KNOWLEDGE MEDIA KMi INSTITUTE

Relation Extraction for Semantic Intranet Annotations

Technical Report kmi-06-17
August 2006

Lucia Specia
Claudio Baldassarre
Enrico Motta

Short versions:

- A hybrid approach for extracting semantic relations from texts. 2nd Workshop on Ontology Learning and Population (OLP2) at COLING/ACL 2006, pp. 57-64. July 22, Sydney.
- A hybrid approach for relation extraction aimed to semantic annotations. 7th Flexible Query Answering Systems, FQAS-2006, LNAI 4027, pp. 564-576. June 7-10, Milan.

The Open University

Relation Extraction for Semantic Intranet Annotations

Lucia Specia, Claudio Baldassarre, and Enrico Motta
Knowledge Media Institute & Centre for Research in Computing
The Open University
Walton Hall, MK7 6AA, Milton Keynes, UK
{L.Specia, C. Baldassarre, E.Motta}@open.ac.uk

Abstract

We present an approach for ontology driven extraction of relations from texts aimed mainly to produce enriched semantic annotations for the Semantic Web. The approach exploits linguistic and empirical strategies, by means of a pipeline method involving processes such as a parser, part-of-speech tagger, named entity recognition system, and pattern-based classification, and resources including ontology, knowledge and lexical databases. A preliminary evaluation with 25 sentences showed that the use of knowledge intensive resources and strategies together with corpus-based techniques to process the input data allows identifying and discovering relevant relations between known and new entity pairs mentioned in the text. Besides semantic web annotations, the system can be used for other tasks, including ontology population, since it identifies new instantiations of existent relations and entities, and ontology learning, since it discovers new relations, which are not part of the ontology.

1. Introduction

Relation Extraction is generally concerned about the identification of the semantic relations between pairs of terms in unstructured or semi-structured natural language documents. In our context, we use an *ontology-oriented* form of relation extraction, following the approximation suggested by (Bontcheva and Cunningham, 2003), *Ontology Driven Information Extraction*, regarding the research area in which ontologies are used to guide process of information extraction, in our case, relation extraction. In this report, we use the term “relation extraction” to refer to both (1) the identification of relations between two terms in texts when these relations are already “known”, that is, they belong to our domain ontology as possible relations between the concepts underlying those terms; (2) the discovery of new relations between two concepts underlying terms in the text, that is, relations that do not belong to our ontology.

Semantic relations extracted from texts are useful several applications, including acquisition of terminological data, development and extension of lexical resources or ontologies, question answering, information retrieval, semantic web annotation, etc. We focus on the application of both known and new relations to semantically annotate knowledge coming from raw text, within a framework to automatically acquire high quality semantic metadata for the Semantic Web. In that framework, applications such as a semantic web portal (Lei et al., 2006) acquire, integrate and manage heterogeneous data coming from texts, databases, domain ontologies, and knowledge bases. Known entities occurring in the text, i.e., entities that are included in the instantiation of our domain ontology, which we call “knowledge base”, are semantically annotated with their properties, also available in that knowledge base or in other databases. New entities, as given by a named entity recognition system, are annotated without any additional information. In that context, the goal of the relation extraction approach presented here is to extract relational knowledge about entities, i.e., to identify the semantic relations between pairs of entities in the input texts. Entities can be both known and new, since named entity recognition is also performed on the input texts. As previously mentioned, relations include those already predicted as possible by the domain ontology and new (unpredicted) relations.

The relation extraction approach makes use of the domain ontology and its corresponding instantiated knowledge base as main resources to guide the extraction, but also uses other knowledge-based and empirical resources and strategies. These include a lexical database, lemmatizer, syntactic parser, part-of-speech tagger, named entity recognition system, and pattern matching model. We experiment with this approach to annotate part of our intranet data, more specifically, texts from the Knowledge Media Institute (KMi)[1] internal newsletter.

Our hypothesis is that by integrating corpus and knowledge-based techniques and using rich linguistic processing strategies in an automated fashion, we can achieve effective results, accurately acquiring relevant relational knowledge. For semantic annotation tasks, this helps producing a rich representation of the input data. Moreover, this can be used for to populate the already existent ontology, since new instances of concepts and new instances of relations between concepts are identified. As emphasized by Magnini et al (2005), although the process of identifying new instances of concepts can be thought of as the well known task of Named Entity Recognition, in ontology population we are interested in classifying a term independently of the context in which it appears, differently from the named entity classification task, where all the occurrences of the recognized terms have to be classified.

