        compound=true;
        node [shape="box"];
        subgraph clusterdocument1 {
                label="Trusting Information Sources One Citizen at a Time";
                concept52 [label="a different issue on the Web of Trust regarding whether to trust the  content of a Web resource depending on its
                concept40 [label="ACE/Bertrand"];
                concept54 [label="advanced functions for collaboration (human-human, computer-human, computer-computer), sharing of Web resources,
                concept50 [label="analysis/Bertrand"];
                concept39 [label="Authentification and permission issues/Bertrand"];
                concept37 [label="Document annotation/Bertrand"];
                concept20 [label="How people approach documents/Bertrand"];
                concept409 [label="measures of trust in the content of Web resources/Bertrand"];
                concept51 [label="Our work/Bertrand"];
                concept31 [label="Semantic Web/Bertrand"];
                concept53 [label="The Semantic Web can be describedas a substrate to/Bertrand"];
                concept38 [label="Trellis/Bertrand"];
                concept36 [label="Trusting different information sources/Bertrand"];
                concept7 [label="ClaiMaker"];
                concept7 -> concept37 [label="is about/Bertrand"];
                concept38 -> concept36 [label="is about/Bertrand"];
                concept38 -> concept37 [label="is about/Bertrand"];
                concept53 -> concept54 [label="is about/Bertrand"];
                concept38 -> concept39 [label="addresses/Bertrand"];
                concept51 -> concept52 [label="addresses/Bertrand"];
                concept40 -> concept38 [label="shares issues with/Bertrand"];
        }
}
```

Figure 5.32: Virtual tour | A *.dot* file can be generated, representing the concepts and claims submitted as a set of nodes and edges. This file can be rendered as a graph in a compatible application. (19-1/20)

5.4.19 Exporting as a graph (19/20)

An export option is also available from the interface. Clicking on the 'Export' button generates text output (c.f. figure 5.32) that can be saved in a file and loaded in a graph application to be visualised. The generated graph-file takes as input any combination of document and annotator, and generates a *.dot* file [17]. Figure 5.33 gives an example of a graph generated by an application reading *.dot* files.

Additional export options A similar export option could be created to generate an XML map for ClaiMapper (c.f. page 27) - in which it could be edited further - or for the ClaimFinder system (c.f. page 27) which also provides a self-organising graph layout, with zoom, filtering and rotation options [Buckingham Shum et al., 2005].

[17] A *.dot* file describes a set of nodes and the edges connecting these nodes. More information can be found at `http://www.graphviz.org/`

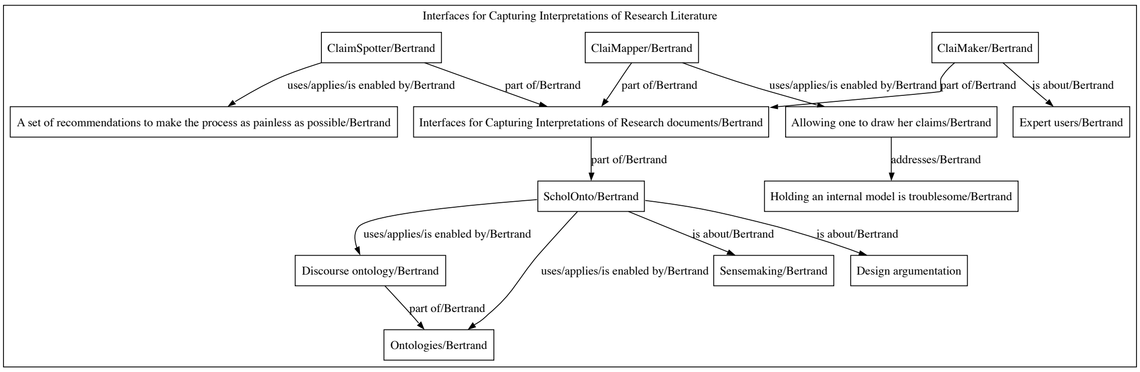


Figure 5.33: Virtual tour | Concepts and relations defined by *Bertrand* over the document '*Interfaces for Capturing Interpretations of Research Literature.*' (19-2/20)

5.4.20 Notifications (20/20)

Notification (another aspect we have identified in our design phase to support the transition from scholarly documents to ScholOnto annotations, c.f. figure 4.26, page 101) is supported via the definition of RSS feeds. RSS [18] is a '*format for syndicating news and the content of news-like sites.*' The information published to the ScholOnto repository is serialised and exported as feeds which can be subscribed to by annotators. A feed is generated on the fly: it contains the most recent information about any item in the repository. Feeds are available for any:

- *Concept*: the feed contains answers to questions '*Who has reused this concept?*', '*For which document?*' and '*Has it been used in a claim?*'
- *Claim*: the feed contains the latest claims reusing it. Annotators are notified if one of their claims is challenged.
- *Document*: the feed contains the newest concepts and claims defined over it. Annotators are notified of any added interpretation.
- *Annotator*: the feed contains the newest concepts and claims submitted by this annotator. The models of a particular annotator can be tracked.

ClaimSpotter feeds are generated in the RSS 2.0 variant of the language. Appendix B gives the RSS feed (as of the 3rd of February 2005) for the document we have used in this virtual tour. RSS feeds can be read in a newsreader application (c.f. figure 5.34.) Although not a central part of this thesis, this illustrates the potential for future investigations into distributed semantic scientific publishing and alerts.

[18] Quoted from `http://www.xml.com/pub/a/2002/12/18/dive-into-xml.html`

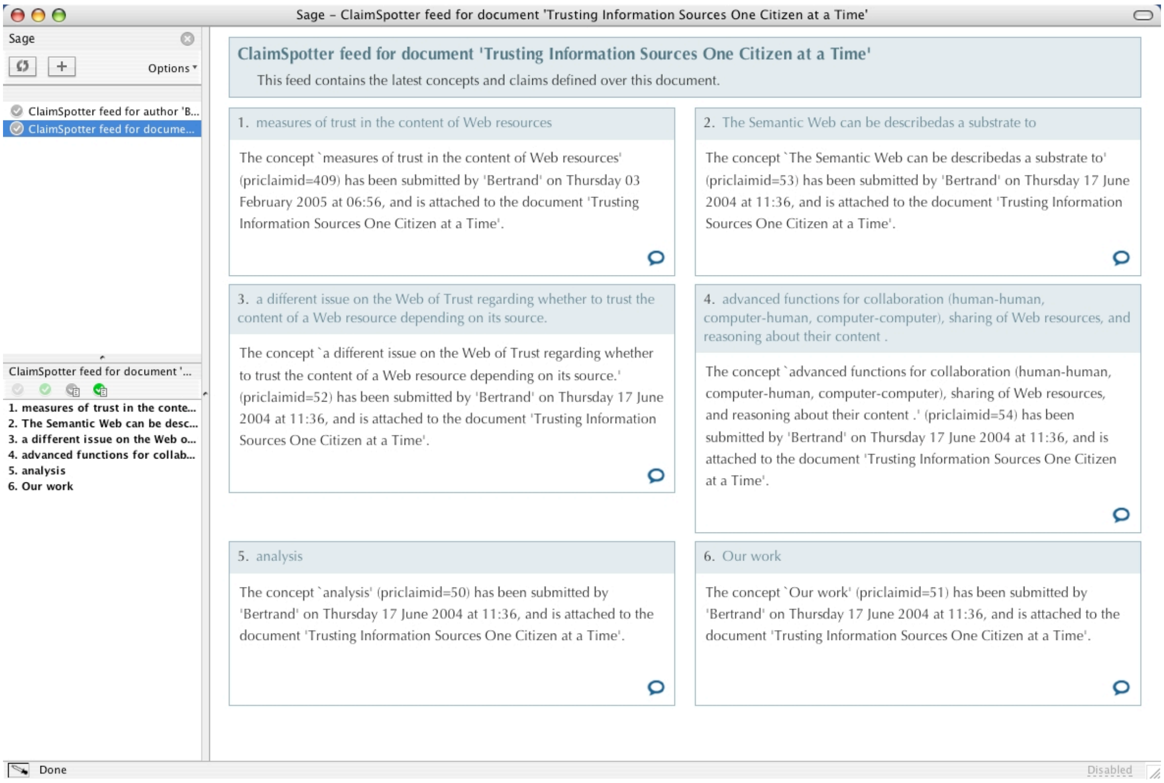


Figure 5.34: Virtual tour | RSS feed in ClaimSpotter (20/20)

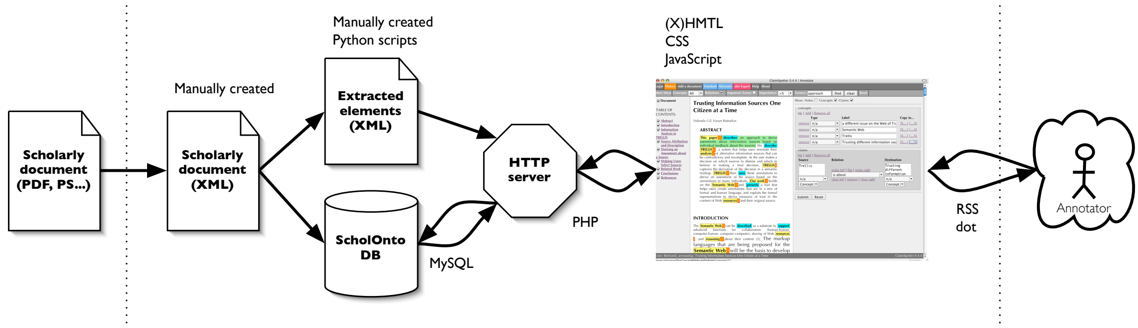


Figure 5.35: Technologies involved in ClaimSpotter, from the scholarly document to the end-user.

5.5 Implementation

ClaimSpotter is a Web application developed and tested on a Mozilla-based Web browser (c.f. figure 5.35 for a presentation of the different technologies involved.) The browser interface is used to send requests and process the output of a set of PHP functions, run on an Apache Web Server.

Stylesheet The choice of Mozilla is guided by the need to rely on proper implementations of the CSS and CSS2 stylesheet languages and of ECMAScript (JavaScript.) CSS is used throughout the interface to present the information contained in the HTML pages generated by the PHP engine. ClaimSpotter runs partially on Safari. It has not been tested under Internet Explorer.

User-side interaction JavaScript is used to dynamically modify the presentation of the different sources of data (for instance highlighting an existing concept in the document or adding a row in the claim input form.) It is also used at a broader level to activate the different views of the interface: each view is assigned a stylesheet that is replaced on the fly when the user requests it.

Server-side interaction Web pages combine static and dynamic information extracted from the document and/or generated from the ScholOnto database. This database runs on MySQL.

This choice of technologies is dictated by the availability of previous work on the database infrastructure [19] and by previous developing experience. We do acknowledge that other possibilities would have also been possible, and even beneficial. For instance, a combination of

[19]Thanks to Gangmin Li for the ScholOnto MySQL architecture.

the Semantic Web languages RDF and OWL could have been used.

RDF [20] is an approach to represent data as triples, and as such would seem to be perfectly suitable for our ScholOnto triples (concepts would be described by their URI which could combine the URL of the (X)HTML representation of the scholarly document considered and a unique anchor containing the concept label.) OWL [21] would be suitable to represent the meaning of the different terms in the triples (more specifically, the scholarly relations), by recording for instance the different categories each relation belongs to ('supports', 'general'..., c.f. section 2.1.2, page 14.)

Syntax considerations aside, the real benefit provided by a joint usage of OWL and RDF would lie in the reasoning facilities which could be developed over OWL, independently of the knowledge domain we deal with. These facilities being difficult and time-consuming to build, a translation from MySQL would enable us to reuse the ones developed for other domains. The point of this thesis does not lie in the reasoning facilities which can be offered; we shall nevertheless give an example of a trivial reasoning which could be applied over data: a triple {*conceptA*, ***isAbout***, *conceptC*} could be generated based on the existence of two triples {*conceptA*, ***isAbout***, *conceptB*} and {*conceptB*, ***isAbout***, *conceptC*}; another example could involve the creation of a triple {*conceptD*, ***isAbout***, *conceptE*} for each and every {*conceptD*, ***xxx***, *conceptE*} triple created since if *conceptD* is connected to *conceptE* with any discourse relation it can be assumed that *conceptD* is also about *conceptE* (i.e., that there is something 'putting them in relation.') A thorough discussion of the semantics of each and every discourse relation in the ScholOnto ontology would have to be performed to assess whether such transitive rules can be applied.

5.6 Conclusion

We have presented in this chapter ClaimSpotter, an active document-centric environment to support the semantic annotation of scholarly documents with concepts and claims. ClaimSpotter (i) is built on an open, extensible architecture which can incorporate new text analysis components as required; (ii) uses text analysis techniques to overlay annotations onto the original text to draw attention to sections which may be particularly significant; (iii) offers facilities to filter and navigate the document in novel ways; (iv) facilitates the recording of new semantic annotations or the reuse of existing ones; and (v) provides pointers to related documents and annotations based on connections mediated by semantic annotations.

[20]More information available at `http://www.w3.org/TR/rdf-concepts/`.
[21]More information available at `http://www.w3.org/TR/owl-features/`.

As we have mentioned at the end of our design phase (chapters 2, 3 and 4), our goal (based on our understanding of annotators' motivations and expectations) is to provide a 'toolbox' of possibilities to help annotators build their networked structure of argument. With the ClaimSpotter prototype running, our task, as designers, is now to assess whether we are on the right track or not, that is, to discover *how* the elements of this toolbox, these possibilities, are used. This is the focus of our next chapter.

Chapter 6

Evaluation

To realise our test phase, we need to answer our second research question (c.f. page 6):

(ii) In what way is the process of interpreting a document, through the elicitation of concepts and their articulation in claims, influenced by the possibility to access its content in an interface, to visualise it and to modify its representation, and by the availability of additional resources?

This can be achieved using several techniques. A questionnaire can partially answer this question but cannot uncover the aspects we are mostly interested in: the interaction, the dialogue between the user and the interface. A careful observation of end-users' interactions is a more promising approach: analysing and understanding as much as possible their actions, their choices and their reaction to these choices can give us the level of detail needed to discover the influence of our document-centric annotation environment. Only via a close observation of the interactions with the environment can we understand the impact and influence of, say, the spotting filters on annotators' actions.

We first break our research question into smaller, more manageable, ones in the next section. We then present the conditions in which this evaluation takes place, and introduce our theory of ClaimSpotter usability. We also present a statistical analysis and the answers to a post-experiment questionnaire.

6.1 Second research question

Our first step is to make our research question somehow easier to approach by studying its components and viewing it from different angles. Let us restate it:

(ii) *In what way* is the process of interpreting a document, through the *elicitation* of *concepts* and their *articulation* in *claims*, *influenced* by the possibility to *access* its content in an *interface*, to *visualise* it and modify its *representation*, and by the *availability* of additional resources?

This time, we have put several elements in italics.

6.1.1 *'Access', 'interface', 'visualise', 'representation', 'availability'*

The first group of words we have italicised can be gathered around the notion of user interface and more precisely ***interaction design***. The maturity of the Human-Computer Interaction field has clearly established the importance of a user interface in any task, an importance that is all the more important for sense-making tasks requiring extensive manipulation of symbolic representations. Our research question can be reformulated as '*Is the interface helpful?*' To answer this question, we can look at multiple aspects.

'Pleasure' The 'pleasure' users have when using the interface is important, especially in a context in which the annotation may not be straightforward and may require some time spent on thinking about the correct way to do it. In this case, making the experience pleasant (and even fun) is an important part of the potential success of an environment.

Features We want to discover which features 'work' and which ones 'do not work.' We also want to find out if they are used in the way we have thought they would be: it may be that some unexpected uses of a feature, or of a combination of features, emerge from annotators.

Intuitiveness The intuitiveness of the interface also has to be studied. We want to find out if it discloses its functionality to new users (both new to the interface itself, and to the underlying formalism, e.g. the ScholOnto language) and if it behaves the way annotators expect it to behave.

Consistency Consistency of the interface is another important notion. We want to find out if, for instance, multiple terms are used to incoherently refer to a single object.

Feedback Letting users know where they are and what the function they activate does is important. Visual feedback is needed to show that the interface reacts to their actions in a proper way. It can take several forms, from the activation of a spotting filter resulting in the display of relevant pieces of information, to warning signals when they are about to perform a potentially hazardous operation such as the deletion of a concept or of a claim.

These design principles [Dix et al., 2004] help us break our initial ***interaction design*** question into the following ones which guide our analysis:

- Is the interface helpful?
- Which features work and which ones do not?
- Is the interface intuitive and consistent?
- Is it giving feedback when needed?

6.1.2 *'In what way', 'influenced', '?'*

The second group of words we have italicised is composed of *In what way*, *influenced*, and the question mark sign, as in *In what way is the process influenced?* These words refer to the notion of ***strategy***.

Finding out if (and how) the environment influences the annotation process may be noticed in the way it is broken down cognitively. To fully answer this question, we would have needed to compare an end-user's annotation process in ClaimSpotter to the same end-user's annotation process, with the same amount of knowledge about the document considered, but without the assistance of the environment. As annotation requires interpretation - an aspect that, as we have seen before, is likely to be influenced by multiple factors, including previous annotations - it is very difficult to perform this comparison in a controlled experiment. This is a long recognised problem in traditional experiments in HCI design. Our strategy to find the 'added value' ClaimSpotter brings is to ask annotators to think about their concepts and claims prior to an annotation experiment (presented later) and to compare it with what they have actually done, using a post-experiment questionnaire.

'Quantity' of annotation The first difference we may notice with the use of ClaimSpotter is related to the number of concepts and claims submitted: does ClaimSpotter encourage annotators to say more, or less, about the document? (compared to what annotators may have done without it.)

We should restate that in our context, more is not better, nor less is worse. Meaningful annotations may be expressed in a very few concepts and claims, if they are expressive enough and if they translate the annotator's opinion (c.f. the annotations taxonomy in section 2.2.2, page 21.) On the other hand, multiple concepts and claims may be submitted to represent background information: although these can be useful, they do not represent the annotator's opinion about the document, and therefore contribute less to the debate in the claim space.

Quality of the annotations The second difference is related to, as we have just seen, the quality of these annotations. While our interest in sense-making processes is not focussed on 'good' or 'bad' annotations, there is however a degree of commitment in the annotations that makes them more or less valuable. A 'valuable' annotation represents a personal opinion; a 'less valuable' annotation is more descriptive and less committed. The presence of suggestions and the availability of peers' annotations may help annotators expressing claims that are more 'valuable.' For instance, if an annotator sees one of her peers having submitted a committed statement, she may feel more confident about doing it herself too.

Strategy We also want to get some understanding on why annotators may say more and/or better things by studying the strategies they adopt to formulate their concepts and claims. We want to find out if the presentation of the document and the possibility to access the different sources of knowledge available shape the way they approach the document. We want to find out how annotators kick-start their annotation process (from the document itself? from the repository of concepts and claims submitted by their peers?) To summarise, we want to find out *in what way* the environment shapes the annotation process. We can reformulate this ***strategy*** question as:

- Do annotators say more with the environment?
- Do annotators say 'better' (i.e. submit 'more committed statements')?
- What are the strategies adopted by the annotators?

6.1.3 *'Elicitation', 'concepts', 'articulation', 'claims'*

The third and last group of words we have highlighted in our second research question is related to the *elicitation of concepts and their articulation in claims*, in other words to the notion of ***formalisation***. This is the end activity of the interpretation process. When an annotator has decided which level of commitment to adopt (e.g. stating background knowledge or modelling her opinion), she has to decide what to use as concepts, how long and detailed they should be and which relation type to use to articulate them [Shipman and McCall, 1994, Buckingham Shum et al., 1997].

Concepts As they are unconstrained in their expression, this freedom may create some difficulties. Another potentially interesting aspect is related to the optional (and this time,

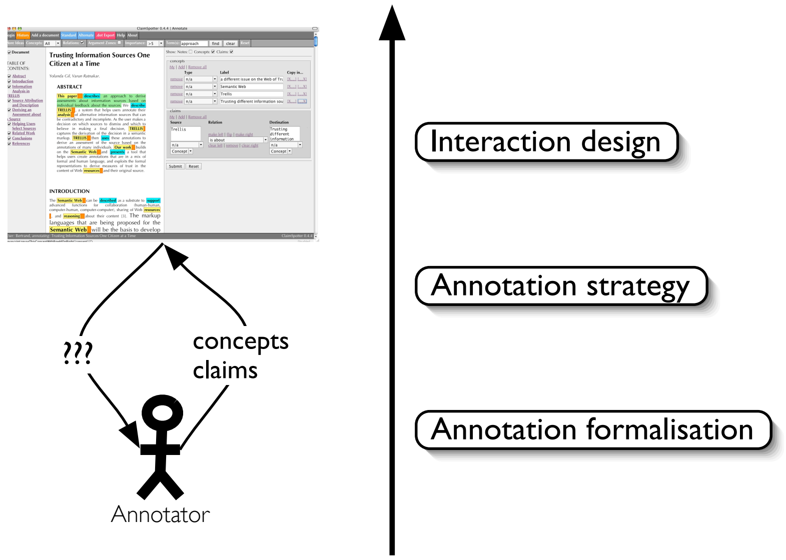


Figure 6.1: In an extended version of our human-computer information flow (c.f. figure 1.2, page 6), we propose to take into account three dimensions to assess the usability of the ClaimSpotter interface: interaction design, annotation strategy and annotation formalisation.

constrained) concept type one can add to a concept. It gives us the opportunity to examine what happens when annotators have to make a choice from a fixed list of types.

Claims To formulate their claims, annotators have to choose the relation they want to use in the ontology. Again, this constraint may create some difficulties. Decisions taken by annotators in this situation need to be captured to understand the annotation process more accurately.

To understand the ***formalisation*** aspect of our research question, we can focus on the following questions:

- How do annotators formalise their concepts?
- How do annotators formalise their claims?

6.1.4 Definition of 'ClaimSpotter usability'

The three themes we have identified are different facets of the overall usability dimension that we seek to assess. Figure 6.1 organises them in a pyramid. Usability is related to the notions of user interface and interaction design, for we are concerned about the presentation of the information and the uses the interface offers. It is also related to the annotators' strategy as the possibilities offered may influence their approach to the task. It is finally concerned with formalisation, as annotators have to break their ideas into concepts and articulate

	Keywords	Theme	Questions
ClaimSpotter usability	*Access, interface, visualise, rendering, availability*	Interaction design	Is the interface helpful?
			Which features work and which ones do not?
			Is it intuitive and consistent?
			Is it giving feedback when needed?
	In what way, influenced, '?'	Annotation strategy	Do annotators say more with the environment?
			Do annotators say 'better' (i.e. submit 'more committed statements')?
			What are the strategies adopted by annotators?
	Elicitation, concepts, articulation, claims	Annotation formalisation	How do annotators formalise their concepts?
			How do annotators formalise their claims?

Table 6.1: Our second research question is viewed through three lenses: interaction design, annotation strategy and annotation formalisation. For each of them, several questions guide our evaluation study.

them into claims. We summarise in table 6.1 the multiple questions that we have identified and associated to these themes, questions that we aim at (at least partly) answering in our evaluation study.

6.2 Experimental protocol

To answer these questions, we evaluate ClaimSpotter using a combination of statistical, observation and query techniques.

We study the interactions of thirteen users of ClaimSpotter in a routine task: the annotation of a scholarly document with the ScholOnto language. Their task is to annotate a short paper - that either (and preferably) they have written, or that they are at least very familiar with - with concepts and claims. This is a strong requirement, as we expect users to already face a lot of new information to digest (including the ScholOnto model and the ClaimSpotter environment.) By ensuring they already know what the document they annotate is about, we hope to reduce this overload. A brief ScholOnto introduction is sent by email prior to the experiment. They are also asked to think about a few concepts and claims before the experiment.

These users are researchers, of whom ten are PhD students, two are research fellows and one is a professor. Four of them are ScholOnto experts (members of the project team and, therefore, familiar with the ontology of relations) but have no training with the interface. The remaining nine are beginners with both the ScholOnto formalism and the interface. Each session is limited to one hour at most, and they are spread over a period of one month [1]. Annotators are free to spend less time on it, if they feel they have said all they have to say.

[1] The experiment took place in August 2004.

Time (hr)	Event
Earlier	Preparation: each participant has to identify the paper she will annotate. Choosing a short paper is encouraged. An electronic version must be sent to the team for preparation and formatting in the appropriate XML format.
	A brief ScholOnto presentation is sent by email. Each participant is asked to think about a few concepts and claims in advance.
	The tutor prepares a few concepts and claims for the paper, to give initial aspects to 'react against' to the annotators.
0:00	Beginning of the annotation process.
	Presentation of the interface and of its features (and optionally of the formalism for beginners.)
0:15	The participant starts to work on her paper. The tutor remains in the room to answer any question she may have.
1:00	End of the annotation process.
Later	Questionnaire sent to participants.

Table 6.2: Steps of our experimental protocol.

Each participant is given an initial tour of the application and of its features. It lasts between 15 and 20 minutes, leaving between 40 and 45 minutes for the annotation itself. Beginners are given an additional presentation of ScholOnto at the same time. Table 6.2 summarises the different steps of the experimental protocol.

A tutor - the author of this dissertation - is present throughout the session to provide assistance when needed, but also to engage discussion when suggestions are made. Participants are asked to explicitly think aloud and voice any question or concern they have.

Their actions (including their vocal reactions to the interface) are captured with a screen capture application. The resulting files are approximatively 200 MB. each. We also record the tutor's presentation of the tool (and optionally of the ScholOnto approach) in order to capture potentially interesting comments during these initial discussions.

6.3 Statistical analysis

We begin our evaluation with a presentation of the salient facts extracted from the statistical analysis we have performed on the literature models submitted by annotators.

6.3.1 Summary

257 concepts and 160 claims are submitted by the 13 participants during the evaluation, giving on average 19.8 concepts and 12.3 claims for each of them, with no major difference between the pool of 4 experts and the pool of 9 beginners: the former input slightly more concepts (a mean of 20.75 against 19.3) and claims (a mean of 14.75 against 11.2), but these

	a1	a2	a3	a4	a5	a6	a7	a8	a9	a10	a11	a12	a13
EXPERT/BEGINNER													
e/b	e	e	e	e	b	b	b	b	b	b	b	b	b
CONCEPTS total: 257 (mean: 19.8); experts: 83 (20.75); beginners: 174 (19.3)													
# submitted	8	17	29	29	27	17	21	5	31	33	17	12	11
CLAIMS total: 160 (mean: 12.3); experts: 59 (14.75); beginners: 101 (11.22)													
# submitted	5	12	20	22	11	7	21	4	18	11	12	6	11

Table 6.3: Each annotator is given an anonymous id, from a1 to a13 (these ids do not correspond to the ones given in chapter 3. However, most participants in the paper study have also taken part in this evaluation.)

differences are not meaningful. The least productive annotator, a beginner, inputs 5 concepts and 4 relations, while the most productive, an expert, inputs 29 concepts and 22 relations. Three beginners also input a great number of concepts and claims - a7, a9 and a10. Table 6.3 gives the breakdown for each annotator, including their level of experience (beginner or expert.)

6.3.2 Concepts

The tutor bootstraps the annotation of each document by providing a small number of concepts (between 5 and 8) and claims (between 2 and 4) to reflect upon and/or react against (c.f. table 6.2.) Data for this experiment is presented in appendices C (c.f. page 251), D (c.f. page 261) and E (c.f. page 269.) Concepts and claims inputted (by the tutor) before the evaluation study are not included in our analysis; they can be consulted in appendix E (c.f. page 269.)

Length

Figures 6.2, 6.3 and 6.4 give the distribution of the length of the concepts (expressed in their number of words) for all the annotators, the beginners and the experts, respectively. Most concepts are short: they are composed of one, two or three words. 164 out of 257 concepts (64%) submitted are three or less than three words long [2].

Beginners vs. Experts The overall shape of the graph for these two categories is broadly similar, and short concepts are as frequently submitted by novices as by experts: 115 concepts out of the 174 (66.1%) concepts submitted by beginners, and 49 concepts out of the 83 (60.2%) submitted by the experts are composed of three words or less. There is a peak on

[2] It is worth mentioning that shorter concepts may be found in descriptive and fine-grained annotations, which we have presented in our study of annotation goals (c.f. section 2.2, page 19.)

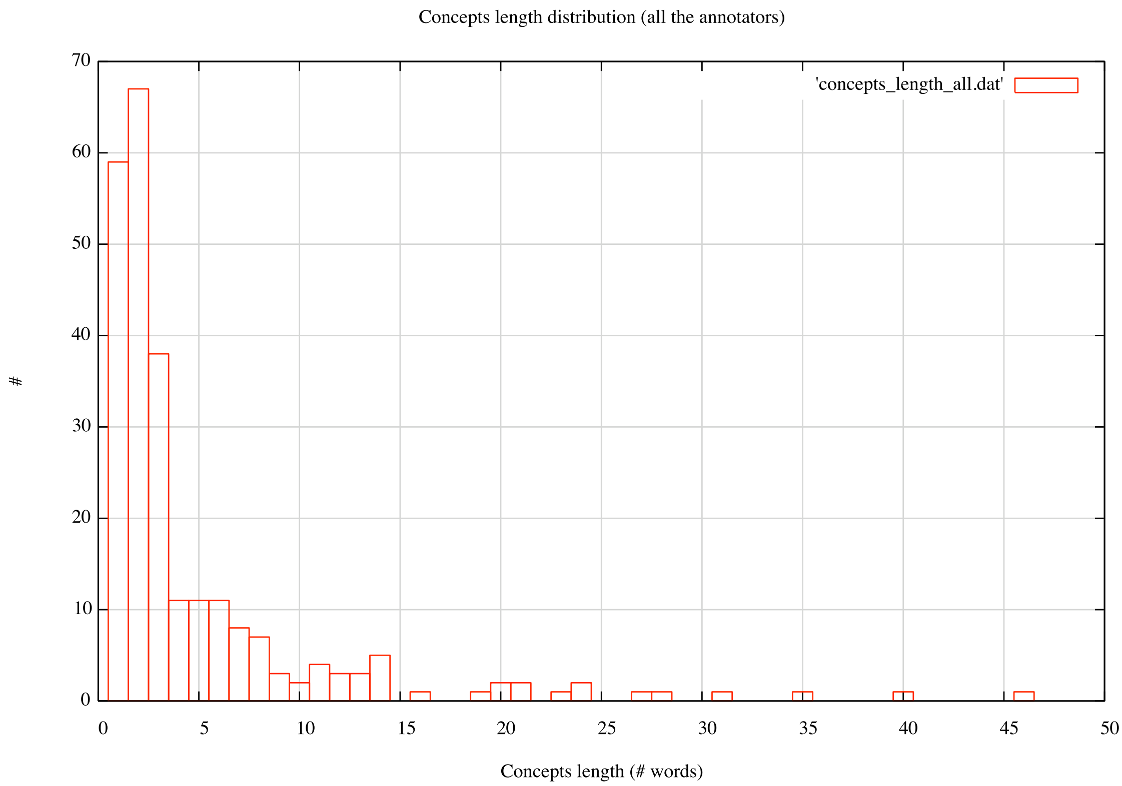


Figure 6.2: Concepts length distribution graph for all the annotators. 164 concepts out of 257 are composed of three words or less.

the experts graph for two-word concepts: expert a3 creates 19 two-word concepts but, on the other hand, only two one-word concepts.

Short concepts Table C.2 (c.f. page 258) lists the concepts containing three words or less. Typical concepts in this subset include proper nouns and acronyms indicating approaches, technologies and project names, such as *CitiTag*, *COHSE*, *Semantic Web* or *Science Citation Indexes*. They also contain noun groups such as *hypertext narrative* or *hypertext discourse coherence*. These elements are 'extractable', via techniques we have reviewed previously, and their importance (they represent two thirds of the concepts submitted) emphasise the relevance of a 'candidate concepts' spotting filter [3].

Long concepts It is more difficult to spot longer candidate concepts (the list of submitted concepts is available in table C.1, page 258) from a document, as some of the heuristics we use (such as, 'what is happening more frequently is more important') cannot be reused as easily: long concepts are less likely to be repeated word for word in the document.

[3] A major part of these concepts is already extracted by our noun groups component, c.f. section 5.2.2, page 104.

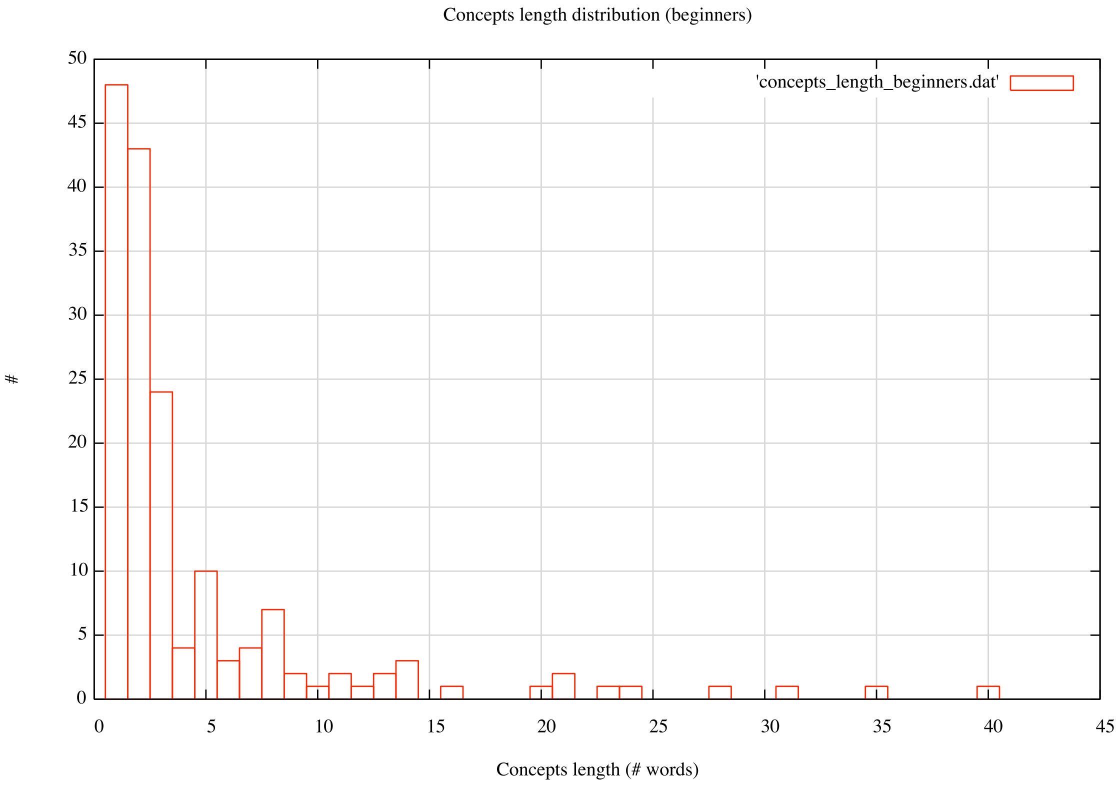


Figure 6.3: Concepts length distribution graph for beginners. 115 concepts out of 174 are composed of three words or less.

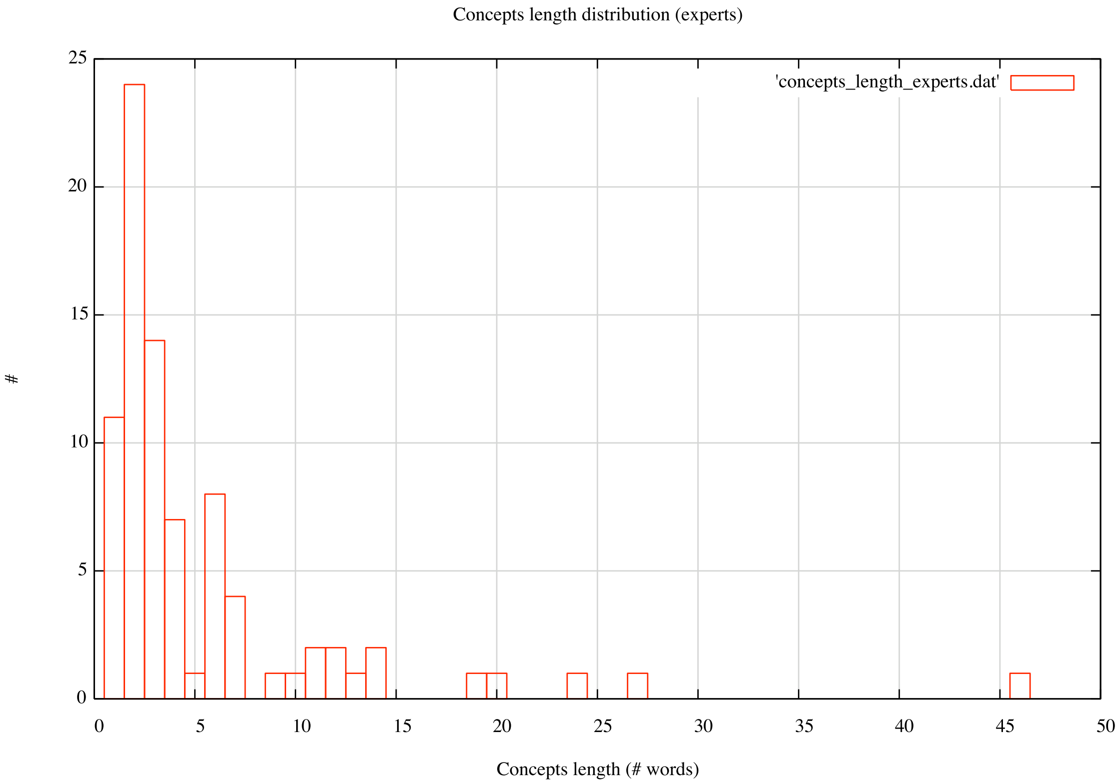


Figure 6.4: Concepts length distribution graph for experts. 49 concepts out of 83 are composed of three words or less.

Explanation A possible explanation for the domination of short concepts, besides the tendency for annotators to write briefly, is that short concepts have a greater potential for reuse, being more generic. Elements such as *Semantic Web* or *hypertext narrative* are easier to reuse than longer concepts (as they convey less information.)

Reused concepts

Table C.3 (c.f. page 259) lists the concepts that have been used several times. Most of these concepts have been used twice, while a handful has been used three times. Duplicated concepts have been either created 'explicitly', by importing a concept previously created in the current document, or implicitly, by typing a text string which happens to be already used to designate a concept. However, the documents chosen by participants are so different that duplicates are mostly due to annotators reusing a concept created beforehand by the tutor during the bootstrapping step. We can also notice that reused concepts are not necessarily composed of short concepts only: some longer concepts have been reused. It seems to indicate that if the effort of creating long concepts has been done already, annotators are ready to reuse them.

Interconnectedness

Reusing concepts is desirable to create models which are more tightly interconnected. A few submitted concepts are near duplicates that could be detected while the annotator is typing them, with suitable replacements proposed to her consideration. Examples include *design rationale* and *design rationale (DR)*, *dimain ontology* (a concept misspelled by the annotator) and *domain ontology*, or *domain ontology* and *domain ontologies*. Additional processing to find the plural or singular form of a concept and propose a form already existing could help.

Another way to merge concepts could be to consider concepts that are more detailed and propose them for consideration. For instance, an annotator inputting a concept *Presence awareness* when *presence awareness of many other people* already exists could be prompted to reuse this existing version to add context, if appropriate. A reversed strategy may also be used to help the annotator break her annotation down into more atomic concepts.

6.3.3 Claims

Table D.1 (c.f. page 267) lists the claims submitted.

Duplicated and challenged claims

Table D.2 (c.f. page 268) lists the 4 claims which have been duplicated (either explicitly, by importing them - from the history window for the current document or from another document - and reusing them; or implicitly, the annotator recreating the same triple without being aware that it was existing before.) Expert a2 challenges a claim made by the tutor and extends one of his own claims to add some evidence. Expert a3 connects two of her claims in a new claim. Beginner a12 connects one of her claims in a new claim to add more information about it.

An explanation to that limited number is that to get a duplicate claim, one has to reuse exactly the same concepts (we have seen how concepts can be very close matches without being exact ones, due to the presence of a plural form or of a typo) and then to decide on the same relation. This makes the probability of getting an exact match much lower.

Interconnectedness

To encourage claim reuse, we can propose claims starting with the current source concept to an annotator's consideration, or claims ending with the current destination concept.

'Strong' claims vs. 'weak' claims

We have raised previously (c.f. section 2.2.2, page 21) the difference between less and more committed claims. Expert a2 notices it too at a point of the experiment:

Listing 6.1: expert a2 @ 24:27 (the code tags introduced in this example - e.g. '`boundaries`' - are presented later.)

```
{boundaries}
''Of course, there are zillions of things one can say''
''I guess, one problem for the annotator is really to decide the
granularity and the boundaries''
''Should I concern myself with making claims like <.. is about ..>
that anybody can say as opposed to ''Scholarly claims that are
more personal'' ?
''And there's also 'when to stop ?'''
...
''But then the automatic support becomes important. Because it'd
be nice if these obvious statements could be found automatically.
{/boundaries}
```

This is an interesting comment for two reasons. The first one is that it highlights one of the problems associated to the annotation, which is: 'how much or how little should I say?' The second interesting point is the awareness by this annotator of a classification into 'trivial' claims that 'anybody can say' and more personal, committed, claims. Annotator a2 makes it clear that she could have skipped some of the claims she has made, in favor of more committed ones requiring more effort and confidence.

The realisation that she could have skipped some of these occurs to her after she has committed them though. As far as the environment is concerned, it may indicate that submitting such 'trivial claims' is a good way for annotators to get started. In the future, more experienced annotators may decide to focus only on the controversial statements made by the author and not on the more generic ones.

To realise if this would happen, we would need to perform another experiment, and assess the impact of factors such as time, ClaimSpotter awareness and experience with the formalism. The only facet we can examine now is based on the relations annotators use in their experiment, which we study bearing in mind that a one-hour slot is indeed too short to derive any too conclusive opinion.

Repartition of the relation types used

Table 6.4 indicates the different relation types used during the experiment. Figures are given for the entire group of annotators, for the subgroups of experts and beginners, and for each annotator individually.

22 relation types (out of the 36 available in the ontology) are used. 7 out of these 22 are used only once or twice. Experts use only 12 relation types to articulate their 59 claims, while beginners use 18 relation types to express their 101 claims.

Discussing the repartition of the relations used seems interesting, but it is difficult because (i) annotators have not annotated identical papers and (ii) even if they had, they would only have annotated elements relevant to their own research interests.

Most frequently used relation groups

Table 6.5 indicates which relation groups have been used most frequently, by the entire group of annotators, by experts and by beginners. 'General' relations are the most frequently used ones.

Relation	A	E	B	a1	a2	a3	a4	a5	a6	a7	a8	a9	a10	a11	a12	a13
Category: general																
is about	22	6	16	1	1	1	3	1		8		1	4	2		
uses/applies/is enabled by	29	11	18	2	1	4	4	1		6		7	1	2	1	
improves on	10	5	5		3	3	2			3			1		1	
impairs	1	1	0				1									
Category: problem																
addresses	12	3	9		2	1	1		1	2		2				4
solves	3	0	3					2		1						
Category: supports																
proves	2	0	2					2								
refutes	0	0	0													
is evidence for	12	1	11		1	1		4			3		2	1	1	
is evidence against	1	0	1								1					
agrees with	0	0	0													
disagrees with	0	0	0													
is consistent with	3	1	2			1								1	1	
is inconsistent with	0	0	0													
Category: taxonomic																
is part of	5	1	4				1	1						3		
is an example of	14	10	4		4	1	5					4				
is a subclass of	10	0	10						3			3	1	1		2
Category: similarity																
is identical to	2	1	1	1					1							
is similar to	4	0	4						2	1				1		
is different to	3	1	2			1										2
is the opposite of	0	0	0													
shares issue with	5	5	0	1		4										
has nothing to do with	0	0	0													
is analogous to	7	7	0			7										
Category: causal																
predicts	0	0	0													
envisages	2	1	1				1									1
causes	2	2	0				2									
is capable of causing	6	1	5				1					1			2	2
is a prerequisite of	0	0	0													
is unlikely to affect	2	0	2										2			
prevents	0	0	0													
Total	160	59	101	5	12	20	22	11	7	21	4	18	11	12	6	11

Table 6.4: Repartition of the different relations used, for all the annotators ('A'), experts ('E'), beginners ('B') and for each annotator ('a1' to 'a13'.)

Relation group	Freq (all)	Freq (exp)	Freq (beg)
General	38.75% (62)	39%	38.6%
Problem	9.38% (15)	5.1%	11.9%
Supports	11.25% (18)	3.4%	15.8%
Taxonomic	18.13% (29)	18.6%	17.8%
Similarity	13.12% (21)	23.7%	6.9%
Causality	7.5% (12)	5%	7.9%

Table 6.5: Relation groups. The most frequently used group of relation is ‘general’, followed by ‘similarity’ for experts and ‘taxonomic’ for beginners.

Pos	Relation type	Frequency
1	uses/applies/is enabled by	18.64% (11 uses out of 59)
2	is an example of	16.95% (10)
3	is analogous to	11.86% (7)
4	is about	10.17% (6)
5	shares issues with	8.47% (5)
	improves on	8.47% (5)
7	addresses	5.08% (3)
8	causes	3.39% (2)
9	is identical to	1.69% (1)
	impairs	1.69% (1)
	envisages	1.69% (1)
	is capable of causing	1.69% (1)
	part of	1.69% (1)
	is evidence for	1.69% (1)
	is different to	1.69% (1)
	is consistent with	1.69% (1)

Table 6.6: Relation types used by the 4 experts, in decreasing order of frequency.

Pos	Relation type	Frequency
1	uses/applies/is enabled by	17.82% (18 uses out of 101)
2	is about	15.84% (16)
3	is evidence for	10.89% (11)
4	subclass of	9.9% (10)
5	addresses	8.9% (9)
6	improves on	4.95% (5)
	is capable of causing	4.95% (5)
8	is similar to	3.96% (4)
	part of	3.96% (4)
	example of	3.96% (4)
11	solves	2.97% (3)
12	proves	1.98% (2)
	is unlikely to affect	1.98% (2)
	is consistent with	1.98% (2)
	is different to	1.98% (2)
16	is evidence against	0.99% (1)
	is identical to	0.99% (1)
	envisages	0.99% (1)

Table 6.7: Relation types used by the 9 beginners, in decreasing order of frequency.

Most frequently used relations

Tables 6.6 and 6.7 list the most frequently used relation types for experts and beginners, respectively. These two tables show again how problematic it is to talk about the relations used, for it depends so much on the original document and on the annotator. All the uses of ***is analogous to***, for instance, are down to a single annotator, a3 (c.f. table 6.4), which prevents any generality to be drawn.

However, relation types such as ***uses/applies/is enabled by*** and ***is about*** are among the most used relations. For the former relation, it may be that most annotated documents are computer science papers that are more likely to reuse, or apply, an existing technology. The use of ***is about*** may be explained by the fact that it is one of the less committing relations, as 'anything' can often be said *to be about*, even remotely, 'something else.'

Most consistently used relation types

A more interesting aspect is to find out the relation types most consistently used by annotators. Table 6.8 organises relation types by the number of annotators having made use of them, at least once.

Uses/applies/is enabled by and ***is about*** are also the most consistently used relation types. Only three annotators do not create an ***uses/applies/is enabled by*** claim, and only four do not create an ***is about*** claim. Once again, the nature of the paper is very likely to play a role here: the papers annotated by a6, a8 and a13 (who do not use either of these two relations) are significantly different (in their nature) from the other papers (appendix E, page 269, contains the titles and abstracts of the documents used in the experiment.)

The relatively small number of annotators having made use of ***addresses*** (only 6 annotators out of 13) may appear strange. The CARS model that we have presented in the literature review (c.f. section 4.6.5, page 83) has shown how important stating the problem being tackled is important for the author. This may indicate that annotators do not think about identifying the problem that the document addresses and/or the approach it proposes. This in turn may indicate a need to support the construction of more robust models, by ensuring, as we have seen in scaffolding approaches in the literature review, that the claim spaces annotators create contain such concepts and claims. We come back to this point in the next chapter.

The 'is about' relational type

A more thorough evaluation of the use of ***is about*** is now given. The main motivation for this study comes from the fact that it is one of the most consistently used relations. It also comes from its generic nature, and from the relatively little commitment it imposes. ***Is about*** is more likely to play the role of a holdall relation, a relation that can be used when nothing else can.

This does not mean that ***is about*** claims are 'worthless.' An ***is about*** claim may contain a valuable and 'surprising' claim, if it connects two unexpected concepts. However, the link created between these two concepts is more likely to indicate a commitment that is not as high as some other links.

Experts vs. beginners The first comparison we can make is whether the experts or the beginners make more use of this relation. Experts submit 6 claims out of 59 with the ***is about*** link (10.2%), while beginners submit 16 claims out of 101 with it (15.8%.) Experts submit proportionally fewer ***is about*** claims. This can come from their higher awareness of the other relations available, but also from their training: they may know what the ontology captures and may therefore focus more on these additional, and richer, ways to articulate concepts.

Beginners, on the other hand, are more likely to use ***is about*** as a holdall relation, especially in these occasions when they start from two concepts without knowing if the relation they want to use exists (a phenomenon that we have recorded with the 'starting from the concepts' in our code taxonomy; more about this in the next section.) In these particular cases, ***is about*** becomes a life buoy. Beginner annotator a7 uses it in 8 of her 21 submitted claims, including:

- {*espotter*, ***is about***, *named entity recognition*}
- {*espotter*, ***is about***, *domain adaptation*}
- {*espotter*, ***is about***, *user adaptation*}
- {*espotter*, ***is about***, *probability estimation using Google search*}

While we are not discussing the intrinsic quality of these claims, it can be guessed that some may have been made stronger with a different relation type. For instance, the fourth claim could have used ***uses/applies/is enabled by***.

The language used by the authors of the document also plays a role. An expression such as ***is about*** is likely to be used more frequently than ***improves on***, resulting in more instances being highlighted throughout the text with the ‘matched relations’ filter. The resulting graphical emphasis may thus influence annotators too.

Talkative vs. less active annotators We consider now a new comparison, by separating the annotators who submit less than 10 claims from the ones who submit 10 claims or more.

Table 6.9 gives results that are more interesting. It seems to indicate that the number of ***is about*** claims grows much more than the number of claims submitted. When annotators submit a high number of relations for a given document, the proportion of ***is about*** claims is more important. For instance, annotator a7 binds 8 out of her 21 claims with this relation.

On the other hand, annotators who write fewer claims make almost no use at all of the ***is about*** relation. It may be because they focus directly on stronger claims and refuse to waste time on less committed ones. Annotator a8 does not use ***is about*** for her 4 claims, nor does a12; expert a1 uses it only once out of 5 claims.

There is not necessarily an overlap between less talkative annotators and experts though. Out of the 4 less talkative annotators, 3 are beginners and 1 is an expert. Out of the more talkative annotators, 6 are beginners and 3 are experts.

The time factor The final dimension we consider is time. We consider the ***is about*** claims submitted in the first half of the annotation process and the ones submitted in the second half (c.f. table 6.10.)

This last analysis seems to show that annotators submit more ***is about*** claims at the beginning of their sense-making process and less in the second half [4]. Here, the ‘less committing’ aspect of ***is about*** may be playing a role: it may help annotators make the first step of their annotation process (such as, for instance, formalise their ‘first’ claim): by the time they have added half their annotation, 8 annotators out of 13 have submitted at least one ***is about*** claim.

Once they reach this point of the experiment, they may be more knowledgeable with the process and the relations available and the need to fall back on ***is about*** may be less crucial. ***Is about*** may be seen as a way to incrementally formalise one’s interpretation: the system could prompt annotators at a later stage to replace less risky claims using ***is about*** relation with stronger relation types.

[4] There are fewer claims in the second half because the claims falling in the middle - for annotators having submitted an odd number of claims - are counted in the first half

Conclusion

This analysis contributes to making this evaluation study not only a ClaimSpotter evaluation, but also a ScholOnto evaluation, as we have seen how experts and beginners have made use of this formalism. Studying concepts length has helped us spot for instance the importance of shorter concepts, while studying the relation types used has shown that only a subset of the relations available has been used, and that only a few have been used both frequently and consistently by three quarters of the annotators (***uses/applies/is enabled by*** and ***is about***.) It is true that 9 out of 13 participants are new to the formalism. It is also true that not every relation type can be used for a single paper. It may also be that over time, with more and more sessions, annotators tend to 'absorb' and reuse more and more relation types, and in particular, that they replace their ***is about*** claims with more committing ones.

This being said, the importance taken by ***is about***, especially for beginners, is interesting. It seems to indicate that in a formalism, whatever this formalism is, some simple constructs are needed to help users make their first step. In the context of ScholOnto, the ***is about*** relation may very well play this role. It is also, incidentally, the first relation displayed in the pull-down menu of the interface.

It is indeed interesting to note the repartition of the relations in the top of the graph (c.f. table 6.4.) As the presentation of these relations follows the order in which relations are displayed in the input form panel of the ClaimSpotter interface, this repartition may be explained by the fact that frequently used relations are the first ones in the input form.

An interesting experience would be to reshuffle the order in which relations are displayed and to find out whether a similar phenomenon is observed.

6.3.4 Documents

Annotators have been asked to annotate a document they are familiar with, in order to limit a cognitive overload that (most of them being new to the formalism and to the environment) is already quite high. As the Knowledge Media Institute is a multi-disciplinary research laboratory, the documents considered for this experiment share this characteristic. There are consequently only a limited number of cross-citations between them. Figure 6.5 (left) shows how interconnected, via their citations only, they are.

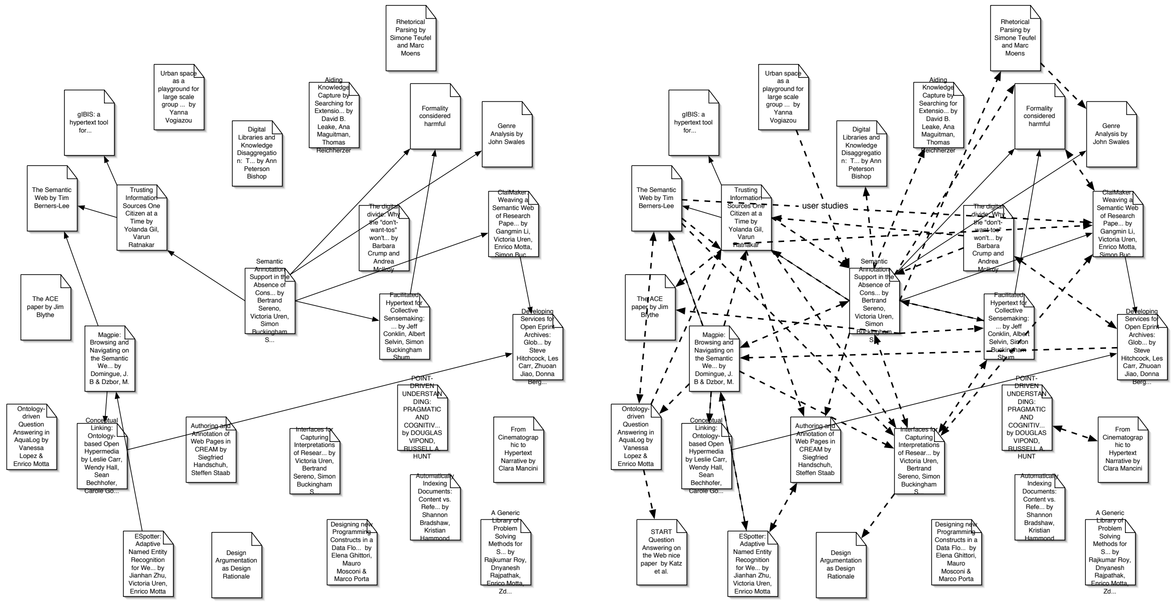


Figure 6.5: Document corpus with citations only (left) and with 'semantic relations' (right.)

Semantic connections

We notice that the network becomes much more interconnected, as interpretations are added. As we have seen in chapter 2, documents sharing a concept or involved in a claim become 'semantically connected.' Figure 6.5 (right) shows a large increase in the number of links between documents, resulting from the interpretations (concepts and claims) added over them.

Example

Creating connections between documents is an interesting side-effect. For example, the documents '*Point-driven understanding: pragmatic and cognitive dimensions of literary reading*' and '*From Cinematographic to Hypertext Narrative*' have become semantically related via their share of a concept *coherence* (c.f. figure 6.6.) *Coherence* can be found in the list of duplicated concepts (c.f. table C.3, page 259.)

In this case, an annotator has created this concept for the former document, and another annotator has created it for the latter. Of course, it might well be that not all these connections are relevant, as concepts may have different meanings in different contexts. It is nevertheless a potentially interesting way to build additional connections (or 'creative leaps' to reuse a concept map expression) between documents.

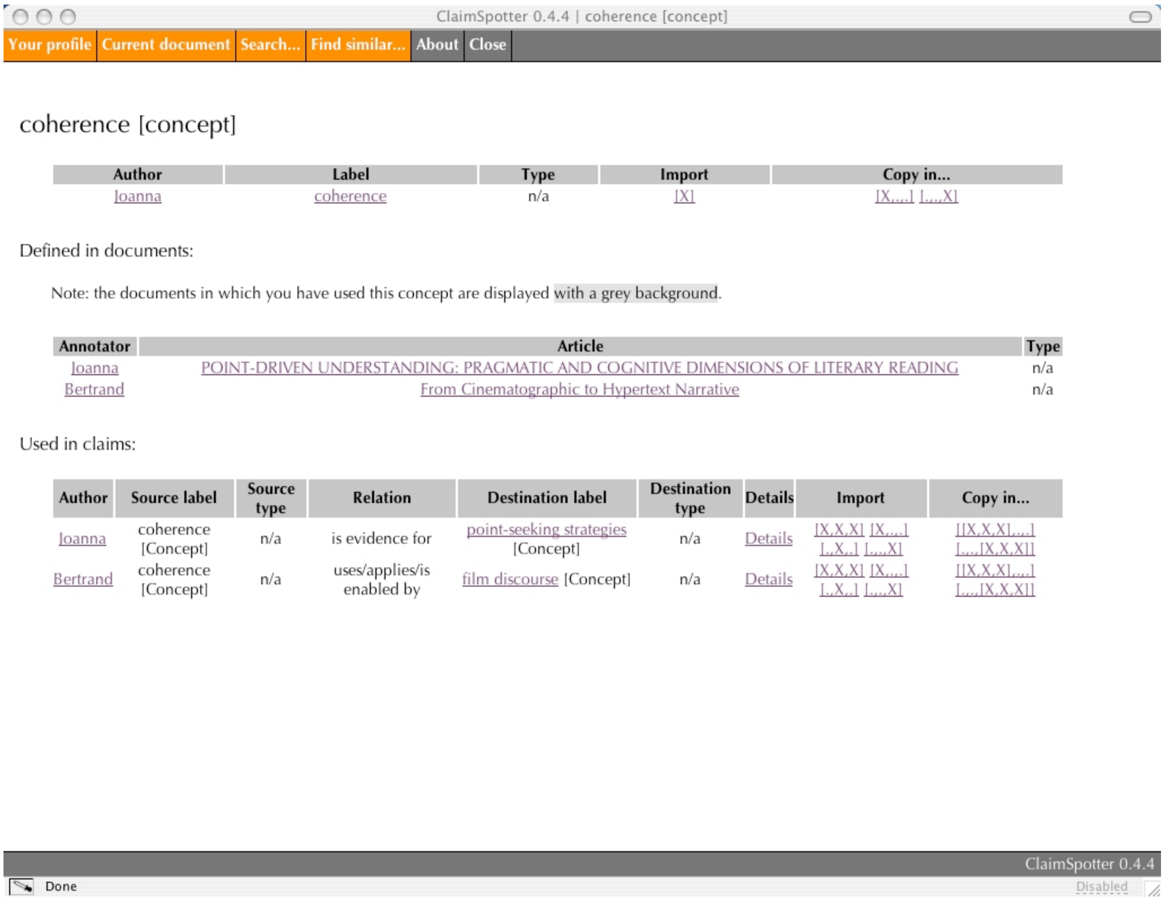


Figure 6.6: The documents '*Point-driven understanding: pragmatic and cognitive dimensions of literary reading*' and '*From Cinematographic to Hypertext narrative*' have become (semantically) related via their annotation: they share a concept *coherence*.

Added documents

Only one new document is added (c.f. bottom-left corner) by beginner annotator a11. Although the focus of this experiment is put on single-document annotation , annotators are free to model any additional document they need to build their literature model. The fact that documents have been seldom added may result, once more, from the cognitive load annotators have had to deal with. In subsequent experiments, placing annotators in a network of documents and asking them to model their connections, as well as their contents, would be interesting. More time and training would have made more space for multiple document annotations, which would have also given additional insight in the sense-making task.

We move to our qualitative analysis, in which we consider and explain several critical incidents to derive a theory of ClaimSpotter usability, that we illustrate with extracts taken from the experiment.

Pos	Relation type	# annotators
1	uses/applies/is enabled by	10 (4, 6)
2	is about	9 (4, 5)
3	addresses	7 (3, 4)
4	is evidence for	7 (2, 5)
5	improves on	6 (3, 3)
6	subclass of	5 (0, 5)
7	example of	4 (3, 1)
8	is capable of causing	4 (1, 3)
9	is consistent with	3 (1, 2)
10	is similar to	3 (0, 3)

Table 6.8: The 10 relations most consistently used by the annotators. Each figure is fractioned in the number of uses by experts and beginners.

Group	# Claims	# 'is about' relations used
4 less talkative annotators	22 (mean: 5.5)	1 out of 22 (4.55%)
9 more talkative annotators	138 (mean: 15.33)	21 out of 138 (15.22%)

Table 6.9: Comparison of the use of 'is about' between talkative and less active annotators

Time	# Claims	# 'is about' relations used
First half	82	16 out of 82 (19.5%)
Second half	78	6 out of 78 (7.9%)

Table 6.10: Comparison of the use of 'is about' in the first 82 claims submitted and in the last 78 ones.

6.4 Interaction analysis

In a qualitative analysis, meanings and intentions are essential to discover aspects which cannot be captured with quantitative analyses, but which can be recognised by social interactions [Orlikowski and Baroudi, 1991]. As the focus is on the interpretative aspect, objective stances cannot be achieved and the study is biased by the investigator's beliefs and assumptions [Dix et al., 2004, page 358]. In spite of this bias, this is the level of knowledge we are interested in: interaction, dialogue between annotators and ClaimSpotter.

6.4.1 Methodology

We analyse our data using a shallow grounded theory model. It is a methodology to think about data (i.e. to create concepts) and organise it (i.e. to draw relations between these concepts) [Glaser and Strauss, 1967]. It is inductive in the sense that the outcome of this methodology (the theory, i.e. a set of plausible relationships holding among multiple concepts), emerges from the data being analysed.

The data collection phase is also influenced by the outcome of an analysis phase (on the previously collected data) that is run in parallel. Concepts and categories emerging are therefore constantly compared against each other (through separation of a code (representing a concept) into multiple sub-codes, or the consolidation of different codes into broader ones) during the analysis phase, until a stable state (also called the point of theoretical saturation) is reached.

6.4.2 Coding

We start our analysis with a flat transcription of the annotators' actions and utterances. Fragments of these transcripts are progressively labelled with codes indicating the phenomenon (i.e. the concept) they refer to. During the transcription process, codes are typically renamed or merged, to incorporate different kind of fragments, or refined into smaller codes if the analyst deems it is necessary. The emerging coding scheme provides a language with which we can describe the data at various levels of detail, from broad themes down to their constituent codes.

We code certain aspects of the interaction, like the activation of a particular feature of the interface. Other codes are created for 'critical incidents' relevant to the questions we have identified in the breakdown of our second research question.

Code	# codes
actions with the interface	41
questions about the interface	1
problem, or any aspect relating to the formalism	63
changing formulation of concept or relation	9
satisfaction with the system	11
consulting, reusing previous stuff	22
reacting to previous stuff	9
interface behaviour	9
making use of the data available	29
misunderstanding of the filters	4
making use of the interface	11
problem of the interface	18
explains strategy for the task	4
auhtors' reactions	2
i...	50

Table 6.11: The first version of the taxonomy. The 'I...' category regroups every action performed by annotators and every statement they uttered. As we were making sense of this information, we needed to have an as broad as possible vision. Distinct codes were created for every single phenomenon.

6.4.3 Towards a theory of usability for ClaimSpotter

The code taxonomy, reflecting the aspects that we deem interesting in our assessment of the usability for ClaimSpotter, evolves over time. The following paragraphs reflect this evolution, from the first version to the latest, 'stable', one.

Version #1

The first version of the taxonomy being our first attempt at the analysis, its codes are listed in a nearly flat structure, with little organisation between them: only a few broad categories are used to regroup codes. It contains 307 codes, organised in 15 categories. Table 6.11 gives an overview of the first version of the taxonomy.

Some codes are designed to match a very explicit situation. For instance, '*importing an existing concept from the text of the document*', and '*importing an existing concept from the history window*' are recorded using two different codes in the '*consulting, reusing previous stuff*' category. That huge number of codes has made the taxonomy unmanageable, prompting us to restart from scratch.

Version #2

For the second version of the taxonomy, we organise codes into more generic categories covering more phenomena. As we structure codes, five broad themes start to emerge: formalisa-

Theme	Category	# codes
formalisation	awareness	1
	appropriate/good things	2
	not perfect	2
	difficulties	4
	suggestions	4
	discussions about concepts	4
	misc	1
interface/filters	annotating	11
	modifying document view	6
	awareness	1
	appropriate/good things	1
	suggestions	2
strategy	use of filters	9
	approaches	2
	misc	7
usability	using on-screen help to sort a potential problem	-
	checking if data has been properly inserted	-
	good things/things that work	2
	mistakes	5
	problems/issues	7
	suggestions	2
misc		2

Table 6.12: Themes and categories are used to organise our different codes (we only indicate the number of sub-codes in each category) in our second version of the code taxonomy.

tion, interface/filters, strategy and usability (thus starting to reflect our understanding of the research question.) Each of these themes has several sub-themes, themselves composed of codes. This second version goes through several minor modifications over time. Table 6.12 lists the themes and categories that we create for the second version of the taxonomy.

Version #3

The third - and current - version (c.f. table 6.13) is based on the second one, keeping three main themes: formalisation, strategy and interaction design (the additional theme '*misc*' is kept to store two codes that we could not put elsewhere, as in the second version.) This version covers our current understanding and accounts for the deemed (interesting) events and phenomena occurring in the movies transcripts. It is presented in the following section.

6.4.4 A theory of usability

This final taxonomy presents our current analysis of the usability of ClaimSpotter (c.f. table 6.13.) It is read from left to right, with the far-left column listing the three main themes, and each subsequent column listing the branches or the leaves of a theme, i.e. its sub-categories.

Three main themes are identified (presented in the column THEME; the fourth theme, 'misc.' contains the different aspects we could not classify in a better way.)

If a code has no neighbour on its right, it means it is a leaf of the taxonomy. For instance, *incremental formalisation* is a code that has been filed under the *strategy* theme. It has been assigned to fragments of the transcripts. It is also a category of codes: in some occasions, it has been found interesting to go into finer detail and characterise more specifically this incremental formalisation aspect, using for instance the code *writing a note to add explanation*. This tree-like organisation means that every instance of *writing a note to add explanation* is an instance of *incremental formalisation*, but it also means that the opposite is not true.

6.4.5 Example of an annotated file

To annotate relevant segments of the transcripts with codes, we have used the open source program TAMS [5]. TAMS is a command-line program, running under Unix and enabling the selection of a chunk of text (from a transcript) and its tagging with a code. We have used the Macintosh OS X graphical front-end to this program to encode our files [6]. Figure 6.7 gives an example of an annotation session with TAMS.

Listing 6.2 is an extract of a movie transcript. Code tags are similar in nature to XHTML tags: data between the tags is described by the tag. Time tags represent the time elapsed since the beginning of the experiment.

[5] It is available at: `http://tamsys.sourceforge.net/osxtams/`

[6] This version is also developed under an open source license, by Matthew Weinstein.

Theme	Category (code)	Sub-category (code)	Code
formalisation	creating a concept	choosing concept type	appropriate concept type
			not perfect concept type but
			problem with or lack of a concept type
			cannot find a concept type
		removes concept type	
		deletes concept	
	creating a claim	makes no sense	suggestion not good enough
		changing a concept formulation because of a relation	
		changing the order of left and right sides	
		changing the relational type	
		choosing a relation	appropriate relation
			not perfect relation but
			problem with or lack of a relation
			cannot find a relational type
		deletes a relation	
	discussion about formalism	I want this concept type	
		I want this relation	
		boundaries	
		difficulties with formalism general	
strategy	keeping things simple		
	reducing amount of information on screen	looking for ideas	
		focusing on a particular area	
		hiding section	
	typing or selecting a concept		
	starting from what i want to say		
	starting from relation		
	incremental formalization	submitting incrementally	
		writing a note to add explanations	
		writing a note as a reminder	
		importing own stuff	
	strategy misc		
interaction design	Intuitiveness	mistake during interaction	
		misunderstanding of the role played by filters	
		problems issues with interface	
	Feedback	on-screen text is helpful to create concepts	
		on-screen text is helpful to create relations	
		using tooltips to decide what to do	
		checking submitted stuff is there	
	writing made ideas clearer		
	consistency	consistency in the interface	
	support	support to create relations	
	suggestions		
	interface misc		
misc.			

Table 6.13: The latest version of our code taxonomy contains 59 codes.

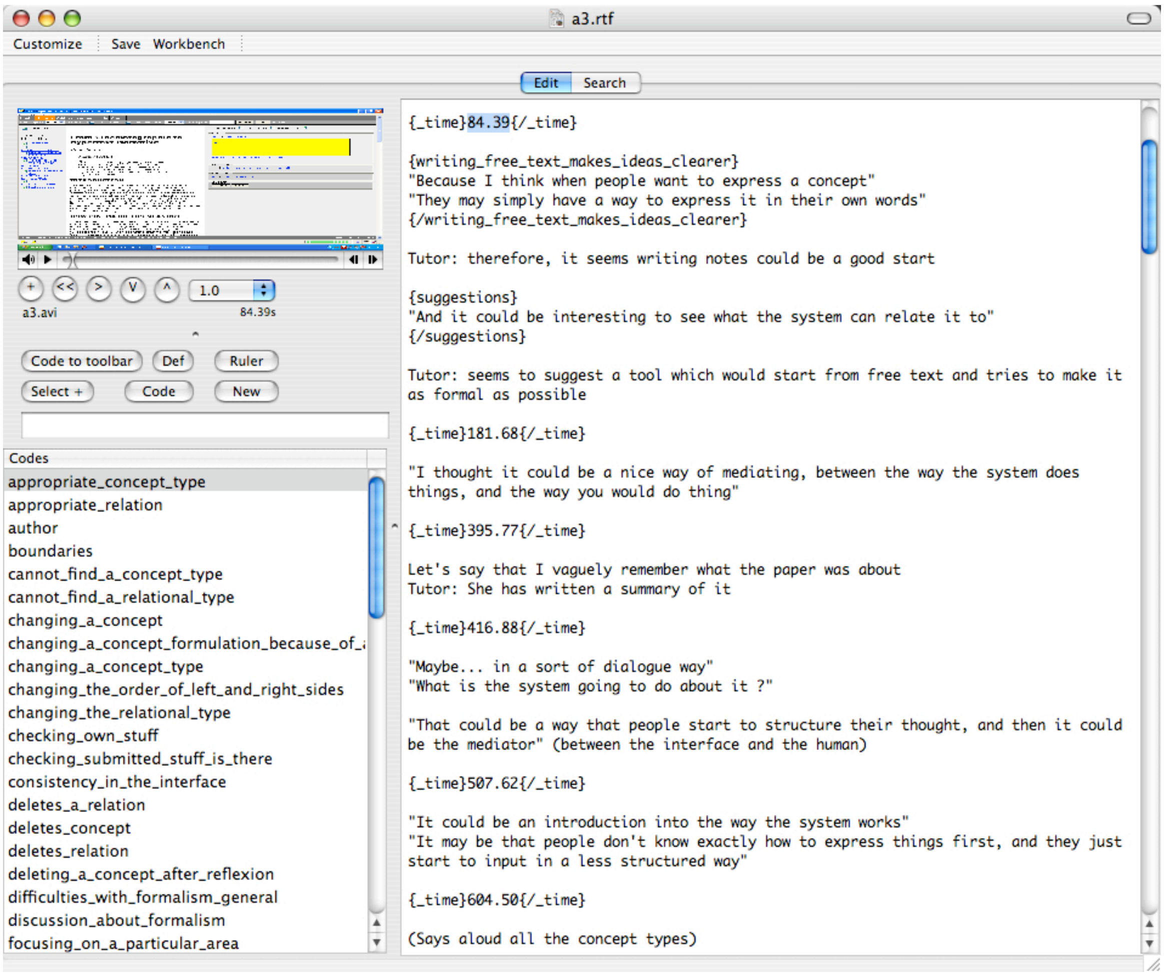


Figure 6.7: A coding session in TAMS: text chunks (main window, right side) are selected and assigned a code (selected from the bottom-left window, or created from scratch.)

Listing 6.2: Extract from a movie transcript. Movies are annotated with time slots and significant statements are assigned a code from the taxonomy (using opening `{a_code}` and closing `{/a_code}` tags.) Statements surrounded by “quotes” are uttered by the annotator; Statements prefixed by ‘Tutor:’ reflect the tutor’s opinions and interpretations.

```
{writing_free_text_makes_ideas_clearer}
"If I say what I... kind of remember about the paper"
"In my own words"
"Vaguely.. this was about..."
{/writing_free_text_makes_ideas_clearer}

{suggestions}
"I would like to see what it (the system, with the note) makes of
it"
{/suggestions}

{_time}84.39{/_time}
{writing_free_text_makes_ideas_clearer}
"Because I think when people want to express a concept"
"They may simply have a way to express it in their own words"
{/writing_free_text_makes_ideas_clearer}

Tutor: therefore, it seems writing notes could be a good start

{suggestions}
"And it could be interesting to see what the system can relate it
to"
{/suggestions}

Tutor: seems to suggest a tool which would start from free text
and tries to make it as formal as possible

{_time}181.68{/_time}
"I thought it could be a nice way of mediating, between the way
the system does things, and the way you would do thing"
```

6.4.6 ClaimSpotter usability: formalisation

We begin our interaction analysis of the usability of ClaimSpotter with a study of the incidents we have recorded and classified under the ‘formalisation’ theme (c.f. table 6.13.)

Creating a concept

We have presented in the previous section a few characteristics of the concepts submitted. We have seen in particular that most of them are short (three words or less.) The freedom left to annotators in their formulation is welcomed and removes some of the strain implied with ‘real’ concepts in a domain ontology, in which aspects like clarity and reusability are enforced.

Concept types We have witnessed several incidents related to the addition of a concept type to a concept (the decision not to add a concept type is motivated by its optional aspect, the difficulty to find the right type for the concept, or the non-availability of a particular type.) In 34 incidents, there is an appropriate concept type matching what the annotator has in mind: “*It’s actually... It’s a phenomenon*.”

Other situations are more problematic:

- in one case, there is only a type by default that can be chosen, a type that is satisfying enough (i.e. that the annotator will use), but not representing exactly what she has in mind (“*I’ll call it ’problem’, but it’s not a problem, it’s a shortcoming*”);

- in two cases, there is a *problem with a concept type*, resulting in a discussion about what type(s) should be added;

- in two more cases, a concept type *cannot not be found*;

- in one case, a concept type *cannot be found* because there are several types fitting the concept (“*The interesting thing is that this specific example* (concept) could fall in different categories.”) When that happens, and after having explained that two instances of the same concept can coexist with multiple types, annotators either add them twice or stick to one concept type only (”*Some people can say different things about this concept, but I don’t want to add it more than once, because I am building a taxonomy*.”) In these cases, the solution is to go back to the default unassigned concept type.

The following extract illustrates the difficulties met to assign a type to a concept:

Listing 6.3: Beginner a10

```
''It's not a problem, and it's not a solution, and it's not a me-
thodology...''
''It's a technology''
{i_want_this_concept_type}
''I think... what you'd look for is stg that says... 'research
field', or...''
{/i_want_this_concept_type}
'Add' (a concept)
{looking_for_ideas}Copies and pastes the text{/looking_for_ideas}
{cannot_find_a_concept_type}
No type selected
{/cannot_find_a_concept_type}
```

Although this situation is faced by a beginner, it is not significant here: a similar situation can be faced by any annotator, if the type she wants to use cannot be found.

Deleting a concept In some cases, concepts that have been created earlier on are removed because a newly created concept expresses the annotator's intention in a better way. Annotators need time to craft their interpretation and the interface should not press them.

Creating a claim

The average number of claims submitted seems to indicate that articulating concepts into claims is not too difficult (at least in the context of annotating a single document annotators are familiar with beforehand.) It is however perceived as a more difficult task, as annotator a8 mentions: "*I can pick up some concepts from the paper, but when it comes to relationships…*"

Difficulties arise from relations not matching the domain of the document. This remark is stated by this particular annotator a8, who is the least 'productive' (although less at ease with the formalism would be a better way to refer to it.) In this case, the formalism and the relations are not appropriate to let her express her ideas and connect them with the right terminology.

An *appropriate relation* is found in 115 incidents. However, as for the '*choosing a concept type*' code, choosing the 'right' relational type creates difficulties:

- 8 incidents in which a relation is *appropriate enough* are found, (meaning that the annotator keeps and submits the claim), although it does not express completely what an annotator has in mind ("*I can say 'is similar to', since there is nothing else better than that*");

- Once, an annotator is left wondering which relation to choose because "*several (relations) seem to be appropriate, in different categories*." The tutor suggests to create as many claims as appropriate relations;

- in 10 incidents, there is a *problem with (or a lack of) a relation*;

- in 6 incidents, the problem is even more acute ("*This relation ... is not there. So in that case, what do we do then ?*") and results in the deletion of the claim being created.

Expressing these figures as percentages of the total number of claims submitted is difficult. Incidents such as finding a relation that is *not perfectly appropriate* but good enough can only be recorded if the annotator expresses her opinion clearly. If she (internally) decides that this relation is not good enough and if she does not voice her lack of satisfaction with it, we cannot record this incident accurately. The total figure of incidents recorded (115 + 8 + 1 + 10 + 6) differs from the total number of claims submitted because we have missed instances for similar reasons. For the record, the 115 '*appropriate relation*' incidents (out of the 140 reported) amount to 82.1%.

Levels of commitment Differences between the level of commitment associated to different relations are also noted. Beginner annotator a9 states how "*'is about' is weak*" and decides to spend some more time choosing a more accurate relation to express her idea:

Listing 6.4: Beginner a9 at 50:36

```
Creates a claim with [X,.,.] and [.,.,X]
{appropriate_relation}
Chooses type 'is about'
{/appropriate_relation}
{discussion_about_formalism}
''Is about is weak... ''
{/discussion_about_formalism}
{changing_the_relational_type}
Reads the relations
{/changing_the_relational_type}
{appropriate_relation}
Chooses 'subclass of' in the end
```

```
{/appropriate_relation}
```

Revising This last transcript extract also gives us an indication that multiple attempts (as we have seen when concepts are removed in favour of better formulated ones) are needed to get a claim right.

A claim may also be revised by reformulating a concept to make it suit a given relation. We witness 11 cases in which an annotator has to *reformulate a concept because of a relation*:

Listing 6.5: Expert a2 at 18:57

```
{strategy_misc}''Now, I'd like to say something about limitations
and open issues''{/strategy_misc}
{starting_from_what_i_want_to_say}
Adds a claim.
{/starting_from_what_i_want_to_say}
Put a concept on the left hand side
{cannot_find_a_relational_type}
Browses the relational types but realises there's no relational
type to say that
{/cannot_find_a_relational_type}
{strategy_misc}
''So you have to say it in a different way.''
''I guess you have to create a concept 'lack of...'''
{/strategy_misc}
{changing_a_concept_formulation_because_of_a_relation}
Creates a concept.
{/changing_a_concept_formulation_because_of_a_relation}
```

In this case, annotator a2 wants to express a limitation with the approach presented in the document she is annotating. However, since no appropriate relation can be found, she decides to revise her strategy and modifies a concept to make it suit one of the relations available. The claim that she ends up submitting is {*inability to use existing semantic annotations*, ***is an example of***, *problem with magpie*}.

6 of these incidents come from experts, while the remaining 5 come from beginners. Although we can expect experts to encounter this problem less often as they are more aware of what is and is not feasible with the formalism, it is surprising to find that the opposite happens.

Claims can also be revised by *changing the order of its left and right concepts* with the 'flip' button [7]. It is used in 11 occasions [8], 4 times by expert a4 and 7 times by beginner annotators:

Listing 6.6: 'Beginner a10 at 8:05'

```
{support_to_create_relations}
(The 'flip' button does what is expected)
''That's exactly it. That's cool.''
''It's a very useful thing because that's a thing a human would do.
You know, not getting my concepts the right way around, because I
am thinking and just throwing concepts in.''
{/support_to_create_relations}
```

Deleting a claim When no satisfying relation is found to connect two concepts, even by flipping or reformulating them, the triple is deleted. It happens on 10 occasions (6.25% of the 160 claims submitted), including 6 times for expert annotator a4, and 4 times for beginner annotators a5 and a9.

Discussion about the formalisms Additional considerations about the formalism, expressed by annotators, are recorded under the *discussion about the formalism* code.

Boundaries Annotator a9 perceives the difference between concepts 'imported' in a document by its author (via citations) and concepts the document itself defines (such as a new methodology or an application.) It is not clear to her whether a particular concept that is not *defined* in the document but only *mentioned* should be included or not ("*Where should the concepts be in the document itself (. . .) So it doesn't matter if this concept is only borrowed (instead of defined) in this document, right?*") On the other hand, annotator a12 states that she would input a particular concept because "*this is a concept I want to put anyway, whether it's external (to the document) or not, because I'm using it.*"

'Silly' and 'good' concepts Concepts quality is an aspect brought forward by annotator a12. She mentions that "*(that particular concept)'s a silly concept but I'll make it anyway*" about a concept that she does end up submitting, because it is of interest to her. She then

[7] This feature did not exist for the first two annotators who participated to the experiment (a1 and a7.) We did realise how important it was, as annotators being faced with this problem had to delete a claim and restart it from scratch: we added this functionality immediately for the remaining annotators.

[8] for the remaining 11 annotators.

raises the notion of quality: "*I'm not sure if that (this concept)'s gonna be good. Maybe some of these concepts are less useful than the others.*"

Discussion is pushed further with her and we ask her explicitly "*what would be a good concept?*" Her answer is most interesting: "*a good concept will be something that is consistent, something that would appear again and again in the document. 'CitiTag' is a good concept for instance. Compared to something I would use only once.*" Her notion of 'good' concepts seems to rely on the frequency and potential utility of a concept.

Important claims A need to differentiate some particular claims and to make them stand out (from an annotator's other claims) is expressed by expert a2: "*I want to say I feel strongly about this.*" This reflects again a separation between maybe more mundane claims (possibly giving background information) and claims expressing clearly the opinion of an annotator.

Add my relations Annotator a8 (who has difficulties finding the right relations to connect her concepts) expresses how "*it would actually be quite nice if you could make your own relations.*" The possibility to redefine the relations to make them suit a particular research field is offered in ScholOnto. We should try this possibility in a next experiment.

Finally, concept types can also function as scaffolds, helping annotator a12 create her concepts:

Listing 6.7: Beginner a12 at 11:47

```
{discussion_about_formalism}
Tutor: You said that the definition of concepts was helping you.
Because it can give you an indication about what can be a con-
cept? What to put?
''Yes. Otherwise, anything can be a concept''
{/discussion_about_formalism}
```

Conclusion about the formalisation analysis

We have given in this section a comprehensive view of the phenomena related to the formalisation of concepts and claims to answer our two initial questions: what happens when an annotator decides to create concepts and claims?

We have tracked several incidents to understand how they are related to each other (such as the transition from the *lack of a particular relation* to the *reformulation of a concept to make it 'fit' in a particular claim.*) When a reformulation (or a flipping of the left and right parts of a claim) is not possible, we have to accept that this part of an interpretation is going to be lost. Of course, the discourse ontology expresses only the original vision of the project and cannot be expected to provide each and every relation. Not being able to express what one wants to express ("*But what if I cannot find any relationships between two concepts?*"), therefore forcing one to delete a claim, is an aspect we should try to avoid though.

How much should I say? The amount of concepts and claims one would submit is another aspect we have understood better. In this experiment, annotators are limited to a one hour-slot. The annotation is also focussed on the one document that had to be annotated, making the modelling of its connections to a network of papers less obvious (but for annotator a11, who adds a document to the network.) These aspects have consequences on the nature and the amount of formalisation that can be made.

The experiment is however appropriate enough to realise the effort (how much does one want to put in annotating a document?) and commitment (in the concepts and claims one submits) needed. A public profile for annotators (containing concepts and claims they will have submitted for a document) may shape this process, making annotators focus on their more important claims, the one they want to see associated to their name. As noted earlier, an annotation methodology (or simply 'hints') or guidelines could help annotators focus on stronger claims in a public space.

6.4.7 ClaimSpotter usability: strategy

We present in this section our analysis of the strategy theme. Although we have expected as many strategies as annotators (since any approach can be different, based on one's knowledge of the formalism and of the application, expectations and needs), we have witnessed several recurrent patterns emerging from the use of ClaimSpotter.

Amount of support used

We note a first difference in the amount of support annotators use from the environment and its ability to extract and suggest elements. Beginner annotator a7 makes little use of the suggestions and spends most of the experiment inputting concepts and relations, while the other participants do actually use the suggestions.

97 instances (spread over 9 annotators, 2 of them being experts and representing 7 instances out of these 97, and the remaining 7 annotators being beginners, representing the 90 remaining occurrences of this incident) of the code *typing or selecting a concept* are

recorded, indicating that annotators type the content of a concept directly, or via the selection of text which is not spotted and highlighted. These 97 incidents represent 45.5% of the total [9] [10].

Expert a1, at some point, prefers to deactivate the suggestions because, in her words, "*I don't want to be too distracted by having too many things going on. At the moment, it seems to be quite complicated. I'd rather keep it simple.*" At another moment of her annotation though, she makes use of the suggestions "*to see if there's anything inspirational here.*"

On the other hand, we witness 116 instances (54.5%) of the *looking for ideas* incident (spread over 12 annotators: 4 experts with 54 instances and 8 beginners with 62 instances.) It seems to suggest that experts make the most use of the support available, which can seem surprising as one would expect them to need less support to model the contributions of a document. A more correct explanation can be that it simply depends on how much annotators want to express in their literature model and how familiar with the document they are.

We learn from these incidents that spotting filters to extract elements from the text can be helpful. Annotator a12 expresses that she "*will play around with this (the sentence importance filter) a bit more, cause I like things coming out automatically.*"

Uses of spotting filters

The following extracts present different uses annotators make of the suggestions extracted by the spotting filters.

- Suggestions can reduce the document to a set of potentially interesting placeholders:

  Listing 6.8: Beginner a12 at 21:55

  ```
  {looking_for_ideas}
  ''So if I want to see arguments with 'CitiTag'''
  Note: she's using the right combination: concepts (I had swi-
  tched them on, but she  didn't remove them), and rhetorical
  zones. She's looking for areas combining the concept and a red
  sentence.
  {/looking_for_ideas}
  ```

  This extract also shows how well the visualisation works. Annotator a12 looks for coloured areas to find out what to focus on next.

[9] As for most of the information extracted from this qualitative analysis, we can only rely on our (as analysts) understanding and perception of what is happening. For instance, there is no way to guess that annotators create a concept from scratch and thus do not react to an element spotted earlier and currently deactivated.

[10] This 'total' includes the instances of '*typing or selecting*' and '*looking for ideas.*'

- Suggestions can also be activated to discover (and reuse) existing concepts:

Listing 6.9: Beginner a10 at 10:38

```
{looking_for_ideas}
Note: about the 'highlight concepts' button
''So this is a whole set of concepts already available. I can
do a first parse over the document''
{/looking_for_ideas}
```

Existing concepts add a layer on the document, representing the perspective of a10's peers.

- They can be used to position an argument with respect to peers'...

Listing 6.10: Expert a2 at 8:21

```
{looking_for_ideas}
''Now I'd like to see if anybody else has said things like
these''
''So I guess if I am going to the history''
Opens history window.
{/looking_for_ideas}
```

- ...to find out how a particular concept has been used over the corpus...

Listing 6.11: Beginner a11 at 24:36

```
{looking_for_ideas}
''I want to see what the other persons have said about Aqua''
(She got the feel of it immediately)
Makes another query
Browses results
{/looking_for_ideas}
```

- or to find peers' concepts and claims:

Listing 6.12: Expert a4 at 1:34

```
{looking_for_ideas}
''So History is where I find concepts made by other people''
Opens that window
Reads its content
{/looking_for_ideas}
```

- Suggestions can indicate which concepts are associated to a cited document,

Listing 6.13: Expert a1 at 4:01

```
{looking_for_ideas}
Clicks on the Magpie paper: ''What have we got about Magpie ?''
Finds a relevant concept there.
Adds it.
{/looking_for_ideas}
```

- or how a cited document is assessed by the author:

Listing 6.14: Expert a1 at 14:40

```
{looking_for_ideas}
Switches on the argumentative zoning and looks for the yellow
sentences
...
Creates a concept by selecting text from a yellow sentence in
the document
{/looking_for_ideas}
```

- Suggestions can also be used to reduce the amount of information on screen by focussing on a particular area of the document, for instance by submitting a query on a particular term "*because this word would bring me to the right regions of the paper.*" Typical choices for such areas include the abstract ("*I think major concepts (...) come there, at least once*") or other sections using a suggestive header such as 'Previous work.'

- Finally, they can be used to hide one of several sections, in particular when annotators have gone through all they wanted to say about it: "*With less scrolling (in the interface), it's a little bit easier.*"

Although a few spotting filters do exhibit some unwanted results, it is not as crucial as it could be expected. Of course, the higher the quality of the suggestions, the better. Their intrinsic goal, however, is to make annotators react and make them ask themselves '*Is this spotted element relevant to me?*'

Code	All	E	B	a1	a2	a3	a4	a5	a6	a7	a8	a9	a10	a11	a12	a13
Starting from the concepts	45	8	37	0	6	2	0	9	1	2	10	5	2	7	1	0
Starting from the relation	21	16	5	2	4	0	10	0	4	0	0	0	1	0	0	0

Table 6.14: Repartition of the use of the codes '*starting from the concepts*' and '*starting from the relation*' for the entire group of annotators ('All'), the groups of experts ('E') and beginners ('B'), and for each annotator ('a1' to 'a13.')

A new annotation process?

It indeed seems that the presence of suggestions shapes the annotation process, as annotators successively assess each element suggested and consider whether it is worth modelling or not. Annotators activate one or several filters and react to their output, by selecting a group of words and creating concepts and claims out of them.

Suggestions may be shifting the annotation a document from a situation in which annotators are left unsupported to a situation in which they 'only' have to decide, for each suggestion proposed (and independently of its intrinsic quality), if it is good material to make a concept or a claim or not. Annotation may be becoming a matter of answering '*yes*' or '*no*' to the question implied by each proposed suggestion: '*is this particular element relevant to you?*'

Starting from a relation vs. starting from concepts

We also observe a striking difference between how (mostly) experts start from the relation type they want to use for a claim and how (mostly) beginners start from the two concepts they want to put in relation, without knowing if the relation type they want actually exists.

45 incidents tagged with the code *starting from the concepts* are recorded, 8 of them being made by two expert annotators, and the remaining 37 by 7 beginners. 21 incidents tagged with the code *starting from the relation* are recorded, 16 being associated to 3 experts and the remaining 5 to 2 beginners. Table 6.14 gives the full breakdown of these two codes.

This phenomenon, coming from, *à priori*, the level of proficiency with the formalism, is not completely unexpected. It is nevertheless interesting to notice it experienced in real situations. The following two extracts are particularly enlightening:

Listing 6.15: Expert a4 at 20:46

```
{starting_from_relation}
''I want to say that (the left concept) does not... or impairs...''
```

```
{/starting_from_relation}
```

Listing 6.16: Beginner a6 at 36:17

```
{starting_from_what_i_want_to_say}
''I want to express that two terms are used for each other all the
time in the document. What do you suggest?''
{/starting_from_what_i_want_to_say}
```

In the former example, expert annotator a4 has in her mind the relation she is about to use, or at least an idea of it (***impairs*** is a relation available in the ontology.) In other words, she knows the different articulations the ontology offers to connect concepts. In the latter, beginner annotator a6 is trying, in a way, to 'squeeze' her interpretation in the ontology.

If the difference between what the annotator wants to express and what is available is too important, it becomes impossible to capture this interpretation (c.f. our discussion of the *cannot find a relational type* code in the formalisation theme.) Some parts of the interpretation are lost. Additionally, the annotator may feel that the system is not interested in what she wants to express. This situation must be avoided as much as possible.

Although the *starting from a relation* incident is found more frequently in experts' annotations, counter-examples can also be found. Expert a2 starts from the relation he wants to use 4 times and from the concepts he wants to combine 6 times. Beginner a5 starts to model a claim with an idea of the relation she ends up using 4 times out of 5.

Incremental formalisation

The *incremental formalisation* code (in the *strategy* theme, c.f. table 6.13, page 179) is related to the way annotators progress in their sense-making process. We observe that annotators typically start by submitting a few concepts and then combining them into claims ("*I'm gonna go through it (the document.) Noting some of the concepts. Sort of play with the tool.*") After this initial contact with the environment, further concepts and claims are added, reusing or building on the ones already submitted.

Annotators break their sense-making process into multiple steps, by focusing on a particular area first and then moving to another area [11]:

Listing 6.17: Expert a1 at 14:26

```
{submitting_incrementally}
```

[11]The code 'submitting incrementally' used in the following extract is a sub-code of 'incremental formalisation' in our taxonomy (c.f. table 6.13, page 179); therefore, every instance of the former is also an instance of the latter.

```
''I've said the things I wanted to say about ESpotter (ie the con-
tribution of the paper), and I know want to tie it into the pre-
vious work''
{/submitting_incrementally}
```

Concepts and claims are however not immediately submitted as soon as they are typed. Instead, they are often kept on the screen because annotators feel the need to have them visible to make their articulation into claims easier. Saying aloud the relations is also a phenomenon we often notice: annotators say aloud all the potential relations they are browsing before stopping on the 'right' one. We expect that such behaviour is needed to find the relation that 'sounds' the best.

When concepts are submitted, we also notice an interesting phenomenon: annotators frequently check a trace of their own annotation work with the '(My) Matched concepts' spotting filter or via the 'History' window. The fact that the interface is able to give immediate feedback and (probably instant gratification) by displaying their work seems to be appreciated [12].

Another strategy-related aspect we notice is related to the order in which annotators access the different resources at their disposal. They seem to focus first on their own annotation (possibly to get their feet wet with the formalism) before browsing through the history and looking for relevant concepts and claims from their peers. It may have to do with the settings of the experiment and with the need to 'get something done' by the hour ("*For the time given, the easiest thing is to see the system suggestions and make your own. Because go back and look through the history may just take too much time.*")

It may also have to do with a need to appropriate the document first, to make it their own, before deciding to consult what their peers have recorded about it:

Listing 6.18: Beginner a8 at 49:30

```
{incremental_formalization}
''What I would also do is create all these things, and then see what
others people have done.''
''I wouldn't go and see what people have done first.''
''*I* want to understand the paper.''
{/incremental_formalization}
```

It seems to show that the participatory dimension is only a bonus: annotation is first of all done for oneself.

[12] This may indicate an unwanted focus on visibility though. We come back to this point later.

Writing notes Although notes are not frequently added to a document (only two annotators make use of them), their use reveals a few aspects related to incremental formalisation. In these two cases, notes are used to enable annotators to keep in mind an idea they are not ready to formalise yet, thus mimicking the function of 'real' (i.e. physical, hand-written) side-margin comments (c.f. Marshall's taxonomy of the roles played by physical annotations on text book, section 4.2.1, page 41.)

Listing 6.19: Expert a2 at 31:55

```
{submitting_incrementally}
''Typing things before writing a claim is probably useful'' (parsing
the note has created a claim)
''You can type your thoughts, the way you'd do with paper.''
{/submitting_incrementally}
```

Conclusion about the strategy analysis

We have presented in this section different aspects of the strategy deployed by annotators to build their literature model. The use of varying levels of suggestions is interesting. It proves that flexibility is a crucial parameter and that we cannot account for the different needs annotators have in different occasions.

The multiple uses annotators have made of the suggestions is also a good indicator of the possibilities offered by the environment.

We have also discovered that suggestions seem to transform the annotation process, implicitly asking annotators to keep or reject each suggestion proposed by the filters:

Listing 6.20: Expert a3 at 2074.92

```
{interface_misc}
''We need visual reference points''
''Representing visual differentiations'' ''Conceptual differentia-
tions''
''Representing visually helps a lot''
''It's like providing you anchors, to get into the text''
''You find reference points to start from''
''That is extremely useful''
''But it's still in the context, in the context of the text.''
''It's not like taking these sentences and isolating them''
''It is taking them and highlighting them, which is completely dif-
```

```
ferent operation and I think that's very useful''
{/interface_misc}
```

It also seems to indicate that suggestions may not need to be perfect (at least for students) and that that they may indeed have to be imperfect to keep a level of awareness and criticism in an annotator's mind:

Listing 6.21: Expert a3 at 2532.88

```
{looking_for_ideas}
''These (the relations (marked as blue areas)) give me suggestions
or hints for relations''
{/looking_for_ideas}

Tutor: ''it could be wrong''

''Yes, but again, you go there, you check it out''
''And if it's not useful, you just disregard it''
''If it's useful, you can use it, which is nice''
```

Elements of an incremental formalisation strategy are found out, with annotators starting by inputting a few concepts and claims first, before considering peers' annotations.

We have finally found out that beginner annotators are more likely to start from the concepts they want to connect in the way they want, without knowing if that particular relation is available. While such behaviours obviously depend on the level of proficiency and experience with the formalism, it also means that we must look at ways to make annotators think more in terms of existing relation types. We do want to avoid such situations:

Listing 6.22: 'Beginner a5 at 21:04'

```
{starting_from_what_i_want_to_say}
''Because what happens is, if I create a concept, then another one,
then I paste them in the relation, and then I realise that there's
no relation for them, I lose a lot of time''
{/starting_from_what_i_want_to_say}
```

One way to address these difficulties could be to *guide their thinking* towards claims that are endorsed by the ontology. This idea is developed in the following chapter.

6.4.8 ClaimSpotter usability: interaction design

We study in more detail annotators' interactions with the interface. We have found that segments of transcripts annotated with the theme code *interaction design* are a very useful source of suggestions for future versions of ClaimSpotter.

Intuitiveness

As we have seen from the summary statistics and from our study so far, annotators, whether experts or beginners with the formalism, manage to input a substantial amount of concepts and claims and make reasonable use of ClaimSpotter features. Annotator a11 goes as far as adding a document to the repository. Three annotators, experts a2 and a3, and beginner a12 even use claim-chaining, connecting a claim into another claim.

It seems to indicate that the environment is reasonably intuitive, at least for the particular process of annotating a single, well-known, scholarly document. No annotator is stuck wondering what to do next, and although occasional questions are asked to the tutor, none of these prevents annotators from carrying their task on.

Mistakes during interaction A few mistakes do happen however. We define a mistake as an action carried out by an annotator that does not match her intent (which she verbally expresses.) The following extract gives an example of a mistake:

Listing 6.23: 'Beginner a12 at 8:22'

```
{mistake_during_interaction}
She wants to add a concept.
Clicks on 'Add' (a relation)
''What did I do ?''
''No''
''Remove'' (Removes the relation)
{/mistake_during_interaction}
```

Paying attention to where these mistakes happen gives us ideas to improve the interface. Most common mistakes result from a confusion between controls for concept and claim operations (such as the 'Add' button in this particular example; similar mistakes also involve the 'remove' button), the use of the form panel scrollbar instead of the main window scrollbar and the use of the '[X,.,.]' and '[.,.,X]' buttons. There is also confusion over the role played by each scroll-bar (the one of the document and the one of the input form panel.)

These examples are instances of interaction breakdown, requiring additional effort from the annotator to resume her task. They are important to fix, for the time she spends on understanding the interface should be spent thinking about concepts and claims.

Filters misunderstandings

Other problems are related to misunderstandings of the role played by filters. The rhetorical filter has proved especially difficult to understand. Two annotators want to change the areas that are coloured (in particular the OWN areas, indicating contributions.) An interesting point is that they have both co-authored the document they are annotating. This may explain why they want to have some additional areas spotted and highlighted, as they are certainly more aware of where exactly their contributions are stated.

Confusion is also experienced for the 'matched relations' filter. Annotators do not always understand that sentences containing an instance of a ScholOnto relation are not ScholOnto claims (i.e. formally encoded as claims) but only signals of the areas where an author defends her position using ('involuntarily') a verb defined in the ontology of discourse relations.

These difficulties indicate that the notions of patterns and rhetorically-consistent zones are not as straightforward as the project team thought they would be. Bringing the system in the real world (and outside of the project team) is indeed most illuminating. Aspects and notions that the project team has taken for granted over the years are shown to be not as intuitive and easy as expected.

Feedback

The tool-tips associated to various buttons and links in the interface help disambiguate their actions:

Listing 6.24: Beginner a12 at 4:34

```
{using_tooltips_to_decide_what_to_do}
Looks for the 'Add a concept' button.
Is about to press the 'Add a relation one, but the tool-tip which
seemed to prevent her from doing this mistake
{/using_tooltips_to_decide_what_to_do}
```

The presence of pull-down lists containing concept types and relational types on the screen is also helpful as a source of scaffolds ("*I'm looking through the types because I'm not familiar with them.*")

Listing 6.25: Beginner a12 at 9:46

```
{onscreen_text_is_helpful_to_create_concepts}
''Hang on. Let me see the definition of concepts, cause that actual-
ly helps me''
Switches on the list of concept types (from a blank concept she has
created just now)
{/onscreen_text_is_helpful_to_create_concepts}

Tutor: You said that the definition of concepts types was helping
you. Because it can give you an indication about what can be a con-
cept, what to put?
''Yes. Otherwise, anything can be a concept''
```

This particular extract gives another hint that more scaffolds may be needed to assist the annotation. Presenting the different concept types available can help annotators focus on aspects of the research reported such as the 'problem' it tackles or the 'methodology' it proposes.

Feedback is also looked for when annotators check existing concepts in the document, right after they have submitted them (c.f. our study of the *incremental formalisation* code.) This is relevant to concepts they create via selecting and pasting text without modification (every concept is accessible in the 'History' window in any case.) Here again, the ability of the interface to give immediate feedback is valued. Information is updated as soon as it is submitted.

Consistency

Statements about the lack of consistency on certain aspects of the interface are also noted. They are mostly related to the use of different terms to refer to a similar element, and to differences in the style and writing conventions used on different screens. These elements have been fixed as soon as possible.

Support

The '[X,.,.]' and '[.,.,X]' shortcut links are particularly appreciated because they encourage a playful approach: the act of combining concepts into a claim is not only made easy (as there is no need to retype them), but it also has a 'combining-bricks' aspect that seems to be pleasing. As annotators spend quite some time creating concepts out of the mass of information, they may expect that making claims is going to be even more tedious. Instead, a series of shortcuts to make the formulation of claims similar to a construction game is offered:

Listing 6.26: 'Expert a4 at 9:30'

```
{support_to_create_relations}
''I love it'' (the [X,.,.] and [.,.,X] buttons)
''It's nice and easy''
''So far, I am quite enjoying that I can quickly put concepts in a
claim''
''It's a good feature''
{/support_to_create_relations}
```

The 'flip' button is also welcomed [13]:

Listing 6.27: 'Beginner a10 45:43'

```
{changing_the_relational_type}
{support_to_create_relations}
''Oh, 'flip' is good. We like 'flip.' It's a very human thing''
(to put concepts in there without knowing how ordered they will be)
{/support_to_create_relations}
{/changing_the_relational_type}
```

Writing makes ideas clearer

The possibility to add free notes to a text in the environment has received a mixed reaction. They are presented during the tutorial session as a notepad replacement, or, more correctly, as a notepad which can be used by annotators to keep track of the ideas they are not ready to (or cannot) conceptualise yet (in a way, as a buffer zone between the text and the concepts and claims.) We have chosen to represent them as yellow rectangles.

This has been a wrong design decision. We notice that notes confuse annotators: they believe that they can be attached to any location of the document and therefore that they are similar to side-margin comments:

[13] It is not included in the version of ClaimSpotter used by the first two annotators. Its need has become immediately clear: these two annotators have been at times combining their concepts in the wrong order, either by mistaking the '[X,.,.]' and '[.,.,X]' buttons or because of their lack of knowledge of the available relations. This has resulted in them having to remove the claim, create a new blank one, and then input concepts at the right position. This tedious process has been solved with this 'flip' button.

Listing 6.28: Beginner a10 45:43

```
{problems_issues_with_interface}
The concept of note wasn't immediately obvious, but the fact that
it was yellow made it clear.
''It explains its purpose.''
''It's an affordance.''
{/problems_issues_with_interface}
```

A future version of ClaimSpotter will implement such notes.

Suggestions

Annotators' suggestions are mostly related to the interface or to the nature and quality of the extracted elements. Annotator a5 expresses his need of a list of available relations displayed at anytime in the interface. This comment is related to the one on the relevance of the pull-down list of concept types as a source of scaffold: having the list of relations visible at any time could guide annotators towards the statements they are allowed to submit.

Several suggestions are related to the quality of the suggestion filters. One common request is to merge the plural form of any spotted concept with its singular form and vice versa. For instance, if a concept 'ontologies' is defined in the repository, any instance of 'ontology' should also be spotted in the text and highlighted.

Another suggested improvement is the addition of a mechanism - a contextual menu associated to a text selection or a hot key - to immediately make a concept out of the currently selected text area (instead of having to drag it with the mouse towards the input form.) It is a very understandable requirement, as we have noticed multiple pointer trips (and therefore, attention shifts from one part of the display to another) while annotators create their concepts. It is another interaction breakdown: annotators' attention is transferred from their model construction activity to an artefact of the interface. A future prototype will feature this facility.

Conclusion about the interaction design analysis

The 'interaction design' questions we have identified in our analysis of our second research question have guided our approach. Several aspects such as the 'playful' element brought to the construction of claims are promising and contribute to let annotators experiment at little cost with different arrangements of concepts and relation types. For this particular point, the interface is useful and reasonably intuitive. It is more difficult to identify further aspects, as there has not been enough time left to try every feature of the interface. A few problems have also happened only once, making the diagnostic of a potential breakdown more difficult. We can only hypothesise that the presentation of the document, its visibility and the possibilities to modify it are appreciated too, since they are used frequently without raising comments.

This interaction-design facet of our analysis has helped us identify and address several interaction breakdowns. The suggestions formulated have also given us leads for future improvements. Chapter 7 extends the discussion on these points.

6.4.9 ClaimSpotter usability: miscellaneous issues

In this fourth and last 'theme', we list several observations.

Information overload

While the presence of suggestions seems to shape the annotation process (and to make it easier, or at least more reactive), we may be proposing too much information and too many sources of potential concepts and claims:

Listing 6.29: Expert a4 at 22:39

```
{looking_for_ideas}
''The problem is 'do you make your own claims', 'do you follow the
system', 'do you go back to the history to see what the other peo-
ple have'''
{/looking_for_ideas}
{looking_for_ideas}
''There's so many different sources of getting claims''
{/looking_for_ideas}
{strategy_misc}
''For the time given, the easiest thing is to see the system's sug-
gestions and make your own. Because go back and look through the
history may just take too much time''
{/strategy_misc}
```

We acknowledge that for a one-hour experiment, there is indeed a lot of information to understand and digest. More experiments would have been needed to introduce the different sources of support and to let annotators decide which ones work best for them. This could have been supported by the creation of a user profile, in which annotators would specify the filters they prefer. We come back to this point in our discussion chapter.

'Document-centredness'

We have reported earlier on the limited number of submitted claims connecting concepts defined in different documents. While the limited amount of time available for the experiment could again explain it, it is also true that ClaimSpotter, as it is, may not be especially suitable to model such connections. The importance given to the current document may make it even more difficult. Different interfaces would need to be devised. We come back to this point in our discussion chapter.

On user 'lazyness'

We have reflected earlier on the annotation process being moved from answering a question such as '*What makes this document relevant to me?*' to answering questions such as '*Is this particular suggestion relevant to my literature model?*' We wish to play the devil's advocate now and ask if annotations are not 'better' when they have to be thought from scratch. This may sound like being at the opposite of the point of this thesis, which is to investigate ways in which a document-centric annotation interface can help. This voluntarily provocative question is however worth studying.

To create concepts without any particular assistance from the text, one has to build a map of the document and to break it into concepts and claims [Buckingham Shum et al., 2005]. While the goal of this interface was originally to help this process by suggesting leads, we observe a tendency [14] to create concepts out of highlighted data (which is in itself not a bad thing.) Some of these copied and pasted segments are modified to suit one's ideas of course, but they are nevertheless heavily inspired by the highlighted suggestions in the original document.

If it is a good thing on the user side, as it lessens the difficulty associated to the construction of a literature model, there is however a risk that less effort is put into the annotation, as the user expects the system to bring her the salient facts about a document (whether these are composed of important sentences, or matched existing concepts.)

In this case, keeping suggestions at an imperfect level, including a few irrelevant elements (for students) may be needed to keep annotators' attention. If suggestions were too good, annotators may be tempted to accept them without any critical assessment.

Matched concepts Let us now consider the 'matched concepts' filter. As it gives immediate feedback to annotators submitting concepts copied and pasted from the text (it relies on

[14] c.f. our our earlier explanation of the 'looking for ideas' code, page 188.

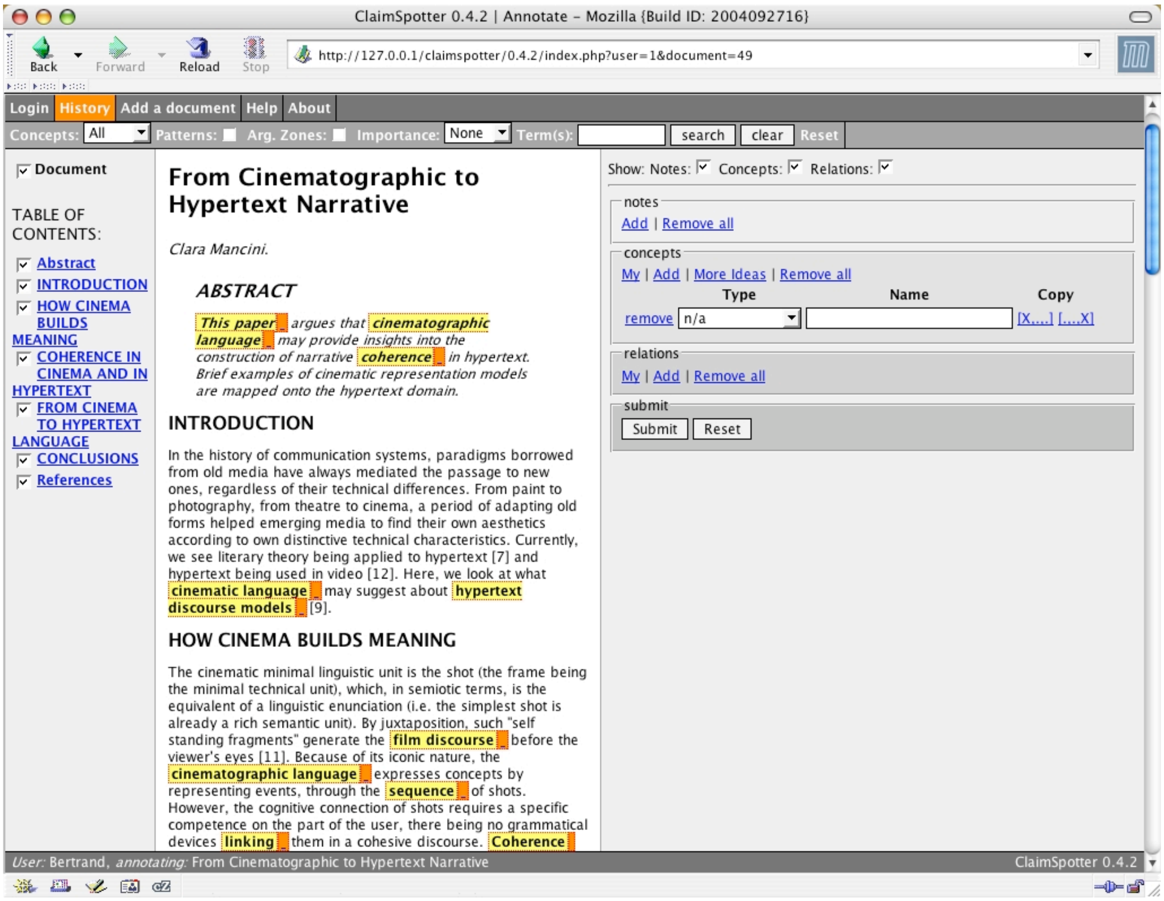


Figure 6.8: Matched concepts spread over the document (yellow areas) suggest that it has been annotated (and read) thoroughly. They are displayed in a more efficient manner than non-matched concepts in the current ClaimSpotter.

exact matches in the document), the emphasis may be shifted from submitting ‘well-thought’ concepts to submitting ‘visible’ (matched) concepts (cf. figure 6.8.) We do not imply that visible concepts are not good or that concepts that are not visible are better. We merely acknowledge that matched concepts are privileged over non-matched ones: the former are visible via the activation of a filter and rendered with bright yellow zones, while the latter are ‘hidden’ in a secondary window (the ‘History’ window.) The blame is on ClaimSpotter. Better presentation options must be devised for a future version, including a mechanism to display both ‘matched’ and ‘non-matched’ concepts on the main window.

Although this has not been verified, it may be that ‘matched concepts’ are equivalent (for annotators) to the whole set of concepts defined for a document. This may lead them to submit copied-and-pasted concepts only. This added focus on visibility may also mean that matched concepts have become a way to ensure that a document is ‘covered’ with concepts throughout its sections. That is, that they have become a way, for annotators, to show to their peers that they have been through a document (by covering its content with bright yellow bricks.) This should not surprise us: as annotations become a part of the document, they participate to its social life, a phenomenon that is extensively reported in [Seely Brown and Duguid, 1996].

6.5 Questionnaire

The third part of our evaluation study is based on query techniques: a questionnaire has been sent by email to annotators at the end of the experiment. They have been asked to take a few days to answer it.

The questionnaire is designed to receive their points of view on the main strengths and weaknesses of the interface, and on the ways it can be improved. This has provided another source of information, giving annotators more time to reflect. The following tables give the (unedited) answers we have received (we have not received answers from annotators a1, a3, a6 and a13; a2 and a7 have answered out of the canvas provided: their answers are given in the first table.)

6.5.1 Evolution and improvements

Table 6.15 lists the answers we have received for the first question, related to ClaimSpotter possible improvements.

WHAT WOULD YOU LIKE THE TOOL TO DO IN THE FUTURE? HOW COULD IT BE IMPROVED?

a2	The one key comment I have (and actually I have been thinking about this for quite a while) is that we really need top-down support for generating scholarly narratives both when writing and when interpretating papers. In other words, one issue is that if I use ClaimSpotter I end up making lots of low-level claims, which are probably not so interesting. While ClaimSpotter could be a bit more proactive and guide me through a narrative (or scholarly model) driving the generation of claim. One possible model could be to start with the problem, then the approach, then the alternative approaches, then the relation between the approach and the alternatives, etc. Different communities may have different narratives, so you can imagine Claim-Spotter could maintain a library of these.

WHAT WOULD YOU LIKE THE TOOL TO DO IN THE FUTURE? HOW COULD IT BE IMPROVED?

a4	I think we talked about this a little bit during the session, but the ideal time-saving feature that I would like to see included would be a kind of automatic claim recognition, where as I am making a claim about something the system alerts me that someone has made a similar claim previously, and that way maybe I would be able to reuse the other person's claim entirely. But that would be in an ideal situation if I could wave a magic wand. As it stands the History part of the system is reasonably adequate for the claim reuse. Maybe you can just touch up the interface a bit to make it easier to visualise the previous claims.
a5	It will be nice to provide some reasoning functionalities embedded in your tool. For instance, once you create the relations then it will be useful to reason about these relations to evaluate the claims. When you select a particular part of the sentence to make the claims based on the existing relations, it will be useful to have a pop-up box (similar to what you get while programming in Java or in Visual Basic) to show all the existing relations such that end-user can decide which relations are appropriate for that particular claim instead of switching the main text and the relations that exist in the right box of the interface.
a7	I would like the tool to automatically generate claims based on some sort of information extraction techniques, I could then just review them and make necessary changes. New concepts and new relations could be learned from documents as well. Make the formalisation of more complex claims possible, such as claims involving more than two concepts. I am thinking about how you are going to use these claims grouped by each document, will some applications be developed?

WHAT WOULD YOU LIKE THE TOOL TO DO IN THE FUTURE? HOW COULD IT BE IMPROVED?

a8	What I liked immediately about the tool is that it may be a way of extracting key points from a paper - it extends the margin note. The idea too, that once you have the initial annotation, you can add to it on subsequent reads of the paper and so you get a kind of history of your thoughts on a paper. (This would be the best feature on first impressions) Related to this is the facility for viewing other people' annotations. Whether you agree or not doesn't really matter because all those people are reading the paper for slightly different reasons. And so this facility can serve to broaden one's perspective.
a9	The tool is quite useful in terms of finding relationships between various parts of a document or cross documents. I hope in the future, I could use the tool to find related literature for my study. For the improvement, I think the user interface is not quite user-friendly when creating concepts and relationships, although there are practical limitations.
a10	Generally, a really nice tool. Possible improvements: some of the specific issues I raised yesterday, e.g. consistency of location of [X,.,.] [.,.,X] buttons within history, being able to easily see the previously saved (submitted) concepts and claims…It would be useful to try testing it by cross indexing two or more docs rather than just one.
a11	I think it could be useful in an academic domain for annotating someone else paper if you are the second author or the reviewer.
a12	Visualise the relationships I have put into it of course! I need to see them visually represented in order to understand the structure of my arguments and also to see where most of the focus goes. How could it be improved? Usability-wise I think I have commented quite a lot during the session, some things are not evident, and for instance, I would like to submit the relationships without the concepts disappearing and me having to go to history to bring them back again. 'Don't make me think' as a usability principle here.

Table 6.15: Evaluation questionnaire. (1/3)

Answers can be organised in four main ideas: support for reading, support for extracting elements, visualisation techniques and usability aspects.

Support for reading Annotators a2 and a9 emphasise the need for a strategy to support efficient reading and interpretation of a document, which would help the user get down to the main points of a document and understand key elements such as what the problem addressed is, and how it is addressed. While the ability to read and interpret a document is not directly dependent on ClaimSpotter/ScholOnto, there is a chance that an annotation tool can help learners acquire this skill. As a first approach to provide this additional source of support, a top-down workflow is presented in the following chapter.

Support for extracting elements Annotators a4 and a7 ask for more advanced extraction techniques, going as far as claim extraction. We have seen how far we could get with the choice of techniques we have adopted so far (noun groups extraction and matched ScholOnto relations) and it seems to cover a theoretically (assuming we could have extracted all of them) major part (two-thirds) of the concepts that have ended up being submitted. New techniques can however be added, due to the openness of the ClaimSpotter architecture.

Visualisation A graphical visualisation of the concepts submitted and of their articulation is also demanded by annotator a12. This indicates that although a text-based representation is acceptable, there is a need, at some point, for a graphical representation of the network of concepts and claims. It also means that a flattened textual representation for a literature model is manageable in simple scenarios only, in which there is only one document to annotate. The (static) graph export facility added to ClaimSpotter (c.f. page 145) is a first step in this direction.

Usability The interaction breakdowns noted by the users have been fixed when possible.

6.5.2 Annotation

Table 6.16 lists answers to our second question. It was related to the models annotators have created.

WOULD YOU SAY THAT THE TOOL MAKES YOU SAY MORE OR LESS? CAN YOU EXPLAIN WHY?

a2	

WOULD YOU SAY THAT THE TOOL MAKES YOU SAY MORE OR LESS? CAN YOU EXPLAIN WHY?

a4	I would have to say that the tool made me say more than I had originally planned. I think the main reason for that was it is very easy to make claims with the system. You can just cut and paste text very easily into right and left nodes and this means that you can construct claims with minimal effort. However, at the risk of contradicting myself a bit, even though it is technologically easy to construct claims, I found myself taking a lot of time editing the claims that I constructed because I wanted to make sure that the claims made sense. I think that was the case because I was aware that the claims would become 'public.' I knew that they were being saved so that other people could view them so I tried to articulate my claims very carefully. If the system had been for my own personal use I probably would not have spent so much time editing and refining the claims I made. So my answer to the question is yes, the system made me say more than I had originally planned because it is easy to construct claims; but at the same time I was constrained a bit by the fact that my claims would be public. So maybe if the system was private I would have said even more.
a5	This particular tool definitely allows you to tell more things than that are you intend to say beforehand. For example, because it allows you to create new claims in terms of the entire sentences or even the new concepts it adds another layer of expressiveness to your document. However, it will be nice if your tool allows adding new concepts [15] as well because it will provide with a seed hierarchy of these concepts.

[15]Note: ClaimSpotter does of course allow the addition of concepts.

WOULD YOU SAY THAT THE TOOL MAKES YOU SAY MORE OR LESS? CAN YOU EXPLAIN WHY?

a7	The tool indeed helped me to think about some relations more clearly so that I can put them into those triples, while on the other hand, there are some relations that are hard to be decomposed into triples so that I gave up. So I said more about certain things and less about other things. The reason is that the rigidity of the format which can be used for relation formalization can help decompose some ideas while at the same time have limit abilities to show others. For example, if we say A concept is better than B concept under the condition that A is a sub-concept of C concept. The tool is a synergy of different techniques such as text mining, bibliographical analysis, databases for scholarly study. I like the idea to show argumentations, conceptualisations in an article into triples, similar to RDF idea. On the other hand, I suspect the model is over-simplistic and the article itself is a much more accurate illustration of what it is about than triples. Sure that the model can help machine analysis and a quick overview of the article.
a8	For improvement, I would allow the user to create relations as subclasses of basic system provided ones. Of course, this will introduce problems for a shared resource but you might then allow users of different domains to say: for purpose P, let us regard relations X and Y as equivalent and so on. The interface look was very neat, clean and well organised. Size was a bit small. I might display the concepts and relations widgets in such a way that they are easily distinguishable and have the submit button directly below the last input box so that it's always visible. I came into the room with about two dozen concepts but forgot most of them. I should have looked at the document on the screen but didn't. I'm sure you have a facility for cutting from text and pasting to concept box - ordinarily, I think I would use that. I would want to drag out concepts as I'm reading the text and then add relations as and when they come to mind[16]. The tool might make me say more but more importantly, I think it would allow me to organise better what I want to say and so might actually reduce it. The only reason why I didn't say more is the time allowed and having to learn the functionality. I really needed more practice time.

[16]Note: it is indeed possible in ClaimSpotter.

WOULD YOU SAY THAT THE TOOL MAKES YOU SAY MORE OR LESS? CAN YOU EXPLAIN WHY?

a9	To me, the tool seems to make me say more things, but I'd like to think these things as tentative thinking but at least, this tool gives me the opportunity to record these things, which could be quite difficult to keep track of. The tool gives me a way to record some bits of interesting info in the document.
a10	For somebody like me who has not encountered this research territory much before, it feels like you are building a really practical tool, very useful for helping researchers analytically consider the content of documents and mark them up more meaningfully. It really helped me think more about the document I was looking at. They always say that as a PhD student you should critically read papers, but its easy to scan them. Your tool brings the critical analysis to the fore.
a11	I miss to have a way to relate concepts or notes to specific paragraphs in the document. If I remember well I think I end up saying as much as I wanted to say. Also, maybe it would have been interesting to be able to highlight parts of the text.
a12	I have to say I am not at all familiar with the research area, so I had no expectations and a lot of curiosity.

Table 6.16: Evaluation questionnaire. (2/3)

We can identify the following themes from these answers: support, model and visibility.

Support Although a need for a top-down strategy is expressed by two annotators in their answers to the previous question, three annotators feel the tool already helps them in their approach of a document. It helps them 'organise better' their ideas, and 'consider analytically their contents.' While this is an aspect that it is more related to sense-making processes than to ScholOnto in particular, a tool such as ClaimSpotter can help analyse a document.

Model The model is thought over-simplistic for one annotator, as it does not enable him to express claims '*A is in relation with B because of C.*' However, such statements can be expressed in ScholOnto, via claim-chaining. A first claim can connect A and B, and a new claim connecting this claim and a concept C can be created. It is true nevertheless that this is not trivial for a first-time user. Better guidelines could be provided, including several tricky situations and ways to model them. An extension of the model with more relations is also listed as a requirement by one annotator.

Visibility Annotator a4 raises an interesting point: the time spent on his claims to make them look good. He is the first annotator expressing his awareness of the public, visible, aspect of the claims (a4 is an expert, thus familiar with this aspect of the project.) The tool may help annotators say more, by providing facilitating ways to create and organise concepts and claims, but it may also make them spend more time on their annotation because of its visibility.

Annotator a4 also mentions how he would have behaved differently in a more restricted context, where, potentially, his claims would have been only visible by a few persons. This seems to suggest two classes of statements: the ones one feels rather 'safe' about sharing (whether because she is an expert in the field, or the author of the annotated document for instance), and the ones which are more 'tentative', in the sense that they reflect an opinion [17].

6.5.3 Positive and negative aspects

The last question is concerned with the most positive and negative aspects annotators experienced in the course of the experiment. Table 6.17 lists the answers we received.

WHAT IS THE ASPECT YOU LIKED THE MOST AND THE ONE YOU LIKED THE LEAST?

a2	
a4	What I liked most: I liked the fact that it was easy to learn the claim building mechanism. I also especially liked the 'Relations' recommendations that the system made, because this added an extra dimension to the claim building process. I also liked the fact that the document was part of the interface (which is a huge advantage over ClaiMaker) and that I could choose to view the different sections of the paper. What I liked least: the two scrollbars to the right are very close to each other. So sometimes when I wanted to scroll down the page I would accidentally use the scrollbar for the claims section, and vice versa. Also, (I am not sure if this is already the case but I didn't notice it), it would be nice to be able to hide the claims that I had already made to make the screen less cluttered, but without submitting them.

[17] The ClaiMaker prototype enables the definition of private claims.

What is the aspect you liked the most and the one you liked the least?

a5	One of the strongest aspect of this tool is that it is designed in such a way that it takes into account different types of users such as novice and also those who are expert in the domain and as a result the overall navigation by using this tool is quite straightforward and easy. The experiment is well designed and it may be because of the ease of the use of your tool but throughout experiment end-user deos not face any ambiguity about the different types of claims he/she wants to make.
a7	
a8	I can't remember all that you showed me on that menu bar and there are probably things there that I should choose as least favourite simply because I wouldn't use them. The only thing I remember is other peoples' annotation and that is a good feature as I have already said. Based on what I remember, I shall just say that the worst feature is the facility for compacting the document; that's not very helpful if you are scanning for concepts. On the other hand, I do like the subsection titles displayed alongisde the document window.
a9	One thing I like most is those shortcuts provided by the interface, e.g. quickly copy the concepts to relationship section by one click, which allow me to quickly accomplish some frequent tasks. The thing I liked the least is the creation of concepts, couple with the fact that I had to scroll up and down to locate the newly created concept line at the top of the concepts section but below those button in this section. This might save some scrolling when creating concepts.

WHAT IS THE ASPECT YOU LIKED THE MOST AND THE ONE YOU LIKED THE LEAST?

a10	The “flip” functionality. Very simple, very useful, human tool. Computer programmers often assume people are rational logical thinkers and put down their thoughts in the correct order every time, but we are human, maybe grab a ’concept’ term and add it, then realise the second term should be the first term. . . you offer a simple, one click method of sorting out the building of a claim, nice one. Second favourite thing? the ability to call up history and display it so as you work on more documents you don’t have to remember the terms you’ve used before (e.g. “This paper”.) Hey, make that the number one favourite thing (presses “flip”.) Third favourite - the yellow background for the “notes” box. Excellent ’affordance’ offered by the cultural cue/reference to PostIt notes. Thing I like the least? The Import button didn’t work.
a11	I like it the part in which you can see if other people have the same annotations in other documents, as for example for “semantic web” and the possibility of importing these concepts. The interface was nice. . . And not too complicated. As I told you for me the confusing thing was de “Delete” button instead of “Hide it.”
a12	Liked most: the automatic highlights of arguments, references, patterns and importance of statements in the document (tools on top menu) Liked least: the long and scary text page appearing after submit where you have to go to the end to find yourself again.

Table 6.17: Evaluation questionnaire. (3/3)

Answers can be organised in four themes: support, sharing, the role of the document and adaptability.

Support The environment seems to be efficiently both supporting the claim creation process and hiding its complexity. The possibility to try and change concepts and relations if they do not fit has ‘worked’ very well. Learning by doing is important for a cognitive task, and the ability to ‘try’ a claim by bringing two concepts in, modifying it by flipping their arrangement and deleting it if it ends up not being satisfactory is a key feature of ClaimSpotter.

Sharing The presence of peers' annotations is also well received. They can be appropriated, reused, merged with one's concepts and claims and combined to create new content.

Document The presence of the document and of its spotting filters is also well-received by expert a4, who is knowledgeable about the other interfaces. An additional study involving the two interfaces and beginner annotators could create an interesting comparison of the aspects one interface or the other is more suitable for.

Adaptability Finally, the adaptability of the interface is noted by annotator a5. The possibility to get as little or as much support as needed (in terms of the sources of information brought forward by the spotting filters, the 'more ideas component' and the 'history' window) lets annotators tailor their own claim-making experience.

6.6 Usage patterns

As this has been the first 'real' evaluation of ClaimSpotter, involving 'real' users in a 'real' scenario and performing a 'real' task, we have learned a tremendous amount. One of the most interesting aspect that we have learned is that - although, as one could expect, there were nearly as many strategies to approach ClaimSpotter and the annotation of a document as annotators - some usage patterns can be discovered.

Bearing in mind our limited samples, these moves nevertheless give valuable insight into the inter-relations between various aspects of the interface and of the annotation process. We did not have this knowledge before starting the design phase of ClaimSpotter. These patterns give us additional knowledge for a future design phase (c.f. section 1.4, page 6) by contributing to a better understanding of the activities carried out in an annotation process.

6.6.1 Examples

We list below a few sequences of actions we have observed frequently.

Answering 'yes' or 'no' to suggestions

The first sequence we have noticed is composed of the following actions: *switching on a filter*, *looking at an highlighted area in the document*, *reflecting on it*, *modelling a concept or a claim out of it*, and *moving to the next highlighted area*. This sequence shows the relevance of displaying highlighted elements in the document text, side by side with the claim input form. Annotators can instantly move from the input form to the document, and vice-versa. Highlighted elements act as attention-catchers throughout the document.

Annotating and checking for visual feedback

Another sequence we have noted is composed of the actions, *selecting, copying and pasting some text into a concept*, *submitting it*, and *activating the 'my concepts' filter to check it is here*. This move illustrates a consequence that we had not foreseen: the importance taken by 'visible' concepts.

Breaking the document in areas

A simpler but equally interesting sequence can be identified by looking at actions *jumping to a particular section in the document*, *reading/skimming it* and *summarising it*. It also shows that keeping together the object being annotated from the product of the annotation is important.

Creating a claim

The creation of a claim can also be considered: actions such as *creating a concept*, *creating another concept*, *inputting them in a claim via the [X,.,.] and [.,.,X] buttons*, *looking for a relation*, *not finding one*, *flipping the order of the concepts in the relation*, and finally *finding an appropriate relation type* indicate a strong user awareness of the actions she needs to perform to get a claim 'right.'

Reacting to previous annotations

Launching the history window, *consulting the concepts available* and *inputting one in one's own space* compose a final sequence showing how annotators can benefit from peers' annotations to create their own.

6.6.2 Towards interaction design patterns

These usage patterns provide the basis for interaction design patterns, '*exemplary, generalizable solutions to specific classes of design problems*' [Cooper and Reimann, 2003, chapter 7], as they give us a better understanding of not only the actions an annotator performs when having to interpret a document, 'translate' it in the ScholOnto language (i.e. break it into concepts and claims) and record this translation, but also on the articulation of these actions and how they are related to each other. This in turn gives us a better idea of how these tasks should be supported by the environment and the interface.

For instance, actions that are commonly performed together can be grouped in a specific area of the interface. Conversely, actions that are not performed together can be separated, logically and visually on the display. An improvement to ClaimSpotter that we could introduce (which we were not previously aware of) is the inclusion of a right-click menu on any highlighted element from the text [18]. The first usage pattern we have observed reveals how spotting an highlighted element and modelling it are conceptually related. In the current prototype, one has to select text, copy it, navigate to the right side of the screen, create a blank concept and paste text in the appropriate form box. A right click menu proposing an option to 'create a new concept from the selected text' would keep these conceptually-related elements altogether and would prevent the annotator from having to shift his attention to the input form. This may sound trivial, and it certainly feels like it is trivial now. However, we had not envisaged that creating concepts out of selected text would happen so often.

6.7 Discussion and lessons learned

We have reported in this chapter the findings of our evaluation study. We can conclude that a document-centric approach is suitable, as annotators have understood what has been expected of them. Based on their behaviour, it can also be concluded that incorporating the annotation and reading processes addresses the disconnection between the document and the product of the annotation, a difficulty that we have identified in ClaiMapper and ClaiMaker.

On a more specific level, each part of our evaluation study has contributed to our analysis of ClaimSpotter usability:

- We have used a statistical analysis to reveal the nature of the concepts and claims being submitted. A focus on the concepts length has confirmed the relevance of a filter spotting tentative concepts based on noun groups and proper nouns. A study of the relation types used has given us some insight on the most easily grasped ones. We have realised that further work to explain the remaining ones, possibly with some examples and training, is needed.

- Following this statistical study, a qualitative study has given us insight into the actual dialogue between annotators and ClaimSpotter. We have focused on three main

[18]Text can indeed already be selected from the document, dragged and dropped into a 'blank' concept box; it does however require navigation between the different elements of the ClaimSpotter interface. A right-click menu would make the creation of concepts easier.

themes, *annotation formalisation* (the act of breaking down an interpretation into concepts and claims and their translation into a semi-formal language), *annotation strategy* (the approach adopted by the user to make the best use of the resources at her disposal) and finally *interaction design* (interaction breakdowns.) This study has revealed what happens in 'real' tasks.

- Finally, a questionnaire, sent to the annotators after the experiment, has given us additional insight about their perception of the task they have faced. We have elicited further information about their understanding, the aspects they have liked and the ones they have not.

We have also looked at three wider issues. The first is concerned with a new information overload we may have created, when our intention has been to address an information overload. However, this overload is of a different nature. The potential overload is now caused by the multiple sources of support at one's disposal. As expert annotator a4 reports, "*the problem is 'do you make your own claims?', 'do you follow the system?', 'do you go back to the history to see what the other people have? ... There's so many different sources of getting claims.*" Additional training may help by indicating clearly what each source of information provides and what is more suited for a particular situation. The overload also indicates a need for scaffolds to guide (or constrain, depending on the situation) annotation. A set of guidelines, or a blank canvas with 'problem tackled' and 'approach proposed' slots to fill may help annotators who do not know where to start from, and may also guide annotators towards a better analysis of the contributions and connections of a document with the existing literature.

We have also discussed the possible influence which the interface may have on the way annotators create their concepts and claims. We have hypothesised that annotation may be shifted to answering a set of questions with 'yes' or 'no.'

Another important consequence of the interface is its 'current document'-centric aspect. In other words, the fact that it puts the emphasis on the current document and that it does not make it clear enough that annotating this document is part of annotating a literature in order to build a large networked structure of arguments.

Our evaluation of the current ClaimSpotter prototype has raised several questions and uncovered difficulties that could not have been anticipated without studying 'real' users performing a 'real' task. In the next chapter, we make use of this user test knowledge to motivate an additional design phase and propose improvements.

Chapter 7

Beyond ClaimSpotter

Designing, prototyping and testing ClaimSpotter has helped us identify the motivations of annotators in their literature modelling task. In this chapter, we first discuss what we believe are the strengths and weaknesses of ClaimSpotter, in order to identify research questions meriting further investigation. We then use our findings from the previous chapter to initiate a new design phase (c.f. page 8) to improve ClaimSpotter. We conclude with several deployment scenarios.

7.1 A critical view

We begin this discussion with an assessment of the strengths and weaknesses of ClaimSpotter.

7.1.1 Strengths

The following positive points have been identified from the evaluation study.

A document-centric interface, integrating reading with annotating

ClaimSpotter integrates reading with annotation. It was one of the main motivations for starting this document-centric annotation environment. Keeping 'side by side' the document and the content created by the annotator is a strength of ClaimSpotter: annotators record their concepts and claims in the system while making use of the document (if only for reading and browsing it, i.e. without benefiting from the spotting filters.)

Personalisation

The different filters are another successful point '*bringing life*' to the document and making the claim-modelling task more interactive. The document is not only a long succession of screens of text: it becomes an area where multiple parts of its content can be activated and coloured. The document becomes an artefact that can be personalised. The ClaimSpotter interface enables users to tailor the document to their expectations by highlighting elements they are interested in.

The idea of personalisation can also be found in the immediate feedback brought by ClaimSpotter, when annotators submit concepts that are copied and pasted from the text itself (this may also be a potential drawback, as we shall see shortly.) As the newly created concepts are immediately visible (not only for the current user, but also for any other user), annotators realise that what they are submitting is taken into account by the system. They make the document their own.

Filters

The information suggested by filters helps annotators. It does not matter if suggestions are less than perfect. What matters is that their presence transforms the document into a set of placeholders that annotators can use to go through its content. The danger we have noted is that at its worst, it transforms the primary open-ended task of interpreting a document (which could be formulated as '*What makes this document relevant to my research?*') into a set of '*Is this particular suggestion good or not?*' decisions, which can be answered with '*yes*' or '*no*' with less reflective work.

Claim-making support

The support brought to formulate claims by associating concepts and arranging them into claims can be considered a success too. Facilities to create claims in a playful atmosphere - by combining concepts, removing and replacing them and flipping their order - make annotation easier and more enjoyable.

Share

Every user's concepts and claims are visible in the history. They realise that their efforts are visible to their peers. It 'forces' them to express their opinions in a more careful way (although, as we shall see, this can also be a drawback.)

Approaching a document

The actual methodology implied by ScholOnto- asking annotators to think in terms of concepts, optionally enriched by concept types and connected together into claims via a relation picked from a fixed list - also helps annotators approach a document, as we have heard from several annotators in the questionnaire. This is not a proper ClaimSpotter strength, but rather a ScholOnto one. However, the very fact that ClaimSpotter keeps the document 'handy' and under the annotator's eyes makes the ScholOnto methodology more visible (the SECAI methodology (in CLARE) could be even more helpful, c.f. section 4.8.4, page 97.)

7.1.2 Weaknesses

We have also recorded the following weaknesses.

A 'current-document-centric' interface

ClaimSpotter, as it stands, makes it very difficult for an annotator to go beyond the document she is currently annotating. We only got glimpses of this problem, as the task of our user test phase was to annotate a single document. This problem may have occurred more visibly had the task been to model a collection of documents and their inter-relations.

This is however partly dependent on the ScholOnto model, which asks every concept and claim created to be grounded in a document. Annotating a document is therefore requiring the document into which concepts and claims are grounded, to be somehow 'present.'

From 'well-thought' concepts to 'visible' concepts

ClaimSpotter gives more importance to concepts that are cut and pasted (without further edition) from the content of the document than to the ones that are not edited. The former can be activated and visible in a one-click operation and are displayed as bright yellow highlights in the main document window. The latter are only visible in an external different window (the history window.)

This may have another consequence that we have already discussed: the move from 'well-thought' concepts to 'visible' concepts. ClaimSpotter should treat these two categories of concepts in a similar fashion.

Concepts and claims table

The input method used by ClaimSpotter relies on a form as in the original ClaiMaker. This creates difficulties to hold a mental map of the annotation one submits. Although the case has not occurred during the annotation itself (building a model of a single document may not be complex enough to require it), the need for a graphical representation has been pointed out by an annotator in the post-experiment questionnaire. It may be that with this representation and further time, she would have altered her representation and deleted or inputted additional concepts and claims to represent her ideas in a better way.

The form also has problems of its own, as it lists all concepts and claims in a long table that has to be scrolled back and forth when it becomes larger than the width of the screen.

Share

Annotators may want to keep a part of their interpretation at a private level and share only the elements that do not suffer controversy or the ones they are confident enough to defend. ClaimSpotter does not provide a way to hide parts of an interpretation (this functionality is however provided in ClaiMaker.)

Information overload

While our goal has clearly been to reduce the cognitive overhead brought by both the document and the translation of one's interpretation into the ScholOnto language, we may have triggered, at times, the opposite effect, as annotators may not have always known which spotting filter would give the best results [1].

Supporting annotation

If ClaimSpotter, indirectly, helps annotators to read a document, some additional support may be needed to guide them towards 'ScholOnto-compatible' statements. From what we have learned in the evaluation study, it is desirable to make annotators think more in terms of the 'relations available' and not in terms of 'the concepts they want to connect.' Approaches to support the formalisation of free text such as ontology paraphrasing may help (c.f. page 73.)

[1] Additional experience with ClaimSpotter and its filter set would obviously help matters.

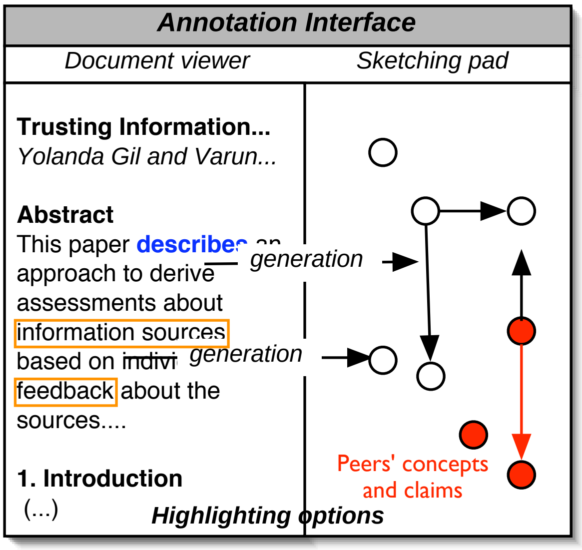


Figure 7.1: A sketching pad could replace the current input form to help annotators maintain a clearer mental model of their network of concepts and claims as they are building it.

7.2 Future research questions

The main asset we have gained with the design, development and evaluation of ClaimSpotter is this improved knowledge of the literature modelling process. Some design choices have been successful. Some have been less successful. Some have led to new difficulties that we had not planned. We present several research questions raised by the current ClaimSpotter prototype.

7.2.1 Document*s*-centric interfaces

We have purposefully used the term 'document' in its plural form in the header of this section: designing an interface accommodating different scenarios - from the annotation of a single document to the annotation of two documents in parallel (while modelling a claim involving a concept in each document, for instance) - and integrating text-based support with an efficient way to support the representation of one's interpretation forms a first future research question.

Integration of a sketching pad with the document view

The form-based input frame does not support the construction of complex models of the contributions of a document. The integration of a graphical sketching component *à la* ClaiMapper may help. Figure 7.1 gives an example sketch of an interface incorporating a document component and a sketching component.

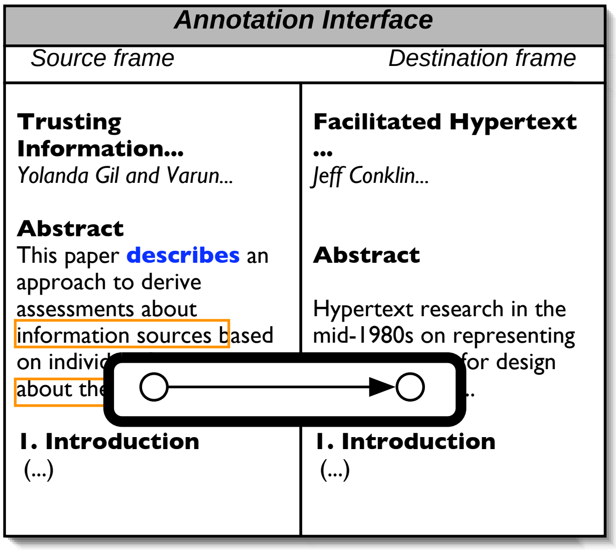


Figure 7.2: An interface displaying side by side two documents can assist the modelling of 'connections' between two documents, by letting annotators compare passages.

Spotted elements could be dragged and dropped in the sketching pad, in which they could be edited [2]. A 'graphical' history could also be proposed, representing a map of all the contributions made about this particular document and about any document citing it or being related to it.

Displaying documents side by side

While we have decided to keep the current document visible at any time (to remain compatible with the ScholOnto model), an additional interface would be needed to efficiently model connections between separate documents.

The document containing the source concept could be displayed on a left-hand side frame, and the document containing the destination concept could be displayed on a right-hand side frame. An annotator would be able to browse each document and to create several concepts for each of them. To model a connection between these two documents, a pop-up window would be raised, enabling her to create the source concept (or to select it from the ones she has created earlier) from the source document and the destination concept from the destination document. Figure 7.2 gives an overview of an interface displaying two documents side by side. Source and destination document could be flipped, as could the source and destination components of a claim.

Displaying citing documents side by side would help making sense of their contents and of the motivation underlying the citation (in the source document author's eyes.) This view

[2]Dragging and dropping text extracts can already be done in ClaiMapper, with text elements selected from an external application. There is however no possibility to filter the contents of the document.

is reminiscent of 3Book, that displays two pages extracted from different locations of a book side-by-side [Card et al., 2004].

Additional plug-ins

More robust filters could be added to the system. For instance, our rhetorical parsing filter could be replaced with the state of the art version reported in [Teufel and Moens, 2002] [3]. Another valuable addition could be a citation-parsing component, based on OpCit for instance [Hitchcock et al., 2000], to create 'blank' connections between the current document and the ones it cites. The annotator could then add her own content by modelling the reason underlying this particular citation (using the BASIS, CONTRAST and OTHER zones.)

7.2.2 Commitment

A private claim-space would enable annotators to publish the claims and concepts they believe are too provocative and/or personal to share with a wider audience. The impact of a private research space on the claim-making process would be worth investigating. Questions such as how much would get published publicly and how much would be kept private, the different relations types one would use for both public and private spaces and the possible transitions of a concept or claim from the private space to the public space after some time has passed (and presumably, some knowledge about the domain of the concept or the claim has been gained by the annotator) would be worth studying further.

7.2.3 Personalisation

Personalisation is also an important theme for ClaimSpotter, for the very task of annotating a document does require personal investment. Tailoring the environment within which the annotator expresses herself is therefore a desirable feature.

Some elements of a user profile already exist, from simple touches such as mentioning the current annotator's name on the footer of the main screen to the possibility offered to highlight only the current annotator's concepts matched in the text. Additional aspects include colouring the current annotator's concepts and claims in the History browsing window and the possibility to recall one's concepts defined for the current document in the concepts panel.

[3] Preliminary discussions have taken place for joint work.

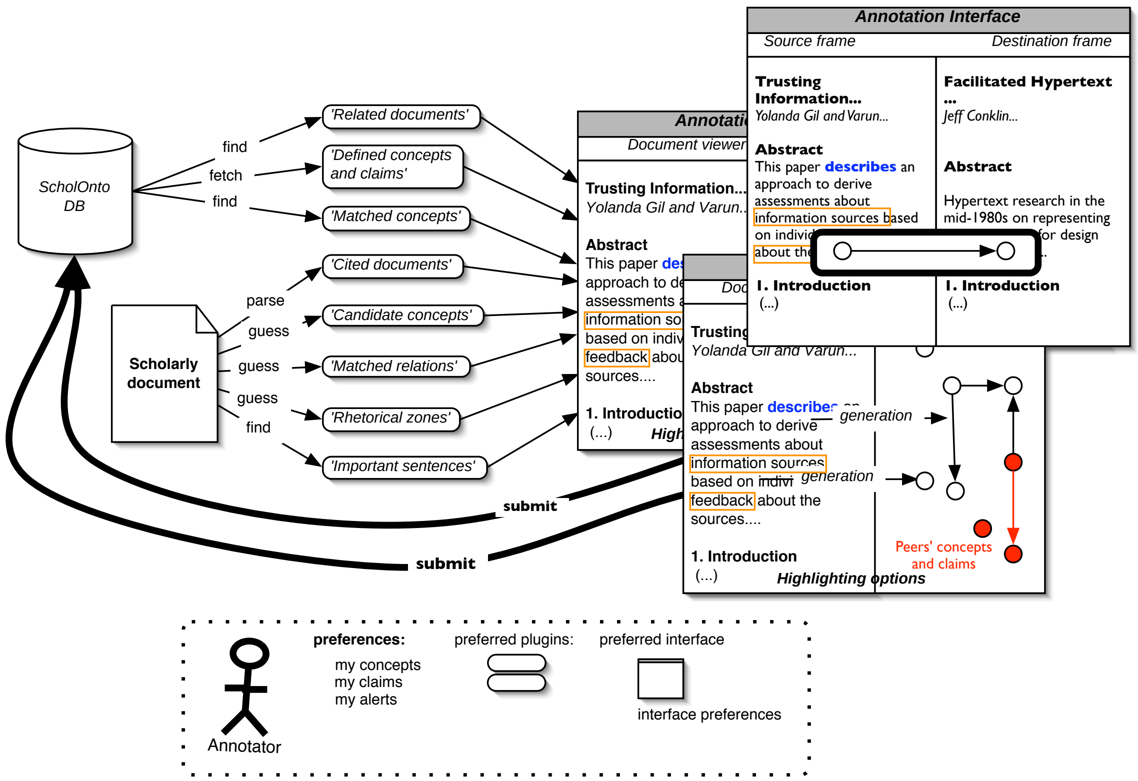


Figure 7.3: In this updated architecture (the current architecture is given in figure 5.1, page 104), an annotator could specify several preferences, including the interface she wishes to use. Such interfaces would still provide spotted elements from the document on demand.

User profiles

Extending this idea of personalisation, user profiles could also be added to ClaimSpotter. Users would specify their preferred filters, define particular filters over words or expressions and receive notifications when a matching document is submitted. Figure 7.3 gives a summary of an updated architecture for the ClaimSpotter/ScholOnto environment, in which multiple interfaces could be activated to suit an annotator's needs. Additional research would be needed to assess the impact of a set of personalisation options on the annotation.

7.2.4 Enhanced search options

As with any annotation-based approach, adding content is of little value if this content cannot be searched and retrieved. Abilities to discover a concept or a claim defined by a peer, and to reuse or combine it in a new claim triple lie at the core of ScholOnto. While ClaimSpotter provides support to search and reuse a concept of a claim, these possibilities could be enhanced.

Search facilities in ClaimSpotter

Although they have not been used very often by our annotators' pool, possibilities to search in ClaimSpotter are available. It might very well be that there was simply too much to do and that some features have been therefore not used. A more thorough evaluation, taking place over a longer period of time and giving enough time to annotators to 'digest' the features of ClaimSpotter may create a higher number of searches in the repository, especially as annotators start to connect documents with concepts they have defined elsewhere and which they want to import in the current document again.

The 'History' browser of ClaimSpotter implements a search feature with which one can look for any concept, claim or document title involving a particular term or expression. While trivial, we have reported earlier on a case in which it proved itself useful to discover what had already been said about a particular document.

Syntactic search

Another application for a search mechanism, and one that we believe may become more and more important as the repository grows, is to find out the syntactically closest concepts to the one being inputted in the input form. Let us imagine an annotator creating a concept 'ontology.' An enhanced 'search' mechanism would be able to browse through the repository and come back with not only (1) concepts in which 'ontology' is matched (for instance *discourse ontology*), but also (2) concepts which are syntactically the closest in the list of concepts (*ontologies* would be the first candidate here.) We have reported earlier a search mechanism including an algorithm to return the closest concepts to a query term, in terms of the operations needed to transform the query term in each concept. This possibility could be integrated more tightly in ClaimSpotter.

In our previous example, advising an annotator about to create a concept *ontology* that there is already a concept *ontologies* and a more precise *discourse ontology* may help. She may want to stick to her initial concept; the presentation of these alternatives may also help her characterise her idea in more precise terms.

Semantic search

An additional search option could be provided by looking at the source or destination ends of the claims involving a concept proposed in the syntactic search, reusing an approach similar to ACE [Blythe and Gil, 2004].

For an annotator typing ‘ontology’ in a new concept box, the syntactically-closest suggestions could be enhanced for instance, with concepts such as *a formal representation of the concepts of a domain*, provided that a claim {*ontologies*, ***is about***, *a formal representation of the concepts of a domain*} has been defined by a peer in the repository. This additional support may help users by proposing semantically related (for there is a semantic relation between them) concepts.

Annotation islands

In ScholOnto, the benefits of gathering all the annotations in a single place and the added value it provides, in terms for instance of related documents, comes from the fact that concepts are shared and reused by multiple annotators over multiple documents. If no concept was ever reused by anybody over any document, annotations would behave as ‘disconnected islands.’ A parallel could be drawn to a Web site that would contain no links other than links to its own pages.

However, as soon as links would be drawn to and from external Web sites (related because they share a domain of interest), this Web site would become reachable while browsing the network (assuming, for the sake of the argument, that there would be no way to type the particular URL of this site in a Web browser.) Similarly, as soon as a claim-based interpretation shares a concept with another claim and/or another document, it becomes ‘reachable’: this particular ‘annotation island’ is connected to the remainder of the network.

Encouraging an annotator to reuse existing concepts when appropriate is a way to increase the interconnectedness of the network of annotations. Of course, this has limits, and an annotator should not be forced to reuse an existing concept if she does not want to. Our objective is to prevent situations such as ‘*had I known this (nearly-identical) concept was there, I would have reused it.*’ This relates to what has been mentioned by annotator a12 about her own perception of a ‘good’ concept: “*a good concept will be something that is consistent, something that would appear again and again in the document. ‘CitiTag’ is a good concept for instance. Compared to something I would use only once.*”

The amount of reusability that we should target for concepts, and a study of the trade-off between suggesting appropriate concepts to annotators and making them feel they are forced to use an existing concept, are additional areas that would need investigation. The fundamental challenge is to leverage the power of a good knowledge representation (including notions of reusability, modularity, consistency. . .) without enforcing it and straitjacketing the user.

7.2.5 Scaffolds

Our last future research questions is related to scaffolds. We have seen in the evaluation study an interesting difference of approach to create claims. On one hand, we have had cases where annotators were thinking in advance of the relation they intended to use for their claim, knowing this particular relation was available. On the other hand, we have had cases where annotators (most frequently beginners, but experts also did this sometimes) explicitly started from the concepts they wanted to connect, without knowing which relation to use.

In the current prototype, the latter situation is not desirable because it can end (and it indeed has, in a few occasions) in the annotator having to pull back her interpretation because there is no relation matching what she intends to say [4].

Annotating a document within ScholOnto does imply, obviously, that the user subscribes to the underlying ontology, which states what ways of connecting concepts are accepted. Providing further support may nevertheless help annotators move from 'thinking about the concepts first' to 'thinking more in terms of the relations available and of the different ways at their disposal to connect these concepts.' This is not to be understood as an attempt to constrain their freedom of expression or to make them less able to express themselves (which, anyhow, has always been constrained by the ontology of relations and by the fact that their interpretation has to be broken up into triples.) This is to be understood as a mean to introduce elements of the formalism at a peaceful rhythm and to guide them towards expressing 'ScholOnto-compatible' statements more easily. Our initial experiment (c.f. chapter 3, page 29) and our literature review (c.f. section 4.3, page 52) have already shown that scaffolds can help the construction of committed literature models.

7.3 An additional design phase

We present in this section some preliminary work we have carried out in an additional design phase (c.f. figure 1.3, page 8) on this last research question.

To implement such scaffolds, we can make use of the structuration of an argument in a scholarly paper. In his study (c.f. section 4.6.5, page 83), Swales identifies three main actions carried out by authors in their introduction sections to convince their audience: the identification of a domain/field, the establishment of a niche within this domain and the occupation of this niche [Swales, 1990]. Each of these moves answers one particular question,

[4] A solution in this case is to make the 'unfeasible' triple a single concept (that would belong to the left half of the taxonomy of annotations we have presented page 21.)

q1	What is (are) the problem(s) identified in this document ?
q2	How each of these problems is related to other problem(s) ?
q3	What are the approach(es) and solutions proposed in this document ?
q4	What are the claims between the problem(s) and the solution(s) ?
q5	Which claims involving the solution(s) are defined in this document ?

Table 7.1: A questionnaire to assist annotators in their interpretation task.

similar to the ones we have used in our paper-based study in chapter 3: "*What is the field of this document?*", "*What is the problem identified in this paper?*" and "*How is the author going to address this problem?*"

Concepts are suitable objects to hold answers to these questions:

- a domain could be assigned several satisfactory types: *analysis*, *data*, *hypothesis*, or *opinion*.
- a problem could be, quite naturally, assigned a *problem* tag.
- a contribution, finally, could be assigned a *methodology* or a *solution* tag.

Relation types could also be used to connect these concepts: triples such as {*a solution*, ***addresses***, *a problem*} or {*a solution*, ***solves***, *a problem*} would be appropriate. One could also state that a *problem* ***is about*** *another problem*, a *solution* ***is related to*** *another solution*, a *problem* is a ***part of*** *another problem* and so on.

Questionnaire

A future research question is therefore to explore how the annotation process can be scaffolded with a questionnaire guiding annotators to certain elements of a scholarly document and helping them model its argument. It is also interesting to study the impact of this questionnaire, in terms of the way annotators think about their claims, and in particular if it makes them think more in terms of the relations available. This questionnaire remains at the suggestion level and annotators are free to dismiss it if they wish.

Table 7.1 lists questions which can be asked at the beginning of an annotation process. This list of questions is not revolutionary by any means and is also independent of the ScholOnto approach. In a typical 'annotation' process where we would read a document and annotate it with margin comments, answering such questions about the problem identified, the proposed solution and its connections to the other solutions proposed in the literature would be natural. It is not certain however that every participant in the evaluation study has approached her task with a similar list of questions in mind. Stating these questions explicitly can be interesting to remind them what they should focus on.

Proposing possible answers

While asking these questions can help annotators structuring their annotation by thinking first in terms of the problem, then of the approach and so on, further support can be proposed by extracting candidate answers for each of these. Candidate answers can be extracted from the areas of a text playing a particular rhetorical role [Teufel and Moens, 2002]. These areas can provide the original author's point of view. We can also look at the repository to find additional answers.

Answering the first question in table 7.1 can be supported by finding out (for the current document) all the concepts that have been assigned a type *problem*. To this list, we can add the right part of any claim using a type ***addresses*** or ***solves*** as we expect that such claims follow a canvas {*a solution*, ***addresses***, *a problem*}. Question 2 can be answered by looking at claims connecting two concepts typed as *problem*s. Question 3 can be answered by any concept typed as solution or approach, to which we can add the left part of any claim using a type ***addresses*** or ***solves***.

'Teaching' the ScholOnto language

While the above approach relies on a feature that has until now been only optional (the provision of a concept type), it does however have a positive consequence: it subtly introduces elements of the ScholOnto language into the annotation process. Structuring the approach of a document in terms of concept types, concepts and relations is a good way to make the annotators more fluent with the ScholOnto language.

It can have an additional (and beneficial) side effect, by introducing some of the more complex filters available in ClaimSpotter such as the rhetorical filter, whose function has sometimes been mistaken in the evaluation study.

Initial prototype

A scaffolding module is integrated in the latest version of ClaimSpotter (c.f. figure 7.4.)

This module (i) asks questions to shape the annotation process, and (ii) provides suggestions to answer these questions, combining the original author's stance (based on a rhetorical parsing of the document; not shown in the screenshot) and the concepts and claims of fellow annotators. For each question, elements from the database are fetched according to their type or to the way they are connected in a claim (i.e. with which relation type.) Each of these elements can then be imported and 're-appropriated' by the current annotator.

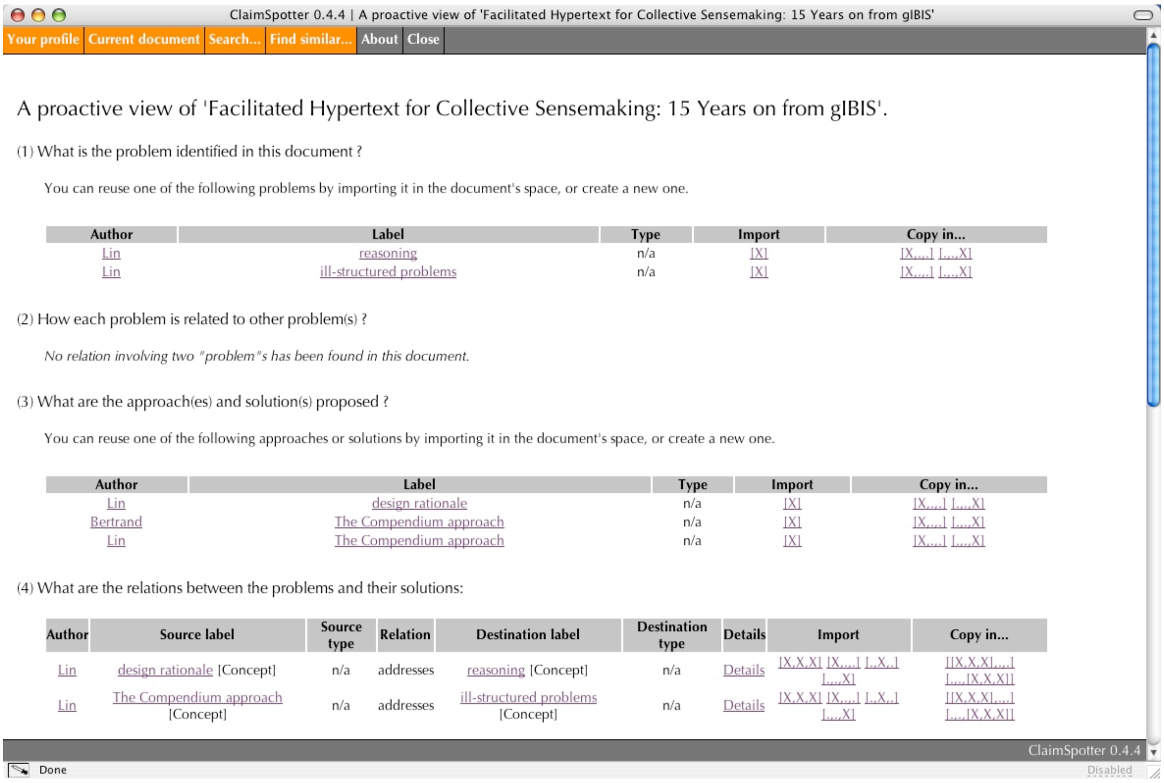


Figure 7.4: Prototype scaffolding module integrated in the last version of ClaimSpotter.

Initial evaluation

We have carried out an initial evaluation of this added source of support, by asking two participants (beginner annotator a12 and expert a4 from the main evaluation study) to assess the usefulness of the questionnaire proposed in table 7.1 and of the answers extracted automatically from the document and from the repository. We have sent a questionnaire similar to the one presented in figure 7.4 to each of them, using the document they have annotated in the experiment. Table 7.2 presents the questions we have asked them and their answers (*in italics*), with our comments and explanations.

#	Question and answers
q1	**Would the questions alone help you by providing a kind of walkthrough?** Both annotators agree it would be useful, for a similar reason: as a "*walk-through/overview (especially if I am not totally familiar with the document)*" or to "*make you think about a paper and identify its structure*."
q2	**Would you find them useful?**

#	Question and answers
	The answers highlight a difficulty with the set of questions as it is: for the first annotator, "*they would probably be confusing because I don't have a 'problem' and these questions work best with specific problems rather than just 'approach'*"; whereas for the second, it would be useful because "*I am going to be arguing that the problem/solution way of viewing a document is the way to go.*" This seems to highlight the need to devise separate sets of questions matching different kinds of information one can find in a paper. We could imagine sets of questions being designed by experts in the field, in a way similar to the set of relations that would be fine-tuned to match a community and its argumentation vocabulary.
q3	**Would the presentation of previous participants' answers be useful?** It would be useful to both, especially as "*it would aid reuse of claims, and therefore speed up the whole claim-making process.*" Both express concern however about the amount of information that could be presented ("*I wouldn't want to be flooded with EVERYONE else's answers.*") and its quality ("*I am not sure about concepts from other papers, could be chaotic.*") Filtering options would be needed to ensure annotators get as much, or as little, information as they want.
q4	**Would you find it intrusive ?** The need to make this additional source of support non-intrusive is highlighted by both annotators.
q5	**Would it prevent you from stating your own ideas?** One annotator points out that there will be times where she would "*spend time making my own claims.*" Conversely, "*there would be times where extensive reuse of claims is the best approach.*" This is again an interesting comment, for it seems to suggest two different scenarios for future users: the annotation of papers that do really matter to the users (potentially their own papers), in which cases they have enough confidence to 'translate' it in concepts and claims; and the annotation of papers which are important to a lesser degree, in which case support for a rapid annotation will be welcomed (e.g., to quickly annotate a background paper in order to reuse one of its concepts or claims.)
q6	**Would it be more useful at the beginning or at the end (of your annotation process)?**

#	Question and answers
	Flexibility is again required to call in this new source of support. For the first annotator, it is "*definitely at the end, because (my paper) might or might not work, so I would feel confused and disappointed if it brought a bunch or irrelevant stuff for my 'problem.'*" For the second annotator, it would be more "*at the beginning, as a means of getting started in the claim-making.*" She also hinted at a potential use towards the end of the process to check "*if other people interpreted the document in a similar way.*" This would be exactly the kind of commitment we would like to achieve, and indeed, this would give the extra little bit of motivation that would make annotators annotate a document. Possibilities to make your interpretation visible, to enable others to disagree and the compilation of these potentially different statements materialise our idea of a 'prototype Internet infrastructure for scholarly publishing.'
q7	**Would it help you understand what the content of an annotation should be?** We did not get any satisfying answers. Our idea was to ask if reading such questions and their answers (as concepts and claims) would help annotators understand what is expected from them (identify concepts, relate them to each other, access related documents, reuse their concepts and so on.) It was probably wrongly formulated.
q8	**Would the possibility to answer questions such as 'tell me about the documents tackling this problem' help? (by giving you leads to new papers you may not have heard of before, for instance.)** It seems to be an aspect that both annotators would appreciate. Again, it just 'clicks in' because discovering and assessing documents is part of the daily activities of a researcher (c.f. figure 4.2, page 41.) In this case, the power brought by the compilation of multiple interpretations would give something concrete to annotators, new documents they might have overseen, with a justification of why it has been proposed and by who.

Table 7.2: Answers to our initial evaluation of the usefulness of a scaffold module for ClaimSpotter.

Scaffolding the interpretation

Guiding the interpretation of a document seems to be valuable, according to our two testers' opinions. As this approach relies more on concept types, their 'correct' selection becomes more important: "*one needs to be careful with the definition of problem, it has to be a real problem, not approach or another category.*"

The usefulness of this scaffolding module (in terms of the support is provides) would be interesting to study. Additional aspects such as the best way to integrate it in the interface and the amount and quality of the answers we could propose would have to be studied too. Finally, the influence of scaffolds on annotators and on their ability to read a scholarly document and to extract its salient points would be worth assessing too.

7.4 ClaimSpotter and ScholOnto deployment

Before concluding this discussion chapter, we would like to reflect on the genericity of ClaimSpotter and to present several scenarios in which it can be deployed.

7.4.1 ScholOnto usage scenarios

We present two usage scenarios for ClaimSpotter.

The publisher's scenario

In a publishing scenario, the annotation of a document would provide a possibility, for a pool of reviewers, to discuss its contributions in a more direct way, possibly with the author participating in the construction of the network of ideas. Areas of debate could be flagged up by identifying the claims using a particular relation (***disagrees with*** would be a particularly apt relation to monitor.) This could in turn assist the revision of the document by indicating areas where the author is likely to be challenged.

The students' scenario

Students could also take part in a joint discussion about a scholarly paper. The ScholOnto language not only gives them a rich language in which they can express their claims, but it also supports the presence of contradicting claims and the reuse (for emphasis or debate) of previously submitted claims. In such a scenario, a methodology similar to the one proposed in CLARE, in which students begin with the construction of their own mental model of the document that is progressively revealed to their peers at a later stage, would be beneficial. The scaffolds identified earlier would also help students by asking them to position themselves with respect to previously submitted concepts and claims.

The difficulties we have identified earlier in this chapter (in terms of privacy and will to share only a subset of the annotation) would be important, especially in the publisher scenario. It may be that peers want to keep to themselves a part of their opinion, yet be able to record it for private use. On the other hand, asking students to voice their opinion may be a clear requirement of the exercise and may constitute a valuable outcome.

7.4.2 Beyond ScholOnto

We could also imagine ClaimSpotter being deployed with alternative formalisms.

For instance, ClaimSpotter could be applied with a tailored version of ScholOnto for a different domain. We have mentioned in the second chapter the existing possibility to replace the current relation set defined in the ontology with another set, matching the expressions used to defend an argument in another community. ClaimSpotter could obviously be updated to consider these patterns in its 'matched relations' filter. The rhetorical parser could also be tailored to different rhetorical moves: any set of roles for which discriminating features can be identified is potentially likely to be discovered by a classifier.

Large parts of ClaimSpotter could also be reused with a different formalism. We have seen earlier in this chapter how the right part of the current interface could be replaced with a sketching pad for instance; we could similarly imagine a new form designed to record different types of objects, such as traditional notes, Trellis statements or the utterances stated in a D3E Web discussion:

- A notes frame would let annotators associate notes to any part of a document and provide facilities to search these notes and connect them altogether by drawing connections between them.

- A Trellis frame would help analysts focus on the relevant part of the document (via the highlighted suggestions) and input their analysis. For instance, areas where a comparison between two alternatives is stated in a document may be highlighted by a filter and proposed.

- A 'D3E-like' frame would record a group of annotators' threaded Web discussions about not only a given section or sub-section as is currently the case, but also, at a finer level, about the particular claim defended by the author.

In the end, provided that filters extracting potentially relevant information from a document could be created, any scenario involving a text-based document and a sense-making process may be envisaged. It would not necessarily have to involve scholarly documents. Weblog entries could be considered too. With suitable filters, ClaimSpotter could assist the interpretation of a blog entry (a short document similar to a news item) spotting elements such as its most important sentences (using the words appearing in its title for instance.) An investigation of the possibilities offered by semantic blogging has been carried out in 2004-05, in parallel to this PhD research.

In all these scenarios, it would also be worthwhile to study if phenomena such as the transformation of the annotation process into a series of 'yes'/'no' questions or the increased visibility of matched 'concepts'/'notes' and its consequences would be reproduced.

7.5 Summary

We have discussed in this next to last chapter the strengths and weaknesses of the ClaimSpotter environment. These two lists, based on the feedback we have received during the evaluation study, have helped us identify future research questions.

The first of these questions is related to the 'current-document-centric' aspect of the interface that makes it difficult to model connections between different documents. We have proposed two schematic interfaces that would keep the strengths of a document-centric interface, while at the same time facilitate the creation of concepts and claims connecting different documents.

We have reflected on the notion of a private annotation space - in which annotators would be able to record concepts and claims which, for any reason, they do not wish to publish. This feature is already available in ClaiMaker and it could be integrated in ClaimSpotter. Its impact on the formalisation process would be interesting to study, especially as some aspects of the interaction we have witnessed were related to the visibility of the concepts.

The creation of a personalised environment, in which annotators could tune the functionality of the system and its presentation to make it suit their needs, has also been presented as a potential source of improvement for ClaimSpotter. This setting would be especially interesting if multiple ways to input information are provided (like for instance a text-based form, as it is the case presently or a sketch pad as in ClaiMapper.)

The final future research question we have presented is related to the scaffolding support we could bring to the annotation process. While we have decided to focus on content-based techniques to push the idea of a document-centric environment as far as possible at the time, we realise now that an additional source of support for sense-making (c.f. chapter 4, page 39) is needed. We have proposed an approach based on a questionnaire to scaffold the annotation process by helping annotators focus on several aspects of a scholarly document such as the problem it tackles and the solution it proposes (if applicable.) By subtly introducing elements of the formalism to the annotators, this approach would incidentally also makes the ScholOnto/ClaimSpotter learning curve gentler.

Chapter 8

Conclusion

We have reported our advances in tackling the challenge of designing computer-support for document annotation in the context of potentially diverse, contested views about the significance of a text. In this concluding chapter, we report on the broader impact of these advances on prototype Internet infrastructures for scholarly publishing.

ClaimSpotter provides a first approach to the semantic annotation of diverse and contested views with a semi-formal language. It supports annotation with a collection of filters that extract and emphasise suggestions from the document and from the repository of annotations. As we have seen in the literature review, there is no correct set of elements to extract that will fit everybody's needs. ClaimSpotter, instead, provides a toolbox, a suite of filters that can be combined in multiple ways to tailor the document and make it show or hide as much as needed. Collaboration is also a part of ClaimSpotter, with possibilities offered to consult - and take position with - peers' annotations. Discussion and debate are facilitated via the presentation and comparison of submitted by several researchers in a group [Buckingham Shum et al., 2005].

These different elements contribute to an infrastructure in which scholarly articles can be enriched [Sumner et al., 1998, Motta et al., 2000] to facilitate making sense of them. They contribute to the creation of a repository in which the impact of a document can be assessed. Interfaces proposing information on-demand and providing support are needed to realise prototype Internet infrastructures and to ensure the main beneficiaries invest the time and effort needed. It is our hope that the design analysis, implementation and evaluation of the ClaimSpotter prototype provides a first step in this direction.

Appendix A

Paper-based study data

This appendix contains the components of the article used by the participants to answer the questionnaire (c.f. chapter 3, page 29.)

A qX mark at the intersection of a component row (sentence or section) and an annotator column (a1 to a7) indicates that this element is picked by this annotator to answer this question. If a whole section is selected as the basis to answer one question, the qX mark is put with the corresponding section header.

S-#	Document component	a1	a2	a3	a4	a5	a6	a7
	EXTRACTING AND VISUALIZING SEMANTIC STRUCTURES IN RETRIEVAL RESULTS FOR BROWSING							q1
	Abstract							
1	The paper introduces an approach that organizes retrieval results semantically and displays them spatially for browsing.	q2	q2	q1	q2	q2 q3	q1	
2	Latent Semantic Analysis as well as cluster techniques are applied for semantic data analysis.	q2	q3	q2		q2 q3		q1
3	A modified Boltzman algorithm is used to layout documents in a two-dimensional space for interactive exploration.	q2	q3	q3		q2 q3		
4	The approach was implemented to visualize retrieval results from two different databases: the Science Citation Index Expanded and the Dido Image Bank. **Keywords**: Digital Libraries, Browsing, LSA, Conceptual Clustering, Boltzman Algorithm, Information Visualization	q3						
	Introduction							
5	The wealth of digitally stored data available today increases the demand to provide effective tools to retrieve and manage relevant data.	q1		q1	q1	q1		
6	Keyword searches over digital libraries, repositories, or the Web easily retrieve lists of several hundreds of documents.		q1		q1			
7	Information visualization - the process of analyzing and transforming data into an effective visual form - is believed to improve our interaction with large volumes of data.				q2			q1
8	First visual interfaces to digital libraries provided full-text searching and full-content retrieval capabilities and visualized documents according to authors, time, place, or citation relationships.							
9	A considerable body of recent research applies powerful mathematical techniques such as Factor Analysis, Multidimensional Scaling, or Latent Semantic Analysis to extract for example the underlying semantic structure of documents, the (evolving) specialty structure of a discipline, author co-citation patterns, changes in authors' influences in a particular field.	q3	q1 q4		q4			

S-#	Document component	a1	a2	a3	a4	a5	a6	a7
10	In order to display the results of the data analysis spatially, computationally expensive techniques have to be applied to transform data analysis results to 2 or 3-dimensional coordinates.		q4		q4			
11	The computational expense of data analysis and visualization generation is very high		q1		q4	q1	q4	
12	Therefore, precompiled, mostly static visualizations of fixed data sets are only displayed		q1		q4	q1	q4	
13	To our knowledge there exists no system that interactively visualizes retrieval results for browsing based on their underlying semantic structure.	q4	q1	q4		q1	q4	
	Data analysis						q3	
14	Latent Semantic Analysis (LSA) [4] has demonstrated improved performance over the traditional vector space techniques.			q3	q3			q2 q4
15	It overcomes the problems of synonymy (variability in human word choice) and polysemy (same word has often different meanings) by automatically organizing documents into a semantic structure more appropriate for information retrieval.							
16	We apply LSA to extract the semantic structure of a particular database in a computationally expensive batch job.		q3	q3	q2	q3		
17	At retrieval time, the result of a database query is hierarchically organized, based on the LSA output.							
18	Nearest-neighbor-based, agglomerative, hierarchical, unsupervised conceptual clustering is applied to create a hierarchy of clusters grouping documents of similar semantic structure.		q3	q2		q3	q2	
19	Clustering starts with a set of singleton clusters, each containing a single document.			q2				
20	The two clusters most similar are merged to form a new cluster that covers both.			q2				
21	This process is repeated for each of the remaining clusters.							
22	At termination, a uniform, binary hierarchy of document clusters is produced.							
23	The partition showing the highest within-cluster similarity and lowest between-cluster similarity is selected for data visualization.							
	Data visualisation						q3	

S-#	Document component	a1	a2	a3	a4	a5	a6	a7
24	Rather than being a static visualization of data, the interface is self-organizing and highly interactive.	q4			q2			
25	Data is displayed in an initially random configuration, which sorts itself out into a more-or-less acceptable display via a modified Boltzman algorithm [1].		q3	q2 q3	q3	q3	q2	q2 q4
26	The algorithm works by computing attraction and repulsion forces among nodes based on the result of the data analysis.			q2 q3	q2			
27	Nodes may represent articles or images, which are attracted to other nodes to which they have a (reference or similarity) link and repelled by nodes to which there is no link.			q2 q3				
28	If the algorithm does not produce a visually acceptable layout, or if the user wishes to view the results differently, nodes can be grabbed and moved.							
	Prototype systems							
29	Two systems have been implemented in Java using the data organization and visualization methods described above.							
30	SCI-E: The first system visualizes query results from the Science Citation Index Expanded (TM) as published by the Institute for Scientific Information							
31	The Citation Index provides access to current bibliographic information and provides access to current bibliographic information and cited references in more than 5,600 journals.							
32	Querying it via the Web of Science Interface at http://webofscience.com/ results in an often huge number of matching documents organized in lists of ten that can be marked, saved, and downloaded for detailed study.							
33	To demonstrate a visual browser to this kind of data base we will use DAIV188, a query result data set from SCI- EXPANDED that contains 188 articles matching the topic 'data AND analysis AND information AND visualization'.							
34	The articles are represented in the usual Web of Science data output format (including author(s), article title and source, cited references, addresses, abstract, language, publisher information, ISSN, document type, keywords, times cited, etc.).							

S-#	Document component	a1	a2	a3	a4	a5	a6	a7
35	LSA was applied over keywords and abstracts of articles.							
36	As a result of conceptual clustering, the 167th partition was selected for visualization							
37	It contains 20 clusters grouping 1 - 53 articles.							
38	Figure 1 shows the Java interface.							
39	Each book article is represented by a rectangle and each journal article by an oval.							
40	Articles are labeled by their first author.							
41	Lines between nodes visually represent co-citation links.							
42	The 2-dimensional layout of articles corresponds to the data mining result as well as to the forces applied by the Boltzman algorithm to generate an acceptable layout.							
43	The higher the similarity of articles within a cluster the lighter its color.							
44	Each cluster is labeled by the keyword used most often.							
45	DIDO: Another instantiation of the system enables users to browse search results from the Dido Image Bank, http://www.dlib.indiana.edu/collections/dido/ provided by the Department of the History of Art, Indiana University.							
46	Dido stores about 9,500 digitized images from the Fine Arts Slide Library collection of over 320,000 images.							
47	Each image in Dido is stored together with its thumbnail representation as well as a textual description.							
48	LSA was applied over the textual descriptions exclusively.							
49	For demonstration purposes the set of images matching the keyword descriptor ’MONET’ were retrieved and displayed for browsing.							
50	It contains 21 documents inclusive two portraits of Claude Monet drawn by Edouard Manet (see Figure 2)							
51	Thumbnail representations of images have been fetched from the Dido Database showing some of Monet’s favorite themes such as haystacks, cathedrals, and water lilies.							
	Conclusions							

S-#	Document component	a1	a2	a3	a4	a5	a6	a7
52	Initial tests show that the presented approach provides easy access to textual materials, such as articles, as well as to documents for which textual descriptions are available, such as images.							
53	Detailed user studies are in preparation.							
54	First results on using an immersive 3-dimensional CAVE environment for the interactive exploration of search results are presented in [3].							
55	An extended version of this paper as well as colored, full-size versions of Figures 1 and 2 are accessible at http://ella.slis.indiana.edu/ katy/DL00.							
	Acknowledgments							
56	Robert Goldstone, Mark Steyvers, Helen Atkins, and Eileen Fry have been valuable discussion partners.	q3		q3	q3		q3	
57	The SVDPACK [2] by M. Berry was used for computing the singular value decomposition	q3		q3	q3		q3	

Table A.1: Results from the paper-based experiment. Explanations are given in chapter 3, page 29.

Appendix B

XML documents used in and generated by ClaimSpotter

This appendix contains examples of the different documents ClaimSpotter parses (when needed to highlight a specific category of information, for instance) and generates (to use in a third-party application, such as the notification feed.) Listings are given for the document '*Trusting Information Sources One Citizen at a Time.*'

- Listing B.1: XML representation of the article. The structure of the document is preserved, including its separation in sections, subsections, ..., paragraphs and finally sentences. Each section and sentence has a unique `id` property. Additional information like the references (in sentence `S-0` for instance) is added (manually) to the text.
- Listing B.2: 'keywords' XML file.
- Listing B.3: 'candidate concepts XML file.' Noun groups are ordered first by their frequency and then alphabetically.
- Listing B.4: 'important sentences' XML file.
- Listing B.5: 'rhetorical filtering component' XML file.
- Listing B.6: this RSS feed lists the most recent contributions (concepts and claims) added to the repository for this document, in reversed chronological order.

Listing B.1: gil02trusting.xml

```
<?xml version='1.0' encoding="iso-8859-1"?>
<!DOCTYPE PAPER SYSTEM "s.dtd" >
<PAPER>
<FILENO>-</FILENO>
<TITLE>Trusting Information Sources One Citizen at a Time</TITLE>
<AUTHORS>
  <AUTHOR>Yolanda Gil</AUTHOR>
  <AUTHOR>Varun Ratnakar</AUTHOR>
</AUTHORS>
<APPEARED>-</APPEARED>
<CLASSIFICATION>-</CLASSIFICATION>
<ABSTRACT>
  <A-S ID='A-0'>This paper describes an approach to derive
   assessments about information sources based on individual
   feedback about the sources.</A-S>
...
</ABSTRACT>
<BODY>
<DIV DEPTH='1'>
<HEADER ID='H-0'>Introduction</HEADER>
<P>
<S ID='S-0'>The Semantic Web can be described as a substrate to
 support advanced functions for collaboration (human-human,
 computer-human, computer-computer), sharing of Web resources, and
  reasoning about their content <REF>[3]</REF>.</S>
<S ID='S-1'>The markup languages that are being proposed for the
 Semantic Web will be the basis to develop reasoners, proof
 checking and derivation tools, and many other functions such as
 Web services.</S>
...
</P>
...
</DIV>
<DIV>
...
</DIV>
...
</BODY>
<REFERENCES>
...
  <REFERENCE>
    <SURNAME>Berners-Lee</SURNAME>, T., <SURNAME>Hendler
    </SURNAME>, J., <SURNAME>Fensel</SURNAME>, D.: The Semantic Web
     . In: Scientific American
    78(3) (<DATE>2001</DATE>): 20-88. <REFLABEL>[3]</REFLABEL>
  </REFERENCE>
  <REFERENCE>
...
</REFERENCES>
</PAPER>
```

Listing B.2: XML keywords file for document gil02trusting.xml

```
<?xml version=’1.0’ encoding="iso-8859-1"?>
<RESULTS>
  <KEYWORD>knowledge acquisition</KEYWORD>
  <KEYWORD>task learning by instruction</KEYWORD>
  <KEYWORD>reasoning about actions</KEYWORD>
</RESULTS>
```

Listing B.3: XML candidate concepts file for document gil02trusting.xml

```
<?xml version=’1.0’ encoding="iso-8859-1"?>
<RESULTS>
 <NOUNGROUP occ=’3’>annotations</NOUNGROUP>
 <NOUNGROUP occ=’2’>decision</NOUNGROUP>
 <NOUNGROUP occ=’2’>sources</NOUNGROUP>
 <NOUNGROUP occ=’1’>alternative information sources</NOUNGROUP>
 <NOUNGROUP occ=’1’>analysis</NOUNGROUP>
 <NOUNGROUP occ=’1’>approach</NOUNGROUP>
 <NOUNGROUP occ=’1’>assessment</NOUNGROUP>
 <NOUNGROUP occ=’1’>assessments</NOUNGROUP>
 ...
 <NOUNGROUP occ=’1’>work</NOUNGROUP>
</RESULTS>
```

Listing B.4: XML important sentences file for document gil02trusting.xml

```
<?xml version=’1.0’ encoding="iso-8859-1"?>
<RESULTS>
  <SCORE id=’S-8’>7</SCORE>
  <SCORE id=’S-20’>10</SCORE>
  <SCORE id=’S-37’>10</SCORE>
  <SCORE id=’S-202’>12</SCORE>
  ...
</RESULTS>
```

Listing B.5: XML rhetorical status file for document gil02trusting.xml

```
<?xml version=’1.0’ encoding="iso-8859-1"?>
<RESULTS>
  ...
  <STATUS id=’S-25’ category=’own’ confidence=’0.799804087983’/>
  <STATUS id=’S-26’ category=’own’ confidence=’0.990823858746’/>
  <STATUS id=’S-27’ category=’own’ confidence=’0.572143435792’/>
  <STATUS id=’S-28’ category=’own’ confidence=’0.442205330657’/>
  <STATUS id=’S-29’ category=’bac’ confidence=’0.580796503459’/>
  <STATUS id=’S-30’ category=’bac’ confidence=’0.462479214876’/>
  ...
</RESULTS>
```

Listing B.6: RSS feed file for document gil02trusting.xml

```
<?xml version=‘1.0‘ encoding=‘ISO-8859-1‘?>
<rss version=‘2.0‘ xmlns:dc="http://purl.org/dc/elements/1.1"
xmlns:content="http://purl.org/rss/1.0/modules/content/">
  <channel>
    <title>ClaimSpotter feed for document ‘Trusting Information
     Sources One Citizen at a Time‘</title>
```

```
    <description>This feed contains the latest concepts and claims
     defined over this document.</description>
    <item>
      <title>measures of trust in the content of Web resources</
       title>
      <description>The concept 'measures of trust in the content of
         Web resources' (priclaimid=409) has been submitted by '
       Bertrand' on Thursday 03 February 2005 at 06:56, and is
       attached to the document 'Trusting Information Sources One
       Citizen at a Time'.</description>
    </item>
    <item>
      <title>The Semantic Web can be describedas a substrate to</
       title>
      <description>The concept 'The Semantic Web can be describedas
         a substrate to' (priclaimid=53) has been submitted by '
       Bertrand' on Thursday 17 June 2004 at 11:36, and is
       attached to the document 'Trusting Information Sources One
       Citizen at a Time'.</description>
    </item>
    <item>
      <title>a different issue on the Web of Trust regarding
       whether to trust the  content of a Web resource depending
       on its source.</title>
      <description>The concept 'a different issue on the Web of
       Trust regarding whether to trust the  content of a Web
       resource depending on its source.' (priclaimid=52) has been
         submitted by 'Bertrand' on Thursday 17 June 2004 at 11:36,
         and is attached to the document 'Trusting Information
       Sources One Citizen at a Time'.</description>
    </item>
    <item>
      <title>advanced functions for collaboration (human-human,
       computer-human, computer-computer), sharing of Web
       resources, and reasoning about their content .</title>
      <description>The concept 'advanced functions for
       collaboration (human-human, computer-human, computer-
       computer), sharing of Web resources, and reasoning about
       their content .' (priclaimid=54) has been submitted by '
       Bertrand' on Thursday 17 June 2004 at 11:36, and is
       attached to the document 'Trusting Information Sources One
       Citizen at a Time'.</description>
    </item>
    <item>
      <title>analysis</title>
      <description>The concept 'analysis' (priclaimid=50) has been
       submitted by 'Bertrand' on Thursday 17 June 2004 at 11:36,
       and is attached to the document 'Trusting Information
       Sources One Citizen at a Time'.</description>
    </item>
  </channel>
</rss>
```

Appendix C

Concepts submitted in the ClaimSpotter evaluation study

This appendix contains the concepts submitted in the ClaimSpotter evaluation study (c.f. chapter 6, page 153.)

- Table C.1, page 258, lists all the concepts submitted in alphabetical order.
- Table C.2, page 258, lists the concepts composed of three words or less.
- Table C.3, page 259, lists the concepts reused during the experiment.

a?	Concepts	Type
a4	*a Digital Object Identifier based system*	n/a
a1	*a tool that assists users with interpreting the web resource*	n/a
a4	*an almost wholly automated and highly efficient organisational framework and distribution mechanism based on the Internet , but without many of the additional services that journals*	n/a
a2	*cohse because we move away from hypermedia towards open service-based architectures*	n/a
a4	*robust services required for large-scale information environments , such as the contents of scholarly communications organised via digital libraries.*	n/a
a4	*the Open Citation (OpCit) project, which will focus on linking papers held in freely accessible eprint archives such as the Los Alamos physics archives and other distributed archives, and which will build on the work of the Open Archives initia*	n/a
a1	*? a tool that assists users with interpreting the web resources*	n/a
a13	*'literary' texts*	n/a
a11	*(more or less) to the approach adopted by START called "object-property-value"*	n/a
a10	*a community-based project that wired four computing centres (hubs) in a lower socio-economic urban area in Wellington, New Zealand's capital city*	Data
a10	*a community-based project that wired four computing centres (hubs) in a lower socio-economic urban area in Wellington, New Zealand's capital city*	n/a
a7	*a hierarchy of URIs on multiple levels*	n/a
a9	*A primary lesson from these early experiments is that the effort required to think and represent hypertextually is comparable to the development of fluency in a new language*	n/a
a5	*A set of criteria for choosing among competing solution schedules*	n/a
a2	*ability to automatically generate semantic layer*	n/a
a10	*access*	n/a
a10	*access*	Hypothesis
a4	*ACM Digital Library*	n/a
a5	*activities*	n/a
a1	*adapting lexicons and patterns to Web domains*	Methodology
a3	*analogical juxtapositions*	n/a
a3	*analogical relations*	n/a
a10	*analysis*	Analysis
a10	*analysis*	n/a
a5	*any job j1 may depend on any other job j2 while constructing a schedule*	n/a
a11	*AquaLog*	n/a
a11	*AquaLog was born under our semantic web vision*	n/a
a9	*argumentation-based DR*	n/a
a5	*assignment*	n/a
a13	*authors*	n/a
a10	*awareness*	Hypothesis
a10	*awareness*	n/a
a11	*belongs to a category*	n/a
a11	*Carniege Mellon University string distance metrics*	n/a

a?	Concepts	Type
a1	*categorises instances to ontology classes*	n/a
a3	*Christian Metz*	n/a
a3	*Christian Metz' theory*	n/a
a3	*cinematic coherence*	n/a
a3	*cinematic discourse coherence*	n/a
a3	*cinematic discourse models*	n/a
a3	*cinematic language*	Definition
a3	*cinematic rhetoric*	n/a
a3	*cinematic rhetoric technics*	n/a
a3	*cinematic screen*	n/a
a3	*cinematic shot juxtaposition*	n/a
a3	*cinematyic narrative*	n/a
a12	*CitiTag*	n/a
a9	*cognitive overhead*	n/a
a13	*cognitive strategies*	n/a
a8	*coherence*	n/a
a2	*COHSE*	n/a
a9	*collaborative hypertext*	n/a
a9	*Collective sensemaking*	n/a
a9	*Compendium*	n/a
a9	*Conceptual frameworks*	n/a
a5	*confirm its generic nature.*	n/a
a5	*constraints*	n/a
a6	*construct*	n/a
a6	*control flow constructs*	n/a
a6	*control signal*	n/a
a6	*control structure*	n/a
a4	*CrossRef*	n/a
a6	*Data-Flow*	Methodology
a6	*Data-Flow*	Problem
a6	*data-flow model needs to be enriched with some forms of control flow constructs*	Solution
a6	*data-flow model*	n/a
a6	*Data-Flow Visual Programming Language*	n/a
a9	*design rationale*	n/a
a9	*design rationale (DR)*	n/a
a6	*DFVPL*	n/a
a10	*Digital divide*	Hypothesis
a10	*Digital divide*	n/a
a7	*dimain ontology*	n/a
a7	*domain adaptation*	n/a
a1	*domain hierarchy*	n/a
a7	*domain hierarchy*	n/a
a7	*domain ontology*	n/a
a10	*don't-want-tos*	Problem
a10	*don't-want-tos*	n/a
a3	*Eisenstein's cinema*	n/a
a4	*Enabling the user to view an integrated set, or selected subset, of all archives*	n/a
a10	*encourage residents to use the hub,*	Solution

a?	Concepts	Type
a4	*Eprint archives*	n/a
a1	*espotter*	n/a
a7	*espotter*	n/a
a7	*espotter highlights documents*	n/a
a10	*explores*	Approach
a10	*explores*	n/a
a2	*feature of magpie*	n/a
a12	*feel good factor*	n/a
a7	*first, users are given the opportunity to add their own knowledge; second, users can customize ESpotter to fulfil their task at hand*	n/a
a6	*Foreach structure*	n/a
a9	*formalism*	n/a
a11	*GATE*	n/a
a3	*Grande Syntagmatique*	n/a
a9	*graphical hypertext*	n/a
a10	*have nots*	n/a
a10	*have nots*	Definition
a5	*he scheduling library proposed in this paper subscribes to the Task-Method-Domain-Application (TMDA)*	n/a
a7	*highlighting web pages*	n/a
a11	*http:__plainmoor.open.ac.uk:8080_JavaAQUAv1.0*	n/a
a9	*hybrid material*	n/a
a3	*hypertext discourse coherence*	n/a
a3	*hypertext discourse models*	n/a
a3	*hypertext language*	Definition
a3	*hypertext narrative*	n/a
a3	*hypertext node juxtaposition*	n/a
a3	*hypertext rhetoric*	n/a
a3	*hypertext screen*	n/a
a9	*Hypertext systems*	n/a
a2	*ianbility to use existing semantic annotations*	Problem
a9	*ill-structured problems*	n/a
a10	*impact of the social context*	
a10	*impact of the social context*	n/a
a12	*In the Bristol trial, the awareness of the presence of other players was correlated with how much our participants enjoyed the game as well as with how engaged they felt.*	Evidence
a2	*inability to use existing semantic annotations*	n/a
a9	*informal DR knowledge*	n/a
a4	*Information environments organised via digital libraries*	n/a
a8	*information seeking strategies*	n/a
a13	*Information-driven reading*	n/a
a3	*intellectual montage*	n/a
a2	*Internet Explorer*	n/a
a2	*interpretation and information gathering*	n/a
a2	*interpretation of web resources*	n/a
a9	*Issue Based Information System (IBIS)*	n/a
a5	*it has been validated on a number of real-life and benchmark applications*	n/a
a5	*job-depends-on*	n/a

a?	Concepts	Type
a5	*jobs*	n/a
a4	*joint NSFJISC international digital libraries programme*	n/a
a7	*KNOWITALL*	n/a
a9	*knowledge management*	n/a
a5	*knowledge modelling*	n/a
a5	*library*	n/a
a11	*Linguistic Component*	n/a
a4	*Linking*	n/a
a3	*logical relations*	n/a
a1	*Magpie*	n/a
a2	*Magpie*	n/a
a7	*magpie about recognizing entities not in the ontology*	n/a
a7	*magpie highlights documents*	n/a
a2	*magpie moves away from hypermedia towards open service-based architectures*	n/a
a9	*meeting facilitation technique*	n/a
a12	*mobile technology*	n/a
a7	*named entity recognition*	n/a
a7	*named entity recognition tool*	n/a
a8	*narrative surface*	n/a
a13	*narrator*	n/a
a9	*native hypertexts*	n/a
a2	*navigation of web resources*	Problem
a2	*navigation of web resources*	n/a
a4	*NCSTRL*	n/a
a6	*necessary programming constructs to deal with complex problems*	Problem
a8	*negotiation*	n/a
a4	*New eprint archive services*	n/a
a10	*non-users*	
a10	*non-users*	n/a
a4	*OpCit*	n/a
a4	*Open Journal project*	n/a
a13	*oral conversational stories*	
a1	*PANKOW*	n/a
a4	*Parsing the document during download to identify and read citations*	n/a
a12	*participating in a parallel virtual experience*	Approach
a3	*pattern*	n/a
a7	*Perkowitz et al.*	n/a
a12	*playground tag*	Approach
a11	*plugin mechanism*	n/a
a13	*Point-driven reading*	n/a
a13	*Point-driven understanding*	n/a
a8	*point-seeking strategies*	n/a
a11	*portable with respect to the ontology and KB*	n/a
a5	*Preferences*	n/a
a9	*premature structuring*	n/a
a12	*presence awareness of many other people*	Definition
a7	*present a tool which improves on current named entity recognition to help user browse web pages.*	n/a

a?	Concepts	Type
a7	*probability calculation given a web page and its URI*	n/a
a7	*probability estimation using Google search*	n/a
a2	*problem with magpie*	n/a
a6	*programming constructs*	n/a
a9	*question-based templates*	n/a
a11	*Questions type "who" corresponds to a person in the example ontology*	n/a
a4	*real paradigm shift taking place in scholarly communication, towards more open and accessible information*	n/a
a9	*real time capture*	n/a
a9	*reasoning*	n/a
a10	*reasons why some people choose not to compute.*	
a10	*reasons why some people choose not to compute.*	n/a
a4	*reference linking*	n/a
a11	*Relation Similarity Service*	n/a
a9	*representational formats*	n/a
a5	*requirements*	n/a
a5	*resources*	n/a
a10	*Responses*	Analysis
a10	*Responses*	n/a
a9	*reusable group memory*	n/a
a9	*reusing DR*	n/a
a4	*robust services required for large-scale information environments , such as the contents of scholarly communications organised via digital libraries.*	n/a
a5	*scheduling*	n/a
a4	*Science Citation Indexes*	n/a
a11	*Semantic Web*	n/a
a2	*semantic web browser*	n/a
a9	*Sensemaking*	n/a
a3	*sequence*	n/a
a9	*shared display*	n/a
a4	*social and business phenomena that are shaping the new information environment*	n/a
a12	*social experiences and group play*	
a12	*social experiences and group play*	n/a
a5	*solution schedule*	n/a
a12	*spontaneous social behaviours can emerge*	Hypothesis
a9	*stakeholders*	n/a
a11	*START*	n/a
a11	*START*	n/a
a13	*stories*	n/a
a13	*stories*	
a13	*Story-driven reading*	n/a
a11	*string metrics algorithms*	n/a
a5	*subscribes to*	n/a
a5	*subscribes to the Task-Method-Domain-Application (TMDA) [10] knowledge modelling framework.*	n/a
a10	*survey*	Approach
a10	*survey*	n/a

a?	Concepts	Type
a6	*synchronization*	Problem
a5	*Task-Method-Domain-Application*	n/a
a3	*text coherence*	n/a
a9	*The Compendium approach*	n/a
a5	*the desired properties of a solution schedule*	n/a
a5	*the following seven PSMs: Hill-Climbing, Propose & Backtrack (P&B), Propose & Revise (P&R), Propose & Exchange (P&E), Propose & Genetic-Exchange (P&GE), Propose & Restore-Fea*	n/a
a10	*The Fujitsu hub wiring experiment*	Model
a10	*The Fujitsu hub wiring experiment*	n/a
a4	*The Internet*	n/a
a11	*the premise that the semantic web will benefit from the availability of NL interfaces*	n/a
a4	*The process of adding citation links dynamically to documents retrieved from an archive*	n/a
a4	*The publishing industry anticipates that links on citations within scholarly papers will be one of the primary new services driving integration between scholarly sources*	n/a
a5	*The scheduling library proposed in*	n/a
a5	*the scheduling task*	n/a
a5	*the Task-Method-Domain-Application (TMDA) [10] knowledge modelling framework.*	n/a
a4	*This paper*	n/a
a10	*This paper*	n/a
a4	*This simplified form of hypertext linking*	n/a
a4	*Three principle objectives: Scale, Compatibility, Universality*	n/a
a5	*time window*	n/a
a4	*traditional content management functions of the archives*	n/a
a11	*Triple data model*	n/a
a7	*two types of behavior of entities on the Web: first, same entity means different types of things on different domains; second, some entities mostly appear on a certain domain and are not likely to appear on the other domains*	n/a
a4	*UK eLib-funded Distributed National Electronic Reserve (DNER)*	n/a
a10	*universal physical access*	Approach
a10	*universal physical access*	n/a
a3	*use of analogical relations in discourse*	n/a
a2	*use of semantic information*	Approach
a2	*use of semantic information*	n/a
a7	*user adaptation*	n/a
a12	*various emergent tactics were displayed: using gestures to attract attention from a distance, following others secretly or running, trying to surround a person in pairs, hiding and waiting for passers by, and other similar ones*	Evidence
a12	*very simple game rules based on presence states (e.g.I am Green and ‘tagged’)*	Approach
a6	*VIPERS*	Solution
a3	*visual field*	n/a
a6	*Visual Programming Language*	n/a
a6	*visual programming languages*	n/a

a?	Concepts	Type
a5	*we have proposed a generic library of PSMs*	n/a
a7	*web browsing problem*	n/a
a10	*Why, when computing is available in a socially situated, convenient environment, at no cost, do people choose not to compute?*	Problem
a9	*World Modeling Framework*	n/a

Table C.1: Concepts submitted by the annotators during the experiment.

Short concepts
access | *access* | *activities* | *analysis* | *analysis* | *AquaLog* | *assignment* | *awareness* | *awareness* | *CitiTag* | *coherence* | *COHSE* | *Compendium* | *constraints* | *construct* | *CrossRef* | *Data-Flow* | *Data-Flow* | *DFVPL* | *don't-want-tos* | *don't-want-tos* | *espotter* | *espotter* | *explores* | *explores* | *formalism* | *GATE* | *http:__plainmoor.open.ac.uk:8080_JavaAQUAv1.0* | *job-depends-on* | *jobs* | *KNOWITALL* | *library* | *Linking* | *Magpie* | *Magpie* | *NCSTRL* | *negotiation* | *non-users* | *non-users* | *OpCit* | *PANKOW* | *pattern* | *Preferences* | *reasoning* | *requirements* | *resources* | *Responses* | *Responses* | *scheduling* | *Sensemaking* | *sequence* | *stakeholders* | *START* | *START* | *survey* | *survey* | *synchronization* | *Task-Method-Domain-Application* | *VIPERS* | *analogical juxtapositions* | *analogical relations* | *argumentation-based DR* | *Christian Metz* | *cinematic coherence* | *cinematic language* | *cinematic rhetoric* | *cinematic screen* | *cinematyic narrative* | *cognitive overhead* | *collaborative hypertext* | *Collective sensemaking* | *Conceptual frameworks* | *control signal* | *control structure* | *data-flow model* | *design rationale* | *Digital divide* | *Digital divide* | *dimain ontology* | *domain adaptation* | *domain hierarchy* | *domain hierarchy* | *domain ontology* | *Eisenstein's cinema* | *Eprint archives* | *Foreach structure* | *Grande Syntagmatique* | *graphical hypertext* | *have nots* | *have nots* | *hybrid material* | *hypertext language* | *hypertext narrative* | *hypertext rhetoric* | *hypertext screen* | *Hypertext systems* | *ill-structured problems* | *intellectual montage* | *Internet Explorer* | *knowledge management* | *knowledge modelling* | *Linguistic Component* | *logical relations* | *mobile technology* | *narrative surface* | *native hypertexts* | *playground tag* | *plugin mechanism* | *point-seeking strategies* | *premature structuring* | *programming constructs* | *question-based templates* | *reference linking* | *representational formats* | *reusing DR* | *Semantic Web* | *shared display* | *solution schedule* | *subscribes to* | *text coherence* | *The Internet* | *This paper* | *This paper* | *time window* | *user adaptation* | *visual field* | *ACM Digital Library* | *Christian Metz' theory* | *cinematic discourse coherence* | *cinematic discourse models* | *cinematic rhetoric technics* | *cinematic shot juxtaposition* | *control flow constructs* | *design rationale (DR)* | *espotter highlights documents* | *feature of magpie* | *feel good factor* | *highlighting web pages* | *hypertext discourse coherence* | *hypertext discourse models* | *hypertext node juxtaposition* | *informal DR knowledge* | *information seeking strategies* | *magpie highlights documents* | *meeting facilitation technique* | *named entity recognition* | *Open Journal project* | *Perkowitz et al.* | *problem with magpie* | *real time capture* | *Relation Similarity Service* | *reusable group memory* | *Science Citation Indexes* | *semantic web browser* | *string metrics algorithms* | *The Compendium approach* | *the scheduling task* | *Triple data model* | *universal physical access* | *universal physical access* | *Visual Programming Language* | *visual programming languages* | *web browsing problem* | *World Modeling Framework*

Table C.2: Short concepts (3 words or less) submitted by the annotators during the experiment.

<table>
<tr><th>Reused concepts</th></tr>
<tr><td>a community-based project that wired four computing centres (hubs) in a lower socio-economic urban area | a research project aiming to explore the potential of spontaneous social behaviour and playful group interaction in public spaces | A set of recommendations to make the process as painless as possible | a tool that assists users with interpreting the web resources | a wireless location based multiplayer game | access | Accessing information efficiently | ACE | ACM Digital Library | Adding formalised knowledge to a document | Adding information to help sense-making | analysis | AquaLog | awareness | CitiTag | ClaiMaker | ClaimSpotter | cognitive overhead | Cognitive overload in ClaimSpotter | cognitive strategies | coherence | COHSE | Collective sensemaking | Data-Flow | data-flow model | Digital divide | Discourse ontology | Document annotation | domain hierarchy | don't-want-tos | Eprint archives | ePrint services | espotter | explores | Formalization overhead | GATE | have nots | Holding an internal model is troublesome | How people approach documents | impact of the social context | Information environments organised via digital libraries | Information-driven reading | interpretation and information gathering | Linking | Magpie | mobile technologies | navigation of web resources | non-users | OpCit | Point-driven reading | Presence awareness | reasons why some people choose not to compute. | Recognising entities likes names and organisations in a document | robust services required for large-scale information environments | ScholOnto | Semantic services | Semantic Web | Sensemaking | social experiences and group play | START | stories | Story-driven reading | subscribes to | survey | The Compendium approach | The Fujitsu hub wiring experiment | The information in there does not exist in the document | The Internet | This paper | universal physical access | use of semantic information | User studies | VIPERS</td></tr>
</table>

Table C.3: Concepts reused by the annotators.

Appendix D

Claims submitted in the ClaimSpotter evaluation study

This appendix contains the claims submitted in the ClaimSpotter evaluation study (c.f. chapter 6, page 153).

- Table D.1, page 267, lists all the claims submitted in alphabetical order (left concept first, then relation type, and finally right concept.)
- Table D.2, page 268, lists the claims reused during the experiment.

a?	Claims
a13	{*'literary' texts (type: n/a)*, ***subclass of***, *stories (type:)*}
a2	{*ability to automatically generate semantic layer (type:)*, ***example of***, *feature of magpie (type:)*}
a4	{*ACM Digital Library (type:)*, ***example of***, *Information environments organised via digital libraries (type:)*}
a1	{*adapting lexicons and patterns to Web domains (type: Methodology)*, ***uses/applies/is enabled by***, *domain hierarchy (type: n/a)*}
a3	{*analogical relations (type: n/a)*, ***is different to***, *logical relations (type: n/a)*}
a10	{*analysis (type:)*, ***is evidence for***, *impact of the social context (type:)*}
a10	{*analysis (type:)*, ***is evidence for***, *reasons why some people choose not to compute. (type:)*}
a11	{*AquaLog (type: n/a)*, ***is about***, *AquaLog was born under our semantic web vision (type: n/a)*}
a11	{*AquaLog (type: n/a)*, ***is consistent with***, *the premise that the semantic web will benefit from the availability of NL interfaces (type: n/a)*}
a11	{*AquaLog (type: n/a)*, ***uses/applies/is enabled by***, *GATE (type: n/a)*}
a9	{*argumentation-based DR (type: n/a)*, ***subclass of***, *design rationale (type: n/a)*}
a9	{*argumentation-based DR (type: n/a)*, ***uses/applies/is enabled by***, *Hypertext systems (type: n/a)*}
a5	{*assignment (type:)*, ***solves***, *activities (type:)*}
a5	{*assignment (type:)*, ***solves***, *jobs (type:)*}
a13	{*authors (type: n/a)*, ***is capable of causing***, *stories (type:)*}
a3	{*Christian Metz' theory (type: n/a)*, ***is about***, *Grande Syntagmatique (type: n/a)*}
a3	{*cinematic coherence (type: n/a)*, ***shares issues with***, *text coherence (type: n/a)*}
a3	{*cinematic discourse coherence (type: n/a)*, ***is analogous to***, *hypertext discourse coherence (type: n/a)*}
a3	{*cinematic language (type: Definition)*, ***is analogous to***, *hypertext language (type: Definition)*}
a3	{*cinematic screen (type: n/a)*, ***is analogous to***, *visual field (type: n/a)*}
a3	{*cinematyic narrative (type: n/a)*, ***shares issues with***, *hypertext narrative (type: n/a)*}
a8	{*coherence (type: n/a)*, ***is evidence for***, *point-seeking strategies (type: n/a)*}
a9	{*Collective sensemaking (type: n/a)*, ***subclass of***, *Sensemaking (type: n/a)*}
a9	{*Compendium (type: n/a)*, ***is about***, *meeting facilitation technique (type: n/a)*}
a9	{*Compendium (type: n/a)*, ***uses/applies/is enabled by***, *native hypertexts (type: n/a)*}
a5	{*constraints (type:)*, ***proves***, *solution schedule (type:)*}
a6	{*construct (type: n/a)*, ***is similar to***, *programming constructs (type: n/a)*}
a6	{*control flow constructs (type: n/a)*, ***subclass of***, *control structure (type: n/a)*}
a4	{*CrossRef (type:)*, ***uses/applies/is enabled by***, *a Digital Object Identifier based system (type:)*}
a6	{*data-flow model needs to be enriched with some forms of control flow constructs (type: Solution)*, ***addresses***, *necessary programming constructs to deal with complex problems (type: Problem)*}
a6	{*Data-Flow Visual Programming Language (type: n/a)*, ***is identical to***, *DFVPL (type: n/a)*}
a6	{*Data-Flow Visual Programming Language (type: n/a)*, ***subclass of***, *Visual Programming Language (type: n/a)*}
a9	{*design rationale (type: n/a)*, ***addresses***, *reasoning (type: n/a)*}

a?	Claims
a7	{*domain adaptation (type:)*, ***addresses***, *two types of behavior of entities on the Web: first, same entity means different types of things on different domains; second, some entities mostly appear on a certain domain and are not likely to appear on the other domains (type:)*}
a7	{*domain adaptation (type:)*, ***uses/applies/is enabled by***, *domain hierarchy (type:)*}
a7	{*domain adaptation (type:)*, ***uses/applies/is enabled by***, *domain ontology (type:)*}
a7	{*domain hierarchy (type:)*, ***is about***, *a hierarchy of URIs on multiple levels (type:)*}
a7	{*domain ontology (type:)*, ***is about***, *a hierarchy of URIs on multiple levels (type:)*}
a10	{*don't-want-tos (type:)*, ***subclass of***, *Digital divide (type:)*}
a3	{*Eisenstein's cinema (type: n/a)*, ***example of***, *use of analogical relations in discourse (type: n/a)*}
a4	{*Enabling the user to view an integrated set, or selected subset, of all archives (type:)*, ***example of***, *New eprint archive services (type:)*}
a4	{*Eprint archives (type:)*, ***is about***, *an almost wholly automated and highly efficient organisational framework and distribution mechanism based on the Internet , but without many of the additional services that journals (type:)*}
a7	{*espotter (type:)*, ***addresses***, *named entity recognition (type:)*}
a7	{*espotter (type:)*, ***improves on***, *magpie about recognizing entities not in the ontology (type:)*}
a7	{*espotter (type:)*, ***improves on***, *named entity recognition tool (type:)*}
a7	{*espotter (type:)*, ***improves on***, *PANKOW (type:)*}
a7	{*espotter (type:)*, ***is about***, *domain adaptation (type:)*}
a7	{*espotter (type:)*, ***is about***, *named entity recognition (type:)*}
a7	{*espotter (type:)*, ***is about***, *probability estimation using Google search (type:)*}
a7	{*espotter (type:)*, ***is about***, *user adaptation (type:)*}
a1	{*espotter (type: n/a)*, ***is identical to***, *a tool that assists users with interpreting the web resource (type: n/a)*}
a1	{*espotter (type: n/a)*, ***shares issues with***, *Magpie (type: n/a)*}
a1	{*espotter (type: n/a)*, ***uses/applies/is enabled by***, *adapting lexicons and patterns to Web domains (type: n/a)*}
a7	{*espotter highlights documents (type:)*, ***is similar to***, *magpie highlights documents (type:)*}
a10	{*explores (type:)*, ***is about***, *impact of the social context (type:)*}
a6	{*Foreach structure (type: n/a)*, ***subclass of***, *control structure (type: n/a)*}
a3	{*Grande Syntagmatique (type: n/a)*, ***addresses***, *cinematic discourse coherence (type: n/a)*}
a7	{*highlighting web pages (type:)*, ***solves***, *web browsing problem (type:)*}
a7	{*highlighting web pages (type:)*, ***uses/applies/is enabled by***, *named entity recognition tool (type:)*}
a3	{*hypertext discourse models (type: n/a)*, ***uses/applies/is enabled by***, *cinematic discourse models (type: n/a)*}
a3	{*hypertext language (type: Definition)*, ***shares issues with***, *cinematic language (type: Definition)*}
a3	{*hypertext node juxtaposition (type: n/a)*, ***is analogous to***, *cinematic shot juxtaposition (type: n/a)*}
a3	{*hypertext rhetoric (type: n/a)*, ***shares issues with***, *cinematic rhetoric (type: n/a)*}

a?	Claims
a3	{*hypertext rhetoric (type: n/a)*, ***uses/applies/is enabled by***, *cinematic rhetoric technics (type: n/a)*}
a3	{*hypertext screen (type: n/a)*, ***is analogous to***, *cinematic screen (type: n/a)*}
a3	{*hypertext screen (type: n/a)*, ***is analogous to***, *visual field (type: n/a)*}
a2	{*inability to use existing semantic annotations (type:)*, ***example of***, *problem with magpie (type:)*}
a9	{*informal DR knowledge (type: n/a)*, ***subclass of***, *design rationale (type: n/a)*}
a13	{*Information-driven reading (type: n/a)*, ***addresses***, *'literary' texts (type: n/a)*}
a3	{*intellectual montage (type: n/a)*, ***uses/applies/is enabled by***, *analogical juxtapositions (type: n/a)*}
a3	{*intellectual montage (type: n/a)*, ***uses/applies/is enabled by***, *analogical relations (type: n/a)*}
a12	{*In the Bristol trial, the awareness of the presence of other players was correlated with how much our participants enjoyed the game as well as with how engaged they felt. (type: Evidence)*, ***is consistent with***, ***{presence awareness of many other people, is capable of causing, feel good factor}***}
a9	{*Issue Based Information System (IBIS) (type: n/a)*, ***example of***, *Conceptual frameworks (type: n/a)*}
a9	{*Issue Based Information System (IBIS) (type: n/a)*, ***example of***, *formalism (type: n/a)*}
a5	{*it has been validated on a number of real-life and benchmark applications (type:)*, ***is evidence for***, *confirm its generic nature. (type:)*}
a5	{*job-depends-on (type:)*, ***is evidence for***, *any job j1 may depend on any other job j2 while constructing a schedule (type:)*}
a7	{*KNOWITALL (type:)*, ***is about***, *probability estimation using Google search (type:)*}
a11	{*Linguistic Component (type: n/a)*, ***part of***, *AquaLog (type: n/a)*}
a4	{*Linking (type:)*, ***part of***, *social and business phenomena that are shaping the new information environment (type:)*}
a2	{*Magpie (type:)*, ***addresses***, *interpretation of web resources (type:)*}
a2	{*Magpie (type:)*, ***example of***, *semantic web browser (type:)*}
a2	{*Magpie (type:)*, ***example of***, *use of semantic information (type:)*}
a2	{*Magpie (type:)*, ***improves on***, *COHSE (type:)*}
a2	{*Magpie (type:)*, ***improves on***, *cohse because we move away from hypermedia towards open service-based architectures (type:)*}
a2	{*Magpie (type:)*, ***is about***, *interpretation and information gathering (type:)*}
a7	{*Magpie (type:)*, ***uses/applies/is enabled by***, *dimain ontology (type:)*}
a2	{*Magpie (type:)*, ***uses/applies/is enabled by***, *Internet Explorer (type:)*}
a2	{*magpie moves away from hypermedia towards open service-based architectures (type:)*, ***is evidence for***, ***{Magpie, improves on, COHSE}***}
a12	{*mobile technology (type: n/a)*, ***improves on***, *playground tag (type: Approach)*}
a8	{*narrative surface (type: n/a)*, ***is evidence for***, *point-seeking strategies (type: n/a)*}
a13	{*narrator (type: n/a)*, ***is capable of causing***, *stories (type:)*}
a4	{*NCSTRL (type:)*, ***example of***, *Information environments organised via digital libraries (type:)*}
a8	{*negotiation (type: n/a)*, ***is evidence against***, *information seeking strategies (type: n/a)*}
a8	{*negotiation (type: n/a)*, ***is evidence for***, *point-seeking strategies (type: n/a)*}
a4	{*New eprint archive services (type:)*, ***improves on***, *traditional content management functions of the archives (type:)*}

a?	Claims
a4	{*New eprint archive services (type:)*, ***uses/applies/is enabled by***, *real paradigm shift taking place in scholarly communication, towards more open and accessible information (type:)*}
a4	{*OpCit (type:)*, ***example of***, *New eprint archive services (type:)*}
a4	{*OpCit (type:)*, ***uses/applies/is enabled by***, *joint NSFJISC international digital libraries programme (type:)*}
a4	{*Open Journal project (type:)*, ***envisages***, *OpCit (type:)*}
a13	{*oral conversational stories (type:)*, ***subclass of***, *stories (type:)*}
a1	{*PANKOW (type: n/a)*, ***is about***, *categorises instances to ontology classes (type: n/a)*}
a4	{*Parsing the document during download to identify and read citations (type:)*, , *The process of adding citation links dynamically to documents retrieved from an archive (type:)*}
a12	{*participating in a parallel virtual experience (type: Approach)*, ***uses/applies/is enabled by***, *mobile technology (type: n/a)*}
a7	{*Perkowitz et al. (type:)*, ***uses/applies/is enabled by***, *probability estimation using Google search (type:)*}
a11	{*plugin mechanism (type: n/a)*, ***part of***, *AquaLog (type: n/a)*}
a13	{*Point-driven reading (type: n/a)*, ***addresses***, *'literary' texts (type: n/a)*}
a13	{*Point-driven reading (type: n/a)*, ***is different to***, *Information-driven reading (type: n/a)*}
a13	{*Point-driven reading (type: n/a)*, ***is different to***, *Story-driven reading (type: n/a)*}
a13	{*Point-driven understanding (type: n/a)*, ***addresses***, *stories (type: n/a)*}
a13	{*Point-driven understanding (type: n/a)*, ***envisages***, *Point-driven reading (type: n/a)*}
a11	{*portable with respect to the ontology and KB (type: n/a)*, , *AquaLog (type: n/a)*}
a5	{*Preferences (type:)*, ***proves***, *A set of criteria for choosing among competing solution schedules (type:)*}
a12	{*presence awareness of many other people (type: Definition)*, ***is capable of causing***, *feel good factor (type: n/a)*}
a7	{*probability calculation given a web page and its URI (type:)*, ***uses/applies/is enabled by***, *domain hierarchy (type:)*}
a6	{*programming constructs (type: n/a)*, ***is similar to***, *control structure (type: n/a)*}
a4	{*reference linking (type:)*, ***improves on***, *This simplified form of hypertext linking (type:)*}
a4	{*reference linking (type:)*, ***is capable of causing***, *robust services required for large-scale information environments , such as the contents of scholarly communications organised via digital libraries. (type:)*}
a11	{*Relation Similarity Service (type: n/a)*, ***part of***, *AquaLog (type: n/a)*}
a5	{*requirements (type:)*, ***is evidence for***, *the desired properties of a solution schedule (type:)*}
a10	{*Responses (type:)*, ***is about***, *access (type:)*}
a10	{*Responses (type:)*, ***is about***, *awareness (type:)*}
a5	{*scheduling (type:)*, ***is about***, *assignment (type:)*}
a4	{*Science Citation Indexes (type:)*, ***uses/applies/is enabled by***, *reference linking (type:)*}
a11	{*Semantic Web (type: n/a)*, ***is evidence for***, *AquaLog (type: n/a)*}
a3	{*sequence (type: n/a)*, ***is analogous to***, *pattern (type: n/a)*}
a9	{*stakeholders (type: n/a)*, ***is capable of causing***, *ill-structured problems (type: n/a)*}

a?	Claims
a11	{*START (type: n/a)*, ***is about***, *(more or less) to the approach adopted by START called "object-property-value" (type: n/a)*}
a13	{*Story-driven reading (type: n/a)*, ***addresses***, *'literary' texts (type: n/a)*}
a11	{*string metrics algorithms (type: n/a)*, ***uses/applies/is enabled by***, *Carniege Mellon University string distance metrics (type: n/a)*}
a10	{*survey (type:)*, ***uses/applies/is enabled by***, *non-users (type:)*}
a9	{*The Compendium approach (type: n/a)*, ***addresses***, *ill-structured problems (type: n/a)*}
a9	{*The Compendium approach (type: n/a)*, ***uses/applies/is enabled by***, *hybrid material (type: n/a)*}
a9	{*The Compendium approach (type: n/a)*, ***uses/applies/is enabled by***, *real time capture (type: n/a)*}
a9	{*The Compendium approach (type: n/a)*, ***uses/applies/is enabled by***, *reusable group memory (type: n/a)*}
a9	{*The Compendium approach (type: n/a)*, ***uses/applies/is enabled by***, *shared display (type: n/a)*}
a5	{*the following seven PSMs: Hill-Climbing, Propose & Backtrack (P&B), Propose & Revise (P&R), Propose & Exchange (P&E), Propose & Genetic-Exchange (P&GE), Propose & Restore-Fea (type:)*, ***part of***, *library (type:)*}
a10	{*The Fujitsu hub wiring experiment (type:)*, ***improves on***, *access (type:)*}
a4	{*The Internet (type:)*, ***causes***, *real paradigm shift taking place in scholarly communication, towards more open and accessible information (type:)*}
a4	{*The publishing industry anticipates that links on citations within scholarly papers will be one of the primary new services driving integration between scholarly sources (type:)*, ***causes***, *OpCit (type:)*}
a5	{*The scheduling library proposed in (type:)*, ***uses/applies/is enabled by***, *the Task-Method-Domain-Application (TMDA) [10] knowledge modelling framework. (type:)*}
a10	{*This paper (type:)*, ***is about***, *a community-based project that wired four computing centres (hubs) in a lower socio-economic urban area in Wellington, New Zealand's capital city (type:)*}
a4	{*This paper (type:)*, ***is about***, *real paradigm shift taking place in scholarly communication, towards more open and accessible information (type:)*}
a4	{*This paper (type:)*, ***is about***, *the Open Citation (OpCit) project, which will focus on linking papers held in freely accessible eprint archives such as the Los Alamos physics archives and other distributed archives, and which will build on the work of the Open Archives initia (type:)*}
a4	{*This simplified form of hypertext linking (type:)*, ***impairs***, *robust services required for large-scale information environments , such as the contents of scholarly communications organised via digital libraries. (type:)*}
a4	{*Three principle objectives: Scale, Compatibility, Universality (type:)*, , *OpCit (type:)*}
a11	{*Triple data model (type: n/a)*, ***is similar to***, *(more or less) to the approach adopted by START called "object-property-value" (type: n/a)*}
a11	{*Triple data model (type: n/a)*, ***subclass of***, *belongs to a category (type: n/a)*}
a4	{*UK eLib-funded Distributed National Electronic Reserve (DNER) (type:)*, ***example of***, *Information environments organised via digital libraries (type:)*}
a10	{*universal physical access (type:)*, ***is unlikely to affect***, *Digital divide (type:)*}
a10	{*universal physical access (type:)*, ***is unlikely to affect***, *have nots (type:)*}

a?	Claims
a2	{*use of semantic information (type:)*, ***addresses***, *navigation of web resources (type:)*}
a7	{*user adaptation (type:)*, ***is about***, *first, users are given the opportunity to add their own knowledge; second, users can customize ESpotter to fulfil their task at hand (type:)*}
a12	{*various emergent tactics were displayed: using gestures to attract attention from a distance, following others secretly or running, trying to surround a person in pairs, hiding and waiting for passers by, and other similar ones (type: Evidence)*, ***is evidence for***, *spontaneous social behaviours can emerge (type: Hypothesis)*}
a12	{*very simple game rules based on presence states (e.g.I am Green and 'tagged') (type: Approach)*, ***is capable of causing***, *social experiences and group play (type: n/a)*}
a5	{*we have proposed a generic library of PSMs (type:)*, ***is evidence for***, *the scheduling task (type:)*}
a9	{*World Modeling Framework (type: n/a)*, ***example of***, *Conceptual frameworks (type: n/a)*}
a9	{*World Modeling Framework (type: n/a)*, ***example of***, *formalism (type: n/a)*}
a9	{*World Modeling Framework (type: n/a)*, ***uses/applies/is enabled by***, *question-based templates (type: n/a)*}
a3	{***{hypertext screen, is analogous to, cinematic screen}***, ***is consistent with***, ***{cinematic language, is analogous to, hypertext language}***}
a2	{***{Magpie, addresses, interpretation of web resources}***, ***improves on***, ***{Magpie, is about, interpretation and information gathering}***}

Table D.1: Claims submitted by the annotators during the experiment.

Reused claims ({*source*, ***relation***, *destination*})
{*Magpie*, ***is about***, *interpretation and information gathering*}
{*Magpie*, ***addresses***, *interpretation of web resources*}
{*Magpie*, ***improves on***, *COHSE*}
{*hypertext screen*, ***is analogous to***, *cinematic screen*}
{*cinematic language*, ***is analogous to***, *hypertext language*}
{*presence awareness of many other people*, ***is capable of causing***, *feel good factor*}

Table D.2: Claims reused by the annotators.

Appendix E

Concepts and claims submitted by each annotator in the ClaimSpotter evaluation study

This appendix contains the annotations submitted by each annotator a1 to a13 in the Claim-Spotter evaluation study (c.f. chapter 6, page 153.)

Annotator a1 (status: expert)
Document TITLE: ESpotter: Adaptive Named Entity Recognition for Web Browsing ABSTRACT: Web users are facing information overload problems, i.e., it is hard for them to find desired information on the web. Hence the growing interest in named entity recognition (NER) for discovering relevant information on users' behalf. We present a browser plug-in called ESpotter which adapts lexicons and patterns to a domain hierarchy consisting of domains on the web and user preferences for accurate and efficient NER. Mappings are created from domain independent types to domain specific types. Entities are highlighted according to their types, and users are assisted by navigational functionalities between these highlighted entities.
Tutor's concepts and claims (inputted prior to the experiment) **Concepts** *espotter (type: Solution)* *NER (type: n/a)* *Recognising entities likes names and organisations in a document (type: n/a)* **Claims** {*espotter (type:)*, ***addresses***, *Recognising entities likes names and organisations in a document (type:)*}
Experiment **Concepts** *a tool that assists users with interpreting the web resource (type: n/a)* *? a tool that assists users with interpreting the web resources (type: n/a)* *adapting lexicons and patterns to Web domains (type: Methodology)* *categorises instances to ontology classes (type: n/a)* *domain hierarchy (type: n/a)* *espotter (type: n/a)* *Magpie (type: n/a)* *PANKOW (type: n/a)* **Claims** {*PANKOW (type: n/a)*, ***is about***, *categorises instances to ontology classes (type: n/a)*} {*espotter (type: n/a)*, ***uses/applies/is enabled by***, *adapting lexicons and patterns to Web domains (type: n/a)*} {*adapting lexicons and patterns to Web domains (type: Methodology)*, ***uses/applies/is enabled by***, *domain hierarchy (type: n/a)*} {*espotter (type: n/a)*, ***is identical to***, *a tool that assists users with interpreting the web resource (type: n/a)*} {*espotter (type: n/a)*, ***shares issues with***, *Magpie (type: n/a)*}

Table E.1: Experiment data for annotator a1

Annotator a2 (status: expert)

Document

TITLE: Magpie: Browsing and Navigating on the Semantic Web

ABSTRACT: We describe several advanced functionalities of Magpie a tool that assists users with interpreting the web resources. Magpie is an extension to the Internet Explorer that automatically creates a semantic layer for web pages using a user-selected ontology. Semantic layers are annotations of a web page, with a set of applicable semantic services attached to the annotated items. We argue that the ability to generate different semantic layers for a web resource is vital to support the interpretation of web pages. Moreover, the assignment of semantic web services to the entities allows users to browse their neighbourhood semantically. At the same time, the Magpie suite offers trigger functionality based on the patterns of an automatically updated semantic log. The benefits of such an approach are illustrated by a semantically enriched browsing history management.

Tutor's concepts and claims (inputted prior to the experiment)

<u>**Concepts**</u>

? a tool that assists users with interpreting the web resources (type: n/a)
Accessing information efficiently (type: n/a)
Adding information to help sense-making (type: n/a)
interpretation and information gathering (type: n/a)
Magpie (type: n/a)

<u>**Claims**</u>

{*Magpie (type:)*, ***is about***, *interpretation and information gathering (type:)*}
{*Magpie (type:)*, ***is about***, *Adding information to help sense-making (type:)*}
{*Magpie (type:)*, ***is evidence for***, *Semantic Web (type:)*}
{*Magpie (type:)*, ***shares issues with***, *ClaimSpotter (type:)*}

Experiment

<u>**Concepts**</u>

cohse because we move away from hypermedia towards open service-based architectures (type: n/a)
ability to automatically generate semantic layer (type: n/a)
COHSE (type: n/a)
feature of magpie (type: n/a)
ianbility to use existing semantic annotations (type: Problem)
inability to use existing semantic annotations (type: n/a)
Internet Explorer (type: n/a)
interpretation and information gathering (type: n/a)
interpretation of web resources (type: n/a)
Magpie (type: n/a)
magpie moves away from hypermedia towards open service-based architectures (type: n/a)
navigation of web resources (type: Problem)
navigation of web resources (type: n/a)
problem with magpie (type: n/a)
semantic web browser (type: n/a)
use of semantic information (type: Approach)
use of semantic information (type: n/a)

<u>**Claims**</u>

{*Magpie (type:)*, ***is about***, *interpretation and information gathering (type:)*}
{*Magpie (type:)*, ***uses/applies/is enabled by***, *Internet Explorer (type:)*}
{*Magpie (type:)*, ***improves on***, *COHSE (type:)*}

{*Magpie (type:)*, ***improves on***, *cohse because we move away from hypermedia towards open service-based architectures (type:)*}
{***{Magpie, addresses, interpretation of web resources}***, ***improves on***, ***{Magpie, is about, interpretation and information gathering}***}
{*Magpie (type:)*, ***addresses***, *interpretation of web resources (type:)*}
{*use of semantic information (type:)*, ***addresses***, *navigation of web resources (type:)*}
{*magpie moves away from hypermedia towards open service-based architectures (type:)*, ***is evidence for***, ***{Magpie, improves on, COHSE}***}
{*Magpie (type:)*, ***example of***, *semantic web browser (type:)*}
{*Magpie (type:)*, ***example of***, *use of semantic information (type:)*}
{*inability to use existing semantic annotations (type:)*, ***example of***, *problem with magpie (type:)*}
{*ability to automatically generate semantic layer (type:)*, ***example of***, *feature of magpie (type:)*}

Table E.2: Experiment data for annotator a2

Annotator a3 (status: expert)

Document

TITLE: From Cinematographic to Hypertext Narrative

ABSTRACT: This paper argues that cinematographic language may provide insights into the construction of narrative coherence in hypertext. Brief examples of cinematic representation models are mapped onto the hypertext domain.

Tutor's concepts and claims (inputted prior to the experiment)

Concepts

Cinematographic language (type: n/a)
cinematographic rhetoric models (type: n/a)
coherence (type: n/a)
constructing narrative coherence in hypertext (type: n/a)
film discourse (type: n/a)
hypertext medium (type: n/a)
juxtaposition of shots (type: n/a)

Claims

{*coherence (type: n/a)*, ***uses/applies/is enabled by***, *film discourse (type: n/a)*}
{*film discourse (type: n/a)*, ***uses/applies/is enabled by***, *juxtaposition of shots (type: n/a)*}
{*Cinematographic language (type: n/a)*, ***is consistent with***, *constructing narrative coherence in hypertext (type: n/a)*}
{*cinematographic rhetoric models (type: n/a)*, ***predicts***, *hypertext medium (type: n/a)*}

Experiment

Concepts

analogical juxtapositions (type: n/a)
analogical relations (type: n/a)
Christian Metz (type: n/a)
Christian Metz' theory (type: n/a)
cinematic coherence (type: n/a)
cinematic discourse coherence (type: n/a)
cinematic discourse models (type: n/a)
cinematic language (type: Definition)
cinematic rhetoric (type: n/a)
cinematic rhetoric technics (type: n/a)
cinematic screen (type: n/a)
cinematic shot juxtaposition (type: n/a)
cinematyic narrative (type: n/a)
Eisenstein's cinema (type: n/a)
Grande Syntagmatique (type: n/a)
hypertext discourse coherence (type: n/a)
hypertext discourse models (type: n/a)
hypertext language (type: Definition)
hypertext narrative (type: n/a)
hypertext node juxtaposition (type: n/a)
hypertext rhetoric (type: n/a)
hypertext screen (type: n/a)
intellectual montage (type: n/a)
logical relations (type: n/a)
pattern (type: n/a)
sequence (type: n/a)
text coherence (type: n/a)

use of analogical relations in discourse (type: n/a)
visual field (type: n/a)

Claims

{*Christian Metz' theory (type: n/a)*, ***is about***, *Grande Syntagmatique (type: n/a)*}
{*intellectual montage (type: n/a)*, ***uses/applies/is enabled by***, *analogical relations (type: n/a)*}
{*intellectual montage (type: n/a)*, ***uses/applies/is enabled by***, *analogical juxtapositions (type: n/a)*}
{*hypertext discourse models (type: n/a)*, ***uses/applies/is enabled by***, *cinematic discourse models (type: n/a)*}
{*hypertext rhetoric (type: n/a)*, ***uses/applies/is enabled by***, *cinematic rhetoric technics (type: n/a)*}
{*Grande Syntagmatique (type: n/a)*, ***addresses***, *cinematic discourse coherence (type: n/a)*}
{***{hypertext screen, is analogous to, cinematic screen}***, ***is consistent with***, ***{cinematic language, is analogous to, hypertext language}***}
{*Eisenstein's cinema (type: n/a)*, ***example of***, *use of analogical relations in discourse (type: n/a)*}
{*analogical relations (type: n/a)*, ***is different to***, *logical relations (type: n/a)*}
{*hypertext language (type: Definition)*, ***shares issues with***, *cinematic language (type: Definition)*}
{*cinematic coherence (type: n/a)*, ***shares issues with***, *text coherence (type: n/a)*}
{*cinematyic narrative (type: n/a)*, ***shares issues with***, *hypertext narrative (type: n/a)*}
{*hypertext rhetoric (type: n/a)*, ***shares issues with***, *cinematic rhetoric (type: n/a)*}
{*cinematic discourse coherence (type: n/a)*, ***is analogous to***, *hypertext discourse coherence (type: n/a)*}
{*cinematic language (type: Definition)*, ***is analogous to***, *hypertext language (type: Definition)*}
{*hypertext node juxtaposition (type: n/a)*, ***is analogous to***, *cinematic shot juxtaposition (type: n/a)*}
{*sequence (type: n/a)*, ***is analogous to***, *pattern (type: n/a)*}
{*cinematic screen (type: n/a)*, ***is analogous to***, *visual field (type: n/a)*}
{*hypertext screen (type: n/a)*, ***is analogous to***, *visual field (type: n/a)*}
{*hypertext screen (type: n/a)*, ***is analogous to***, *cinematic screen (type: n/a)*}

Table E.3: Experiment data for annotator a3

Annotator a4 (status: expert)

Document

TITLE: Developing Services for Open Eprint Archives: Globalisation, Integration and the Impact of Links

ABSTRACT: The rapid growth of scholarly information resources available in electronic form and their organisation by digital libraries is proving fertile ground for the development of sophisticated new services, of which citation linking will be one indispensable example. Many new projects, partnerships and commercial agreements have been announced to build citation linking applications. This paper describes the Open Citation (OpCit) project, which will focus on linking papers held in freely accessible eprint archives such as the Los Alamos physics archives and other distributed archives, and which will build on the work of the Open Archives initiative to make the data held in such archives available to compliant services. The paper emphasises the work of the project in the context of emerging digital library information environments, explores how a range of new linking tools might be combined and identifies ways in which different linking applications might converge. Some early results of linked pages from the OpCit project are reported.

Tutor's concepts and claims (inputted prior to the experiment)

Concepts

a wholly automated organisational framework and distribution mechanism (type: n/a)
Access to the information (type: n/a)
Accessing information efficiently (type: Problem)
an Open Archives service being developed by the Open Citation (OpCit) project (type: n/a)
citation links uniting large, high-profile and distributed archives (type: n/a)
Eprint archives (type: n/a)
ePrint services (type: n/a)
Globalise, integrate, and assess the impace of links across documents (type: Solution)
How to make data available to Eprint-compliant services ? (type: Problem)
Information environments organised via digital libraries (type: n/a)
Intelligent services (type: Solution)
Linking (type: n/a)
OpCit (type: n/a)
OpCit (type: Solution)
Rapid growth of scholarly information resources available in electronic form and their organisation by digital libraries (type: Problem)
robust services required for large-scale information environments (type: n/a)
the implications of the new wave of eprint archives and the development of open archives (type: n/a)
This paper (type: n/a)
WWW (type: n/a)

Claims

{*This paper (type:)*, ***is about***, *the implications of the new wave of eprint archives and the development of open archives (type:)*}
{*Eprint archives (type:)*, ***is about***, *a wholly automated organisational framework and distribution mechanism (type:)*}
{*Globalise, integrate, and assess the impace of links across documents (type: Solution)*, ***addresses***, *Accessing information efficiently (type: Problem)*}
{*Information environments organised via digital libraries (type:)*, ***part of***, *WWW (type:)*}

{*Linking (type:),* ***prevents***, *robust services required for large-scale information environments (type:)*}

Experiment

Concepts

a Digital Object Identifier based system (type: n/a)
an almost wholly automated and highly efficient organisational framework and distribution mechanism based on the Internet , but without many of the additional services that journals (type: n/a)
robust services required for large-scale information environments , such as the contents of scholarly communications organised via digital libraries. (type: n/a)
the Open Citation (OpCit) project, which will focus on linking papers held in freely accessible eprint archives such as the Los Alamos physics archives and other distributed archives, and which will build on the work of the Open Archives initia (type: n/a)
ACM Digital Library (type: n/a)
CrossRef (type: n/a)
Enabling the user to view an integrated set, or selected subset, of all archives (type: n/a)
Eprint archives (type: n/a)
Information environments organised via digital libraries (type: n/a)
joint NSFJISC international digital libraries programme (type: n/a)
Linking (type: n/a)
NCSTRL (type: n/a)
New eprint archive services (type: n/a)
OpCit (type: n/a)
Open Journal project (type: n/a)
Parsing the document during download to identify and read citations (type: n/a)
real paradigm shift taking place in scholarly communication, towards more open and accessible information (type: n/a)
reference linking (type: n/a)
robust services required for large-scale information environments , such as the contents of scholarly communications organised via digital libraries. (type: n/a)
Science Citation Indexes (type: n/a)
social and business phenomena that are shaping the new information environment (type: n/a)
The Internet (type: n/a)
The process of adding citation links dynamically to documents retrieved from an archive (type: n/a)
The publishing industry anticipates that links on citations within scholarly papers will be one of the primary new services driving integration between scholarly sources (type: n/a)
This paper (type: n/a)
This simplified form of hypertext linking (type: n/a)
Three principle objectives: Scale, Compatibility, Universality (type: n/a)
traditional content management functions of the archives (type: n/a)
UK eLib-funded Distributed National Electronic Reserve (DNER) (type: n/a)

Claims

{*Three principle objectives: Scale, Compatibility, Universality (type:), , OpCit (type:)*}
{*Parsing the document during download to identify and read citations (type:), , The process of adding citation links dynamically to documents retrieved from an archive (type:)*}

{*This paper (type:)*, ***is about***, *the Open Citation (OpCit) project, which will focus on linking papers held in freely accessible eprint archives such as the Los Alamos physics archives and other distributed archives, and which will build on the work of the Open Archives initia (type:)*} {*This paper (type:)*, ***is about***, *real paradigm shift taking place in scholarly communication, towards more open and accessible information (type:)*} {*Eprint archives (type:)*, ***is about***, *an almost wholly automated and highly efficient organisational framework and distribution mechanism based on the Internet , but without many of the additional services that journals (type:)*} {*OpCit (type:)*, ***uses/applies/is enabled by***, *joint NSFJISC international digital libraries programme (type:)*} {*CrossRef (type:)*, ***uses/applies/is enabled by***, *a Digital Object Identifier based system (type:)*} {*New eprint archive services (type:)*, ***uses/applies/is enabled by***, *real paradigm shift taking place in scholarly communication, towards more open and accessible information (type:)*} {*Science Citation Indexes (type:)*, ***uses/applies/is enabled by***, *reference linking (type:)*} {*New eprint archive services (type:)*, ***improves on***, *traditional content management functions of the archives (type:)*} {*reference linking (type:)*, ***improves on***, *This simplified form of hypertext linking (type:)*} {*This simplified form of hypertext linking (type:)*, ***impairs***, *robust services required for large-scale information environments , such as the contents of scholarly communications organised via digital libraries. (type:)*} {*Linking (type:)*, ***part of***, *social and business phenomena that are shaping the new information environment (type:)*} {*OpCit (type:)*, ***example of***, *New eprint archive services (type:)*} {*Enabling the user to view an integrated set, or selected subset, of all archives (type:)*, ***example of***, *New eprint archive services (type:)*} {*UK eLib-funded Distributed National Electronic Reserve (DNER) (type:)*, ***example of***, *Information environments organised via digital libraries (type:)*} {*ACM Digital Library (type:)*, ***example of***, *Information environments organised via digital libraries (type:)*} {*NCSTRL (type:)*, ***example of***, *Information environments organised via digital libraries (type:)*} {*Open Journal project (type:)*, ***envisages***, *OpCit (type:)*} {*The publishing industry anticipates that links on citations within scholarly papers will be one of the primary new services driving integration between scholarly sources (type:)*, ***causes***, *OpCit (type:)*} {*The Internet (type:)*, ***causes***, *real paradigm shift taking place in scholarly communication, towards more open and accessible information (type:)*} {*reference linking (type:)*, ***is capable of causing***, *robust services required for large-scale information environments , such as the contents of scholarly communications organised via digital libraries. (type:)*}

Table E.4: Experiment data for annotator a4

Annotator a5 (status: beginner)

Document

TITLE: A Generic Library of Problem Solving Methods for Scheduling Applications

ABSTRACT: In this paper we describe a generic library of problem-solving methods (PSMs) for scheduling applications. Although, some attempts have been made in the past at developing libraries of scheduling methods, these only provide limited coverage: in some cases they are specific to a particular scheduling domain; in other cases they simply implement a particular scheduling technique; in other cases they fail to provide the required degree of depth and precision. Our library is based on a structured approach, whereby we first develop a scheduling task ontology, and then construct a task-specific but domain independent model of scheduling problem-solving, which generalises from specific approaches to scheduling problem-solving. Different PSMs are then constructed uniformly by specialising the generic model of scheduling problem-solving. Our library has been evaluated on a number of real-life and benchmark applications to demonstrate its generic and comprehensive nature.

Tutor's concepts and claims (inputted prior to the experiment)

Concepts

CommonKADS (type: n/a)
formalising the scheduling task (type: n/a)
ILOG (type: n/a)
the generic method ontology (type: n/a)
the task ontology (type: n/a)
the vocabulary necessary to characterise the search based problem-solving behaviour of the scheduling task. (type: n/a)

Claims

{*the generic method ontology (type:)*, ***is about***, *the vocabulary necessary to characterise the search based problem-solving behaviour of the scheduling task. (type:)*}
{*the task ontology (type:)*, ***is about***, *formalising the scheduling task (type:)*}

Experiment

Concepts

A set of criteria for choosing among competing solution schedules (type: n/a)
activities (type: n/a)
any job j1 may depend on any other job j2 while constructing a schedule (type: n/a)
assignment (type: n/a)
confirm its generic nature. (type: n/a)
constraints (type: n/a)
he scheduling library proposed in this paper subscribes to the Task-Method-Domain-Application (TMDA) (type: n/a)
it has been validated on a number of real-life and benchmark applications (type: n/a)
job-depends-on (type: n/a)
jobs (type: n/a)
knowledge modelling (type: n/a)
library (type: n/a)
Preferences (type: n/a)
requirements (type: n/a)
resources (type: n/a)
scheduling (type: n/a)
solution schedule (type: n/a)
subscribes to (type: n/a)

subscribes to the Task-Method-Domain-Application (TMDA) [10] knowledge modelling framework. (type: n/a)
Task-Method-Domain-Application (type: n/a)
the desired properties of a solution schedule (type: n/a)
the following seven PSMs: Hill-Climbing, Propose & Backtrack (P&B), Propose & Revise (P&R), Propose & Exchange (P&E), Propose & Genetic-Exchange (P&GE), Propose & Restore-Fea (type: n/a)
The scheduling library proposed in (type: n/a)
the scheduling task (type: n/a)
the Task-Method-Domain-Application (TMDA) [10] knowledge modelling framework. (type: n/a)
time window (type: n/a)
we have proposed a generic library of PSMs (type: n/a)

Claims

{*scheduling (type:)*, ***is about***, *assignment (type:)*}
{*The scheduling library proposed in (type:)*, ***uses/applies/is enabled by***, *the Task-Method-Domain-Application (TMDA) [10] knowledge modelling framework. (type:)*}
{*assignment (type:)*, ***solves***, *jobs (type:)*}
{*assignment (type:)*, ***solves***, *activities (type:)*}
{*constraints (type:)*, ***proves***, *solution schedule (type:)*}
{*Preferences (type:)*, ***proves***, *A set of criteria for choosing among competing solution schedules (type:)*}
{*requirements (type:)*, ***is evidence for***, *the desired properties of a solution schedule (type:)*}
{*job-depends-on (type:)*, ***is evidence for***, *any job j1 may depend on any other job j2 while constructing a schedule (type:)*}
{*we have proposed a generic library of PSMs (type:)*, ***is evidence for***, *the scheduling task (type:)*}
{*it has been validated on a number of real-life and benchmark applications (type:)*, ***is evidence for***, *confirm its generic nature. (type:)*}
{*the following seven PSMs: Hill-Climbing, Propose & Backtrack (P&B), Propose & Revise (P&R), Propose & Exchange (P&E), Propose & Genetic-Exchange (P&GE), Propose & Restore-Fea (type:)*, ***part of***, *library (type:)*}

Table E.5: Experiment data for annotator a5

Annotator a6 (status: beginner)

Document

TITLE: Designing new Programming Constructs in a Data Flow VL

ABSTRACT: A powerful and useful Data-Flow Visual Programming Language (DFVPL) must provide the necessary programming constructs to deal with complex problem. The main purpose of this paper is to give a contribution to the debate on DFVPL constructs, by presenting the solutions we devised for the VIPERS language.

Tutor's concepts and claims (inputted prior to the experiment)

Concepts

A data-flow model enriched with some control-flow structures (type: n/a)
A powerful and useful Data-Flow Visual Programming Language (type: n/a)
control-flow structures (type: n/a)
data-flow model (type: n/a)
The necessary programming constructs to deal with complex problem (type: n/a)
VIPERS (type: Approach)

Claims

{*VIPERS (type: n/a)*, ***is about***, *A data-flow model enriched with some control-flow structures (type: n/a)*}
{*The necessary programming constructs to deal with complex problem (type: n/a)*, ***uses/applies/is enabled by***, *A powerful and useful Data-Flow Visual Programming Language (type: n/a)*}
{*data-flow model (type: n/a)*, ***part of***, *A data-flow model enriched with some control-flow structures (type: n/a)*}
{*control-flow structures (type: n/a)*, ***part of***, *A data-flow model enriched with some control-flow structures (type: n/a)*}

Experiment

Concepts

construct (type: n/a)
control flow constructs (type: n/a)
control signal (type: n/a)
control structure (type: n/a)
Data-Flow (type: Methodology)
Data-Flow (type: Problem)
data-flow model needs to be enriched with some forms of control flow constructs (type: Solution)
data-flow model (type: n/a)
Data-Flow Visual Programming Language (type: n/a)
DFVPL (type: n/a)
Foreach structure (type: n/a)
necessary programming constructs to deal with complex problems (type: Problem)
programming constructs (type: n/a)
synchronization (type: Problem)
VIPERS (type: Solution)
Visual Programming Language (type: n/a)
visual programming languages (type: n/a)

Claims

{*data-flow model needs to be enriched with some forms of control flow constructs (type: Solution)*, ***addresses***, *necessary programming constructs to deal with complex problems (type: Problem)*}

{*Data-Flow Visual Programming Language (type: n/a)*, ***subclass of***, *Visual Programming Language (type: n/a)*}
{*Foreach structure (type: n/a)*, ***subclass of***, *control structure (type: n/a)*}
{*control flow constructs (type: n/a)*, ***subclass of***, *control structure (type: n/a)*}
{*Data-Flow Visual Programming Language (type: n/a)*, ***is identical to***, *DFVPL (type: n/a)*}
{*programming constructs (type: n/a)*, ***is similar to***, *control structure (type: n/a)*}
{*construct (type: n/a)*, ***is similar to***, *programming constructs (type: n/a)*}

Table E.6: Experiment data for annotator a6

Annotator a7 (status: beginner)

Document

TITLE: ESpotter: Adaptive Named Entity Recognition for Web Browsing

ABSTRACT: Web users are facing information overload problems, i.e., it is hard for them to find desired information on the web. Hence the growing interest in named entity recognition (NER) for discovering relevant information on users' behalf. We present a browser plug-in called ESpotter which adapts lexicons and patterns to a domain hierarchy consisting of domains on the web and user preferences for accurate and efficient NER. Mappings are created from domain independent types to domain specific types. Entities are highlighted according to their types, and users are assisted by navigational functionalities between these highlighted entities.

Tutor's concepts and claims (inputted prior to the experiment)

Concepts

espotter (type: Solution)
NER (type: n/a)
Recognising entities likes names and organisations in a document (type: n/a)

Claims

{*espotter (type:)*, ***addresses***, *Recognising entities likes names and organisations in a document (type:)*}

Experiment

Concepts

a hierarchy of URIs on multiple levels (type: n/a)
dimain ontology (type: n/a)
domain adaptation (type: n/a)
domain hierarchy (type: n/a)
domain ontology (type: n/a)
espotter (type: n/a)
espotter highlights documents (type: n/a)
first, users are given the opportunity to add their own knowledge; second, users can customize ESpotter to fulfil their task at hand (type: n/a)
highlighting web pages (type: n/a)
KNOWITALL (type: n/a)
magpie about recognizing entities not in the ontology (type: n/a)
magpie highlights documents (type: n/a)
named entity recognition (type: n/a)
named entity recognition tool (type: n/a)
Perkowitz et al. (type: n/a)
present a tool which improves on current named entity recognition to help user browse web pages. (type: n/a)
probability calculation given a web page and its URI (type: n/a)
probability estimation using Google search (type: n/a)
two types of behavior of entities on the Web: first, same entity means different types of things on different domains; second, some entities mostly appear on a certain domain and are not likely to appear on the other domains (type: n/a)
user adaptation (type: n/a)
web browsing problem (type: n/a)

Claims

{*espotter (type:)*, ***is about***, *named entity recognition (type:)*}
{*espotter (type:)*, ***is about***, *domain adaptation (type:)*}
{*espotter (type:)*, ***is about***, *user adaptation (type:)*}

{*espotter (type:)*, ***is about***, *probability estimation using Google search (type:)*} {*domain hierarchy (type:)*, ***is about***, *a hierarchy of URIs on multiple levels (type:)*} {*user adaptation (type:)*, ***is about***, *first, users are given the opportunity to add their own knowledge; second, users can customize ESpotter to fulfil their task at hand (type:)*} {*domain ontology (type:)*, ***is about***, *a hierarchy of URIs on multiple levels (type:)*} {*KNOWITALL (type:)*, ***is about***, *probability estimation using Google search (type:)*} {*Magpie (type:)*, ***uses/applies/is enabled by***, *dimain ontology (type:)*} {*highlighting web pages (type:)*, ***uses/applies/is enabled by***, *named entity recognition tool (type:)*} {*domain adaptation (type:)*, ***uses/applies/is enabled by***, *domain hierarchy (type:)*} {*domain adaptation (type:)*, ***uses/applies/is enabled by***, *domain ontology (type:)*} {*probability calculation given a web page and its URI (type:)*, ***uses/applies/is enabled by***, *domain hierarchy (type:)*} {*Perkowitz et al. (type:)*, ***uses/applies/is enabled by***, *probability estimation using Google search (type:)*} {*espotter (type:)*, ***improves on***, *PANKOW (type:)*} {*espotter (type:)*, ***improves on***, *named entity recognition tool (type:)*} {*espotter (type:)*, ***improves on***, *magpie about recognizing entities not in the ontology (type:)*} {*espotter (type:)*, ***addresses***, *named entity recognition (type:)*} {*domain adaptation (type:)*, ***addresses***, *two types of behavior of entities on the Web: first, same entity means different types of things on different domains; second, some entities mostly appear on a certain domain and are not likely to appear on the other domains (type:)*} {*highlighting web pages (type:)*, ***solves***, *web browsing problem (type:)*} {*espotter highlights documents (type:)*, ***is similar to***, *magpie highlights documents (type:)*}

Table E.7: Experiment data for annotator a7

Annotator a8 (status: beginner)
Document TITLE: Point-driven understanding: pragmatic and cognitive dimensions of literary reading ABSTRACT: Listeners generally attempt to understand oral conversational stories by figuring out what the narrator is 'getting at'; their understanding is point-driven in this sense. Analogously, a form of reading in which readers expect to be able to impute motives to authors may also be called point-driven; it is a mode that seems especially useful for reading so-called 'literary' texts. Point-driven reading is conceptually distinguishable from story-driven and information-driven types. We argue that each type is associated with a number of cognitive strategies, with point-driven reading, specifically, characterized by coherence,narrative surface, and transactional strategies. Using a modern short story, we illustrate how point-driven readings might be differentiated from other kinds. An advantage of this conceptualization is that it enables one to generate empirically testable hypotheses about literary reading; we suggest a number of such hypotheses and methods of testing them.
Tutor's concepts and claims (inputted prior to the experiment) **Concepts** *Coherence, narrative, surface, and transactional strategies (type: n/a)* *Information-driven reading (type: n/a)* *Point-driven reading (type: n/a)* *Story-driven reading (type: n/a)* *Understanding the goal of the narrator (type: n/a)* **Claims** {*Point-driven reading (type: n/a)*, ***is about***, *Information-driven reading (type: n/a)*} {*Point-driven reading (type: n/a)*, ***is about***, *Understanding the goal of the narrator (type: n/a)*} {*Point-driven reading (type: n/a)*, ***uses/applies/is enabled by***, *Coherence, narrative, surface, and transactional strategies (type: n/a)*} {*Point-driven reading (type: n/a)*, ***is different to***, *Story-driven reading (type: n/a)*}
Experiment **Concepts** *coherence (type: n/a)* *information seeking strategies (type: n/a)* *narrative surface (type: n/a)* *negotiation (type: n/a)* *point-seeking strategies (type: n/a)* **Claims** {*narrative surface (type: n/a)*, ***is evidence for***, *point-seeking strategies (type: n/a)*} {*negotiation (type: n/a)*, ***is evidence for***, *point-seeking strategies (type: n/a)*} {*coherence (type: n/a)*, ***is evidence for***, *point-seeking strategies (type: n/a)*} {*negotiation (type: n/a)*, ***is evidence against***, *information seeking strategies (type: n/a)*}

Table E.8: Experiment data for annotator a8

Annotator a9 (status: beginner)

Document

TITLE: Facilitated Hypertext for Collective Sensemaking: 15 Years on from gIBIS

ABSTRACT: Hypertext research in the mid-1980s on representing argumentation for design rationale (DR) foreshadowed what are now dominant concerns in knowledge management: representing, codifying and manipulating semiformal concepts, the use of formalisms to mediate collective sensemaking, and the construction of group memory. With the benefit of 15 years' hindsight, we can see the failure of so many hypertext DR systems to be adopted as symptomatic of the more general problem of fostering 'hypertext literacy' in real working environments. Pursuing Englebart's goal of "augmenting human intellect", we describe the Compendium approach to collective sensemaking, which demonstrates the impact that a hypertext facilitator can have on the learning and adoption problems that plagued earlier hypertext systems. We also describe how conventional documents and modelling notations can be morphed into and out of Compendium's 'native hypertext' in order to support other modes of working across diverse communities of practice.

Tutor's concepts and claims (inputted prior to the experiment)

<u>**Concepts**</u>

Collective sensemaking (type: n/a)
Sensemaking (type: n/a)
The Compendium approach (type: n/a)

<u>**Claims**</u>

{*The Compendium approach (type: n/a)*, ***is about***, *Collective sensemaking (type: n/a)*}

Experiment

<u>**Concepts**</u>

A primary lesson from these early experiments is that the effort required to think and represent hypertextually is comparable to the development of fluency in a new language (type: n/a)
argumentation-based DR (type: n/a)
cognitive overhead (type: n/a)
collaborative hypertext (type: n/a)
Collective sensemaking (type: n/a)
Compendium (type: n/a)
Conceptual frameworks (type: n/a)
design rationale (type: n/a)
design rationale (DR) (type: n/a)
formalism (type: n/a)
graphical hypertext (type: n/a)
hybrid material (type: n/a)
Hypertext systems (type: n/a)
ill-structured problems (type: n/a)
informal DR knowledge (type: n/a)
Issue Based Information System (IBIS) (type: n/a)
knowledge management (type: n/a)
meeting facilitation technique (type: n/a)
native hypertexts (type: n/a)
premature structuring (type: n/a)
question-based templates (type: n/a)
real time capture (type: n/a)
reasoning (type: n/a)

representational formats (type: n/a)
reusable group memory (type: n/a)
reusing DR (type: n/a)
Sensemaking (type: n/a)
shared display (type: n/a)
stakeholders (type: n/a)
The Compendium approach (type: n/a)
World Modeling Framework (type: n/a)

Claims

{*Compendium (type: n/a)*, ***is about***, *meeting facilitation technique (type: n/a)*}
{*The Compendium approach (type: n/a)*, ***uses/applies/is enabled by***, *real time capture (type: n/a)*}
{*The Compendium approach (type: n/a)*, ***uses/applies/is enabled by***, *hybrid material (type: n/a)*}
{*The Compendium approach (type: n/a)*, ***uses/applies/is enabled by***, *reusable group memory (type: n/a)*}
{*The Compendium approach (type: n/a)*, ***uses/applies/is enabled by***, *shared display (type: n/a)*}
{*Compendium (type: n/a)*, ***uses/applies/is enabled by***, *native hypertexts (type: n/a)*}
{*argumentation-based DR (type: n/a)*, ***uses/applies/is enabled by***, *Hypertext systems (type: n/a)*}
{*World Modeling Framework (type: n/a)*, ***uses/applies/is enabled by***, *question-based templates (type: n/a)*}
{*The Compendium approach (type: n/a)*, ***addresses***, *ill-structured problems (type: n/a)*}
{*design rationale (type: n/a)*, ***addresses***, *reasoning (type: n/a)*}
{*Issue Based Information System (IBIS) (type: n/a)*, ***example of***, *Conceptual frameworks (type: n/a)*}
{*Issue Based Information System (IBIS) (type: n/a)*, ***example of***, *formalism (type: n/a)*}
{*World Modeling Framework (type: n/a)*, ***example of***, *Conceptual frameworks (type: n/a)*}
{*World Modeling Framework (type: n/a)*, ***example of***, *formalism (type: n/a)*}
{*Collective sensemaking (type: n/a)*, ***subclass of***, *Sensemaking (type: n/a)*}
{*informal DR knowledge (type: n/a)*, ***subclass of***, *design rationale (type: n/a)*}
{*argumentation-based DR (type: n/a)*, ***subclass of***, *design rationale (type: n/a)*}
{*stakeholders (type: n/a)*, ***is capable of causing***, *ill-structured problems (type: n/a)*}

Table E.9: Experiment data for annotator a9

Annotator a10 (status: beginner)

Document

TITLE: The digital divide: Why the "don't-want-tos" won't compute: Lessons from a New Zealand ICT Project

ABSTRACT: Why, when computing is available in a socially situated, convenient environment, at no cost, do people choose not to compute? This paper describes a community-based project that wired four computing centres (hubs) in a lower socio-economic urban area in Wellington, New Zealand's capital city. One of the hubs is situated in a city council high-rise apartment block and after six months of operation it was apparent that many of the residents were not using the free computing facilities. A survey was designed and administered to the non-users in this apartment block. Responses centered on the themes of access, awareness and factors that would encourage residents to use the hub, but the majority stated they were "not interested." Analysis explores the impact of the social context within which the hub is situated and suggests reasons why some people choose not to compute.

Tutor's concepts and claims (inputted prior to the experiment)

<u>**Concepts**</u>

a community-based project that wired four computing centres (hubs) in a lower socio-economic urban area (type: n/a)
Digital divide (type: n/a)
evaluating, using multiple methodologies, a community initiative called the Smart Newtown Project. (type: Approach)
Internet (type: n/a)
providing computers to everybody (type: n/a)
the fact there will always be some people refusing to use them (type: Problem)
the findings of a non-use survey undertaken in a high-rise city council apartment block (type: n/a)
This paper (type: n/a)

<u>**Claims**</u>

{*This paper (type:),* ***is about****, evaluating, using multiple methodologies, a community initiative called the Smart Newtown Project. (type:)*}
{*providing computers to everybody (type:),* ***is unlikely to affect****, the fact there will always be some people refusing to use them (type:)*}

Experiment

<u>**Concepts**</u>

a community-based project that wired four computing centres (hubs) in a lower socio-economic urban area in Wellington, New Zealand's capital city (type: Data)
a community-based project that wired four computing centres (hubs) in a lower socio-economic urban area in Wellington, New Zealand's capital city (type: n/a)
access (type: n/a)
access (type: Hypothesis)
analysis (type: Analysis)
analysis (type: n/a)
awareness (type: Hypothesis)
awareness (type: n/a)
Digital divide (type: Hypothesis)
Digital divide (type: n/a)
don't-want-tos (type: Problem)
don't-want-tos (type: n/a)
encourage residents to use the hub, (type: Solution)

explores (type: Approach)
explores (type: n/a)
have nots (type: n/a)
have nots (type: Definition)
impact of the social context (type:)
impact of the social context (type: n/a)
non-users (type:)
non-users (type: n/a)
reasons why some people choose not to compute. (type:)
reasons why some people choose not to compute. (type: n/a)
Responses (type: Analysis)
Responses (type: n/a)
survey (type: Approach)
survey (type: n/a)
The Fujitsu hub wiring experiment (type: Model)
The Fujitsu hub wiring experiment (type: n/a)
This paper (type: n/a)
universal physical access (type: Approach)
universal physical access (type: n/a)
Why, when computing is available in a socially situated, convenient environment, at no cost, do people choose not to compute? (type: Problem)

Claims

{*This paper (type:)*, ***is about***, *a community-based project that wired four computing centres (hubs) in a lower socio-economic urban area in Wellington, New Zealand's capital city (type:)*}
{*Responses (type:)*, ***is about***, *access (type:)*}
{*Responses (type:)*, ***is about***, *awareness (type:)*}
{*explores (type:)*, ***is about***, *impact of the social context (type:)*}
{*survey (type:)*, ***uses/applies/is enabled by***, *non-users (type:)*}
{*The Fujitsu hub wiring experiment (type:)*, ***improves on***, *access (type:)*}
{*analysis (type:)*, ***is evidence for***, *impact of the social context (type:)*}
{*analysis (type:)*, ***is evidence for***, *reasons why some people choose not to compute. (type:)*}
{*don't-want-tos (type:)*, ***subclass of***, *Digital divide (type:)*}
{*universal physical access (type:)*, ***is unlikely to affect***, *Digital divide (type:)*}
{*universal physical access (type:)*, ***is unlikely to affect***, *have nots (type:)*}

Table E.10: Experiment data for annotator a10

Annotator a11 (status: beginner)
Document TITLE: Ontology-driven Question Answering in AquaLog ABSTRACT: The semantic web vision is one in which rich, ontology-based semantic markup is widely available, both to enable sophisticated interoperability among agents and to support human web users in locating and making sense of information. The availability of semantic markup on the web also opens the way to novel, sophisticated forms of question answering. While semantic information can be used in several different ways to improve question answering, an important (and fairly obvious) consequence of the availability of semantic markup on the web is that this can indeed be queried directly. Hence, in the first instance, the work on the AquaLog query answering system described in this paper is based on the premise that the semantic web will benefit from the availability of natural language query interfaces, which allow users to query semantic markup viewed as a knowledge base. AquaLog is a portable question-answering system which takes queries expressed in natural language and an ontology as input and returns answers drawn from one or more knowledge bases (KBs), which instantiate the input ontology with domain-specific information. AquaLog present an elegant solution in which different strategies are combined together. It makes use of the GATE NLP platform, string metrics algorithms, a learning mechanism as a solution to manage domain-dependent lexicon, WordNet and a novel ontology-based relation similarity service to make sense of user queries with respect to the target knowledge base. Moreover, interestingly from a research point of view, it provides a new 'twist' on the old issues associated with NLDB research. It is our view that the semantic web provides a new and potentially very important context in which results from this area of research can be applied. Finally, although AquaLog has primarily been designed for use with semantic web languages, it makes use of a generic plug-in mechanism, which means it can be easily interfaced to different ontology servers and knowledge representation platforms.
Tutor's concepts and claims (inputted prior to the experiment) **Concepts** *AquaLog (type: n/a)* *GATE (type: n/a)* *querying of resources described in machine format (type: n/a)* **Claims** {*AquaLog (type: n/a)*, ***uses/applies/is enabled by***, *GATE (type: n/a)*} {*AquaLog (type: n/a)*, ***addresses***, *querying of resources described in machine format (type: n/a)*}
Experiment **Concepts** *(more or less) to the approach adopted by START called "object-property-value" (type: n/a)* *AquaLog (type: n/a)* *AquaLog was born under our semantic web vision (type: n/a)* *belongs to a category (type: n/a)* *Carniege Mellon University string distance metrics (type: n/a)* *GATE (type: n/a)* *http:__plainmoor.open.ac.uk:8080_JavaAQUAv1.0 (type: n/a)* *Linguistic Component (type: n/a)* *plugin mechanism (type: n/a)* *portable with respect to the ontology and KB (type: n/a)* *Questions type "who" corresponds to a person in the example ontology (type: n/a)*

Relation Similarity Service (type: n/a)
Semantic Web (type: n/a)
START (type: n/a)
string metrics algorithms (type: n/a)
the premise that the semantic web will benefit from the availability of NL interfaces (type: n/a)
Triple data model (type: n/a)

Claims

{*portable with respect to the ontology and KB (type: n/a)*, , *AquaLog (type: n/a)*}
{*AquaLog (type: n/a)*, ***is about***, *AquaLog was born under our semantic web vision (type: n/a)*}
{*START (type: n/a)*, ***is about***, *(more or less) to the approach adopted by START called "object-property-value" (type: n/a)*}
{*AquaLog (type: n/a)*, ***uses/applies/is enabled by***, *GATE (type: n/a)*}
{*string metrics algorithms (type: n/a)*, ***uses/applies/is enabled by***, *Carniege Mellon University string distance metrics (type: n/a)*}
{*Semantic Web (type: n/a)*, ***is evidence for***, *AquaLog (type: n/a)*}
{*AquaLog (type: n/a)*, ***is consistent with***, *the premise that the semantic web will benefit from the availability of NL interfaces (type: n/a)*}
{*Linguistic Component (type: n/a)*, ***part of***, *AquaLog (type: n/a)*}
{*Relation Similarity Service (type: n/a)*, ***part of***, *AquaLog (type: n/a)*}
{*plugin mechanism (type: n/a)*, ***part of***, *AquaLog (type: n/a)*}
{*Triple data model (type: n/a)*, ***subclass of***, *belongs to a category (type: n/a)*}
{*Triple data model (type: n/a)*, ***is similar to***, *(more or less) to the approach adopted by START called "object-property-value" (type: n/a)*}

Table E.11: Experiment data for annotator a11

Annotator a12 (status: beginner)

Document

TITLE: Urban space as a playground for large scale group interaction: experiences with CitiTag

ABSTRACT: In this paper, I describe CitiTag, a research project aiming to explore the potential of spontaneous social behaviour and playful group interaction in public spaces through the use of mobile technologies. I discuss briefly the idea and motivating themes, the design of CitiTag, a wireless location based multiplayer game and findings from two user studies.

Tutor's concepts and claims (inputted prior to the experiment)

Concepts

a research project aiming to explore the potential of spontaneous social behaviour and playful group interaction in public spaces (type: n/a)
a research project aiming to explore the potential of spontaneous social behaviour and playful group interaction in public spaces (type: Approach)
a wireless location based multiplayer game (type: n/a)
a wireless location based multiplayer game (type: Approach)
CitiTag (type: n/a)
Group play (type: n/a)
How do people communicate ? (type: n/a)
mobile technologies (type: n/a)
mobile technologies (type: Approach)
Presence awareness (type: n/a)
Presence awareness (type: Problem)
Spontaneous behaviours (type: n/a)
User studies (type: n/a)
User studies (type: Approach)

Claims

{*CitiTag (type: n/a)*, ***is about***, *a wireless location based multiplayer game (type: n/a)*}
{*a research project aiming to explore the potential of spontaneous social behaviour and playful group interaction in public spaces (type: n/a)*, ***uses/applies/is enabled by***, *mobile technologies (type: n/a)*}
{*User studies (type: n/a)*, ***is evidence for***, *Presence awareness (type: n/a)*}
{*User studies (type: n/a)*, ***is evidence for***, *Spontaneous behaviours (type: n/a)*}

Experiment

Concepts

CitiTag (type: n/a)
feel good factor (type: n/a)
In the Bristol trial, the awareness of the presence of other players was correlated with how much our participants enjoyed the game as well as with how engaged they felt. (type: Evidence)
mobile technology (type: n/a)
participating in a parallel virtual experience (type: Approach)
playground tag (type: Approach)
presence awareness of many other people (type: Definition)
social experiences and group play (type:)
social experiences and group play (type: n/a)
spontaneous social behaviours can emerge (type: Hypothesis)

various emergent tactics were displayed: using gestures to attract attention from a distance, following others secretly or running, trying to surround a person in pairs, hiding and waiting for passers by, and other similar ones (type: Evidence)
very simple game rules based on presence states (e.g.I am Green and 'tagged') (type: Approach)
Claims
{*participating in a parallel virtual experience (type: Approach)*, ***uses/applies/is enabled by***, *mobile technology (type: n/a)*}
{*mobile technology (type: n/a)*, ***improves on***, *playground tag (type: Approach)*}
{*various emergent tactics were displayed: using gestures to attract attention from a distance, following others secretly or running, trying to surround a person in pairs, hiding and waiting for passers by, and other similar ones (type: Evidence)*, ***is evidence for***, *spontaneous social behaviours can emerge (type: Hypothesis)*}
{*In the Bristol trial, the awareness of the presence of other players was correlated with how much our participants enjoyed the game as well as with how engaged they felt. (type: Evidence)*, ***is consistent with***, ***{presence awareness of many other people, is capable of causing, feel good factor}***}
{*presence awareness of many other people (type: Definition)*, ***is capable of causing***, *feel good factor (type: n/a)*}
{*very simple game rules based on presence states (e.g.I am Green and 'tagged') (type: Approach)*, ***is capable of causing***, *social experiences and group play (type: n/a)*}

Table E.12: Experiment data for annotator a12

Annotator a13 (status: beginner)

Document

TITLE: Point-driven understanding: pragmatic and cognitive dimensions of literary reading

ABSTRACT: Listeners generally attempt to understand oral conversational stories by figuring out what the narrator is 'getting at'; their understanding is point-driven in this sense. Analogously, a form of reading in which readers expect to be able to impute motives to authors may also be called point-driven; it is a mode that seems especially useful for reading so-called 'literary' texts. Point-driven reading is conceptually distinguishable from story-driven and information-driven types. We argue that each type is associated with a number of cognitive strategies, with point-driven reading, specifically, characterized by coherence,narrative surface, and transactional strategies. Using a modern short story, we illustrate how point-driven readings might be differentiated from other kinds. An advantage of this conceptualization is that it enables one to generate empirically testable hypotheses about literary reading; we suggest a number of such hypotheses and methods of testing them.

Tutor's concepts and claims (inputted prior to the experiment)

Concepts

Coherence, narrative, surface, and transactional strategies (type: n/a)
Information-driven reading (type: n/a)
Point-driven reading (type: n/a)
Story-driven reading (type: n/a)
Understanding the goal of the narrator (type: n/a)

Claims

{*Point-driven reading (type: n/a)*, ***is about***, *Information-driven reading (type: n/a)*}
{*Point-driven reading (type: n/a)*, ***is about***, *Understanding the goal of the narrator (type: n/a)*}
{*Point-driven reading (type: n/a)*, ***uses/applies/is enabled by***, *Coherence, narrative, surface, and transactional strategies (type: n/a)*}
{*Point-driven reading (type: n/a)*, ***is different to***, *Story-driven reading (type: n/a)*}

Experiment

Concepts

'literary' texts (type: n/a)
authors (type: n/a)
cognitive strategies (type: n/a)
Information-driven reading (type: n/a)
narrator (type: n/a)
oral conversational stories (type:)
Point-driven reading (type: n/a)
Point-driven understanding (type: n/a)
stories (type: n/a)
stories (type:)
Story-driven reading (type: n/a)

Claims

{*Point-driven reading (type: n/a)*, ***addresses***, *'literary' texts (type: n/a)*}
{*Story-driven reading (type: n/a)*, ***addresses***, *'literary' texts (type: n/a)*}
{*Information-driven reading (type: n/a)*, ***addresses***, *'literary' texts (type: n/a)*}
{*Point-driven understanding (type: n/a)*, ***addresses***, *stories (type: n/a)*}
{*oral conversational stories (type:)*, ***subclass of***, *stories (type:)*}
{*'literary' texts (type: n/a)*, ***subclass of***, *stories (type:)*}

{*Point-driven reading (type: n/a)*, ***is different to***, *Story-driven reading (type: n/a)*}
{*Point-driven reading (type: n/a)*, ***is different to***, *Information-driven reading (type: n/a)*}
{*Point-driven understanding (type: n/a)*, ***envisages***, *Point-driven reading (type: n/a)*}
{*narrator (type: n/a)*, ***is capable of causing***, *stories (type:)*}
{*authors (type: n/a)*, ***is capable of causing***, *stories (type:)*}

Table E.13: Experiment data for annotator a13

Bibliography

Apple. Apple Human Interface Guidelines, 2005.

Philip Bell. Using Argument Representations to Make Thinking Visible for Individuals and Groups. In *Proceedings of the 2^{nd} International Conference on Computer Support for Collaborative Learning*, Toronto, Canada, 1997.

Tim Berners-Lee, James Hendler, and Ora Lassila. The Semantic Web. *Scientific American*, May 2001.

Daniel Billsus, David M. Hilbert, and Dan Maynes-Aminzade. Improving Proactive Information Systems. In IUI05 IUI05.

Steven Bird and Edward Loper. NLTK: The Natural Language ToolKit. In *Proceedings of the Interactive Posters and Demo Session of the 42^{nd} Annual Meeting of the Association for Computational Linguistics*, Barcelona, Spain, July 2004. Association for Computational Linguistics.

Ann Peterson Bishop. Digital Libraries and Knowledge Disaggregation: the Use of Journal Article Components. In *Proceedings of the 3^{rd} International Conference on Digital Libraries*. Association for Computing Machinery, 1998.

James Blustein. Automatically Generated Hypertext Versions of Scholarly Articles and their Evaluation. In *Proceedings of the 11^{th} ACM on Hypertext and Hypermedia*, pages 201–210, 2000.

Jim Blythe and Yolanda Gil. Incremental Formalization of Document Annotations Through Ontology-Based Paraphrasing. In *Proceedings of the 13^{th} International World Wide Web Conference (WWW2004)*, New York City, NY, USA, May 2004. Association for Computing Machinery.

Richard J. Boland and Ramkrishnan V. Tenkasi. Perspective Making and Perspective Taking in Communities of Knowing. *Organization Science*, 6:350–372, July 1995.

Alice J. Bernheim Brush, David Bargeron, Jonathan Grudin, and Anoop Gupta. Notification for Shared Annotation of Digital Documents. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI2002)*, pages 89–96, April 2002.

Simon Buckingham Shum, Allan MacLean, Victoria Bellotti, and Nick Hammond. Graphical Argumentation and Design Cognition. *Human-Computer Interaction*, 12:267–300, 1997.

Simon Buckingham Shum, Enrico Motta, and John Domingue. Representing Scholarly Claims in Internet Digital Libraries: a Knowledge Modelling Approach. In Serge Abiteboul and Anne-Marie Vercoustre, editors, *Proceedings of the 3^{rd} European Conference on Research and Advanced Technology for Digital Libraries (ECDL'99)*, Paris, France, September 1999. Springer Verlaag.

Simon Buckingham Shum, Enrico Motta, and John Domingue. ScholOnto: an Ontology-Based Digital Library Server for Research Documents and Discourse. *International Journal on Digital Libraries*, 3:237–248, September 2000.

Simon Buckingham Shum, Victoria Uren, Gangmin Li, Bertrand Sereno, and Clara Mancini. Computational Modelling of Naturalistic Argumentation in Research Literatures: Representation and Interaction Design Issues. *To appear in the International Journal of Intelligent Systems (IJIS) special issue on Computational Models of Natural Argument*, 2005.

Vannevar Bush. As We May Think. *The Atlantic Monthly*, 1945.

Stuart K. Card, Lichan Hong, Jock D. Mackinlay, and Ed H. Chi. 3Book: a Scalable 3D Virtual Book. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI2004)*, pages 1095–1098, Vienna, Austria, 2004. Association for Computing Machinery.

Jean Carletta. Assessing Agreement on Classification Tasks. the Kappa statistic. *Computational Linguistics*, 2:249–254, 1996.

Leslie Carr, Wendy Hall, Seach Bechhofer, and Carole Goble. Conceptual Linking: Ontology-Based Open Hypermedia. In WWW01 WWW01, pages 334–342.

Chaomei Chen. The Centrality of Pivotal Points in the Evolution of Scientific Networks. In IUI05 IUI05, pages 98–105.

Ed H. Chi, Lichan Hong, Michelle Gumbrecht, and Stuart K. Card. ScentHighlights: Highlighting Conceptually-Related Sentences during Reading. In IUI05 IUI05, pages 272–274.

Timothy Chkolvski, Varun Ratnakar, and Yolanda Gil. User Interfaces with Semi-Formal Representations: a Study of Desgning Argumentation Structures. In IUI05 IUI05, pages 130–137.

Fabio Ciravegna and Yorick Wilks. *Designing Adaptive Information Extraction for the Semantic Web in Amilcare*. IOS Press, 2003.

Chip Cleary and Ray Bareiss. Practical Methods for Automatically Generating Typed Links. In *Proceedings of the 7^{th} ACM Conference on Hypertext*, pages 31–41. Association for Computing Machinery, 1996.

Jeff Conklin and Michael L. Begeman. gIBIS: a Hypertext Tool for Exploratory Policy Discussions. *ACM Transactions on Office Information Systems*, 6(4):303–331, oct 1988.

Alan Cooper and Robert Reimann. *About Face 2.0: the Essentials of Interaction Design*. Wiley, 2003.

Jim Cowie and Wendy Lehnert. Information Extraction. *Communications of the ACM*, 39 (1):80–91, 1996.

Laurent Denoue and Laurence Vignollet. An Annotation Tool for Web Browsers and its Applications to Information Retrieval. In *Proceedings of the 'Recherche d'Information Assistée par Ordinateur' Conference (RIAO 2000)*, Paris, France, April 2000.

Alan Dix, Janet Finlay, Gregory D. Abowd, and Russell Beale. *Human-Computer Interaction*. Prentice Hall, 2004.

John Domingue, Martin Dzbor, and Enrico Motta. Magpie: Browsing and Navigating on the Semantic Web. In *Proceedings of the 8th International Conference on Intelligent User Interfaces (IUI04)*, pages 191–197, Funchal, Madeira, Portugal, January 2004. Association for Computing Machinery.

H. P. Edmundson. New Methods in Automatic Extracting. *Journal of the ACM*, 16:264–285, April 1969.

Doug C. Engelbart. Augmenting Human Intellect: a Conceptual Framework. Technical report, Stanford Research Institute, Menlo Park, CA, USA, October 1962.

Yolanda Gil and Varun Ratnakar. Trusting Information Sources One Citizen at a Time. In *Proceedings of the 1st International Semantic Web Conference (ISWC 2002)*, Sardinia, Italy, 2002.

Barney G. Glaser and Anselm Strauss. *Discovery of Grounded Theory. Strategies for Qualitative Research.* Sociology Press, 1967.

Jamey Graham. The Reader's Helper: a Personalized Document Reading Environment. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI99)*. Association for Computing Machinery, 1999.

Ralph Grishman. Information Extraction. In Maria Teresa Pazienza, editor, *Information Extraction: Techniques and Challenges*, Lecture Notes in Artificial Intelligence. Springer Verlag, 1997.

Thomas R. Gruber. A Translation Approach to Portable Ontology Specifications. *Knowledge Acquisition*, 5:199–220, 1993.

Frank G. Halasz, Thomas P. Moran, and Randall H. Trigg. NoteCards in a Nutshell. In *Proceedings of the Conference of Human Factors in Computing Systems*, pages 45–52, April 1987.

Siegfried Handschuh and Steffen Staab. Authoring and Annotation of Web Pages in CREAM. In *Proceedings of the 11th International World Wide Web Conference (WWW2002)*, 2002.

Marti A. Hearst. Multi-Paragraph Segmentation of Expository Texts. In *Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics*, 1994.

Marti A. Hearst. TileBars: Visualisation of Term Distribution Information in Full Text Information Access. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI95)*, Denver, CO, USA, May 1995. Association for Computing Machinery.

Steve Hitchcock, Les Carr, Zhuoan Jiao, Donna Bergmark, Wendy Hall, Carl Lagoze, and Steven Harnad. Developing Services for Open Eprint Archives: Globalisation, Integration and the Impact of Links. In *Proceedings of the 5th International Conference on Digital Libraries*. Association for Computing Machinery, 2000.

Christopher M. Hoadley and Marcia C. Linn. Teaching Science Through Online, Peer Discussions: SpeakEasy in the Knowledge Integration Process. *International Journal of Science Education*, 22:839–857, 2000.

Robert E. Horn. *Visualizing Argumentation: Software Tools for Collaborative and Educational Sense-Making*, chapter Infrastructure for Navigating Interdisciplinary Debates: Critical Decisions for Representing Argumentation, pages 165–184. Springer Verlag, 2003.

Ken Hyland. Persuasion and Context: the Pragmatics of Academic Metadiscourse. *Journal of Pragmatics*, 30:437–455, 1998.

IUI05. *Proceedings of the 2005 International Conference on Intelligent User Interfaces (IUI05)*, San Diego, CA, USA, January 2005.

José Kahan, Marja Riitta Koivunen, Eric Prud'Hommeaux, and Ralph R. Swick. Annotea: an Open RDF Infrastructure for Shared Web Annotations. In WWW01 WWW01.

Simon Kampa. *Who Are the Experts? e-Scholars in the Semantic Web*. PhD thesis, Department of Electronics and Computer Science, University of Southampton, Southampton, UK, October 2002.

David R. Karger, Boris Katz, Jimmy Lin, and Dennis Quan. Sticky Notes for the Semantic Web. In *Proceedings of the Poster Session of the 7^{th} International Conference on Intelligent User Interfaces (IUI03)*, pages 254–256, Miami, FL, USA, January 2003. Association for Computing Machinery.

Judith Klavans and Min-Yen Kan. Role of Verbs in Document Analysis. In *Proceedings of the COLING-ACL 1998 Conference*, pages 680–686, Montreal, Canada, 1998. Association for Computational Linguistics.

Werner Kunz and Horst Rittel. Issues as Elements of Information Systems. Technical Report Working Paper 131, Institute of Urban & Regional Development, Berkeley, University of California, CA, USA, 1970.

Julian Kupiec, Jan Pedersen, and Francine Chen. A Trainable Document Summarizer. In *Proceedings of the ACM SIGIR'95 conference*, pages 68–73. Association for Computing Machinery, 1995.

Steve Lawrence, C. Lee Giles, and Kurt Bollacker. Digital Libraries and Autonomous Citation Indexing. *IEEE Computer*, 32(6):67–71, 1999.

David B. Leake, Ana Maguitman, Thomas Reichherzer, Alberto Cañas, Marco Carvalho, Marco Arguedas, Sofia Brenes, and Tom Eskridge. Aiding Knowledge Capture by Searching for Extensions of Knowledge Models. In *Proceedings of the International Conference On Knowledge Capture (KCAP03)*, Sanibel, FL, USA, October 2003. Association for Computing Machinery.

John D. Lowrance, Ian W. Harrison, and Andres C. Rodriguez. Capturing Analytic Thought. In *Proceedings of the International Conference On Knowledge Capture (KCAP01)*, Victoria, BC, USA, October 2001. Association for Computing Machinery.

Hans Peter Luhn. The Automatic Creation of Literature Abstracts. *IBM Journal of Research Development*, 2:159–165, 1959.

William C. Mann and Sandra A. Thompson. Rhetorical Structure Theory: Toward a Functional Theory of Text Organization. *Text*, 8:243–281, 1987.

Chris Manning and Hinrich Schütze. *Foundations of Statistical Natural Language Processing*. MIT Press, May 1999.

Daniel Marcu. The Rhetorical Parsing of Natural Language Texts. In *Proceedings of the 35th Annual Meeting of the ACL*, pages 96–103. Association for Computational Linguistics, 1997.

Catherine C. Marshall. Annotation: from Paper Books to the Digital Library. In *Proceedings of the 2nd ACM International Conference on Digital Libraries*, pages 131–140, Philadelphia, PA, USA, 1997. Association for Computing Machinery.

Seiji Miike, Etsuo Itoh, Kenji Ono, and Kazuo Sumita. A Full-Text Retrieval System with a Dynamic Abstract Generation Function. In *Proceedings of the 17th Annual International ACM-SIGIR Conference*, pages 152–161, 1994.

George A. Miller. WordNet: a Lexical Database for English. *Communications of the ACM*, 38:39–41, November 1995.

Enrico Motta, Simon Buckingham Shum, and John Domingue. Case Studies in Ontology-Driven Document Enrichment: Principles, Tools and Applications. *International Journal on Human Computer Studies*, 52:1071–1109, 2000.

Hidetsugu Nanba and Manabu Okumura. Towards Multi-Paper Summarization using Reference Information. In *Proceedings of the IJCAI'99 Conference*, pages 926–931, 1999.

Mary S. Neff and James W. Cooper. ASHRAM: Active Summarization and Markup. In *Proceedings of Hawaii International Conference on System Sciences (HICSS-32)*, Maui, Hawaii, USA, 1999.

Joseph Donald Novak. *Learning, Creating, and Using Knowledge: Concept Maps as Facilitative Tools in Schools and Corporations*. Lawrence Erlbaum and Associates, 1998.

Daniel E. O'Leary. Knowledge Management Systems: Converting and Connecting. *IEEE Intelligent Systems*, 13:30–33, 1998.

Wanda J. Orlikowski and Jack J. Baroudi. Studying Information Technology in Organizations: Research Approaches and Assumptions. *Information Systems Research*, 2, 1991.

Ilia Ovsiannikov, Michael A. Arbib, and Thomas H. McNeill. Annotation Technology. *International Journal on Human Computer Studies*, 50(4):329–362, 1999.

Chris D. Paice and Paul A. Jones. The Identification of Important Concepts in Highly Structured Technical Papers. In *Proceedings of the ACM SIGIR'93 Conference*. Association for Computing Machinery, 1993.

Henry Prakken, Chris Reed, and Douglas Walton. Argumentation Schemes and Burden of Proof. In Floriana Grasso, Chris Reed, and Guiseppe Carenini, editors, *Working notes of the 4th International Workshop on Computational Models of Natural Argument (CMNA2004)*, Valencia, Spain, 2004.

Joel R. Remde, Louis M. Gomez, and Thomas K. Landauer. SuperBook: an Automatic Tool for Information Exploration - Hypertext? In *Proceeding of the ACM conference on Hypertext*, pages 175–188, Chapel Hill, USA, November 1987.

Gerard Salton, Amit Singhal, Chris Buckley, and Mandar Mitra. Automatic Text Decomposition Using Text Segments and Text Themes. In *UK Conference on Hypertext*, pages 53–65, 1996.

Marlene Scardamalia. *CSILE/Knowledge Forum*, pages 183–192. 2004.

Fabrizio Sebastiani. Machine Learning in Automatic Text Categorization. *ACM Computing Surveys*, 34(1):1–47, March 2002.

John Seely Brown and Paul Duguid. The Social Life of Documents. Published on First Monday, http://www.firstmonday.dk/issue1/documents/index.html#03, May 1996.

Albert Selvin, Simon Buckingham Shum, Maarten Sierhuis, Jeff Conklin, Beatrix Zimmermann, Charles Palus, Wilfred Drath, David Horth, John Domingue, Enrico Motta, and Gangmin Li. Compendium: Making Meetings into Knowledge Events. In *Knowledge Technologies*, Austin, TX, USA, 2001.

Bertrand Sereno, Simon Buckingham Shum, and Enrico Motta. Semantic Annotation Support in the Absence of Consensus. In *Proceedings of the 1st European Semantic Web Symposium (ESWS04)*, Heraklion, Crete, May 2004a.

Bertrand Sereno, Simon Buckingham Shum, and Enrico Motta. Semi-Automatic Annotation of Contested Knowledge on the World Wide Web. In *Proceedings of the Alternate Track (Papers and Posters) of the 13th International World Wide Web Conference (WWW2004)*, New York City, NY, USA, May 2004b.

Bertrand Sereno, Simon Buckingham Shum, and Enrico Motta. ClaimSpotter: an Environment to Support Sensemaking with Knowledge Triples. In IUI05 IUI05, pages 199–206.

Frank M. Shipman and Raymond McCall. Supporting Knowledge Base Evolution with Incremental Formalization. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 285–291. Association for Computing Machinery, April 1994.

Karen Sparck Jones. What Might Be in a Summary ? In *Information Retrieval 93: Von der Modellierung zur Anwendung*, 1993.

Kazuo Sumita, Kenji Ono, and Seiji Miike. Document Structure Extraction for Interactive Document Retrieval Systems. In *Proceedings of the SIGDOC'93 Conference*, pages 301–310, 1993.

Tamara Sumner and Simon Buckingham Shum. From Documents to Discourse: Shifting Conceptions of Scholarly Publishing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI98)*, pages 95–102. Association for Computing Machinery, April 1998.

Tamara Sumner, John Domingue, Zdenek Zdrahal, Marek Hatala, Alan Millican, Jayne Murray, Knut Hinkelmann, Ansgar Bernardi, Stefan Weiss, and Ralph Traphoner. Enriching Representations of Work to Support Organizational Learning. In Andreas Abecker, Stefan Decker, Nada Matta, Frank Maurer, and Ulrich Reimer, editors, *Proceedings of the Interdisciplinary Workshop on Building, Maintaining, and Using Organizational Memories (OM '98), held with the 13th European Conference on Artificial Intelligence (ECAI '98)*, Brighton, UK, August 1998.

Dan Suthers, John Connelly, Alan Lesgold, Massimo Paolucci, Eva Erdosne Toth, Joe Toth, and Arlene Weiner. *Smart Machines in Education: the Coming Revolution in Educational Technology*, chapter Representational and Advisory Guidance for Students Learning Scientific Inquiry. AAAI/MIT Press, 2001.

John M. Swales. *Genre Analysis: English in Academic and Research Settings*. Cambridge University Press, 1990.

Simone Teufel. Argumentative Zoning for Improved Citation Indexing. In *Proceedings of the AAAI Spring Symposium on Exploring Attitude and Affect in Text: Theories and Applications*, Stanford University, USA, March 2004. American Association for Artificial Intelligence.

Simone Teufel and Marc Moens. Summarizing Scientific Articles: Experiments with Relevance and Rhetorical Status. *Computational Linguistics*, 28(4):409–445, December 2002.

Randall Trigg and Mark Weiser. TEXTNET: a Network-Based Approach to Text Handling. *ACM Transactions on Office Information Systems*, 4, January 1986.

Victoria Uren, Bertrand Sereno, Simon Buckingham Shum, and Gangmin Li. Interfaces for Capturing Interpretations of Research Literature. Presented at the Distributed and Collective Knowledge Capture Workshop, held with the Knowledge Capture Conference (KCAP), Sanibel, FL, USA, October 2003.

Tim van Gelder. Argument Mapping with Reason!Able. *Philosohpy and Computers*, pages 85–90, 2002.

Maria Vargas-Vera, Enrico Motta, John Domingue, Mattia Lanzoni, Arthur Stutt, and Fabio Ciravegna. MnM: Ontology Driven Semi-Automatic and Automatic Support for Semantic Markup. In *Proceedings of the 13th International Conference on Knowledge Engineering and Management (EKAW 2002)*, 2002.

Nina Wacholder, Judith L. Klavans, and David K. Evans. Evaluation of Automatically Identified Index Terms for Browsing Electronic Documents. In *Proceedings of the 6th conference on Applied Natural Language Processing*, pages 302–309, 2000.

Nina Wacholder, David K. Evans, and Judith L. Klavans. Automatic Identification and Organization of Index Terms for Interactive Browsing. In *Proceedings of the 1st ACM/IEEE-CS Joint Conference on Digital Libraries*, pages 126–134, 2001.

Dadong Wan and Philip M. Johnson. Computer Suported Collaborative Learning Using CLARE: the Approach and Experimental Findings. In *Proceedings of the Conference on Computer Supported Cooperative Work, CSCW 1994*, pages 187–198, 1994.

Karl E. Weick. Cosmos vs. Chaos: Sense and Nonsense in Electronic Contexts. *Organizational Dynamics*, 14:51–64, 1985.

Melvin Weinstock. Citation Indexes. In *Encyclopedia of Library and Information Science*, volume 5, pages 16–40. 1971.

WWW01. *Proceedings of the 10th International World Wide Web Conference (WWW2003)*, Hong-Kong, May 2001.

Yiming Yang and Jan O. Pedersen. A Comparative Study of Feature Selection in Text Categorization. In *Proceedings of the 14th International Conference on Machine Learning*, pages 412–420. Morgan Kaufmann, 1997.

Ka-Ping Yee. CritLink: Advanced Hyperlinks Enable Public Annotation on the Web. In *Proceedings of the Conference on Computer Supported Cooperative Work (CSCW 2002)*, New Orleans, USA, December 2002.