It can also used as a first step for ontology learning (relation learning, but not concept learning), since new, potentially useful, relations are also extracted.

The rest of the report is organized as follows. We first describe related work on relation extraction aiming at various applications, including semantic annotations and ontology learning (Section 2). We then present our approach in detail, showing its architecture and describing each of its main components (Section 3). We also present the results of a preliminary evaluation of 25 sentences (section 4). Finally, we discuss our conclusions and the next steps (Section 5).

2. Related work

Several approaches have been proposed for the extraction of relations from unstructured sources. Recently, they have focused on the use of supervised or unsupervised corpus-based techniques in order to automate the task. A very common approach is based on pattern matching: given a previously defined set of patterns, usually composed by triples including two nominal expressions and one verbal expression, mainly SVO (subject-verb-object) triples, relations are extracted by matching the new text against the patterns, using strategies varying from exact matching to more elaborated metrics to identify the similarity between the terms/verbs in the pattern and in the text. The basic idea of these metrics is to somehow generalize the patterns so that they can be applied to new cases. In general, in approaches involving generalization of patterns, the way the patterns are generalised can vary considerably: some remove restrictions on certain elements of the patterns and then search the corpus instances which match the relaxed patterns (e.g. Soderland, 1999; Yangarber, 2000). Others (e.g. Chai and Biermann, 1999; Catala et al., 2000; Stevenson, 2004) use more linguistically motivated strategies, e.g., based on WordNet. In this process, problems such as sense ambiguity will arise. Most of the approaches keep the ambiguity or require the user to identify the correct sense during the definition of the extraction patterns (e.g., Catala et al., 2000), while a few others use word sense disambiguation methods for that (e.g., Chai and Biermann, 1999).

The patterns can be defined manually or automatically, in an unsupervised way. Among the unsupervised approaches based on pattern-matching, some employ a small number of seed

[1] http://kmi.open.ac.uk/

patterns of relations as a starting point to bootstrap a pattern learning process, using similarity measures to extract new patterns based on the initial ones. The approach proposed by (Yangarber et al., 2000), e.g., extracts relations by means of a process to generalize a set of seed patterns. The seed patterns are composed by SVO triples with named-entities as subject and object (e.g.: Company-hire-Person). Entities in the patterns are grouped in pairs, in order to provide sufficient frequency to obtain reliable statistics, and each pair is considered a pattern. These pairs are then used to gather the missing words in the missing role. A new SVO triple, also annotated with named entities, is classified as relevant or not by means of a probability-based metric. At each iteration, patterns in the corpus matching one of the already existent patterns are evaluated and one of them is chosen to be added to the set of patterns.

(Stevenson, 2004) also used a set of relevant seed patterns to learn other patterns to extract relations, based on the similarity among words in those patterns and in the text, given by a WordNet-based semantic similarity metric. SVO patterns are produced by a parser and those considered similar to already existent patterns are added to the set of patterns. The similarity measure employed is that defined by (Lin, 1998). In subsequent work (Stevenson and Greenwood, 2005), different similarity measures based on vector space models are employed.

Similarly to these two approaches, several others rely on the mapping of syntactic relations/dependencies onto semantic relations, but use other strategies instead of (or together with) pattern matching, mostly empirical strategies. The supervised approach proposed by (Soderland, 1999), e.g., learns extraction patterns of relations described within a single sentence from shallow parsed or non-annotated texts already marked with named entities. The tagging process is interactive and interleaved with the learning: the system presents the user a batch of instances to tag, and, based on the user's choices, induces a set of new rules from the expanded training set.

The work proposed by (Miller et al., 2000) treats relation extraction as a form of probabilistic parsing where parse trees are augmented to identify entities and the relations between them. Once this augmentation is made in a set of tree banks as training examples, the parser is trained using a supervised learning algorithm and can then be used with new sentences to extract their relations

(Gamalho et al., 2002) employ an unsupervised strategy for clustering semantically similar syntactic dependencies, according to their selectional restrictions. A set of interpretation rules are then applied to classify the syntactic dependencies in order to extract semantic relations. The semantic relations are organized according to a hierarchical structure similar to the one used in WordNet.

The supervised approach proposed by (Chieu and Ng, 2002) uses a maximum entropy learning algorithm as classification strategy. A score is computed for each pair of entities which occur in the same sentence (based on parsing information) to determine whether or not they represent a "true" relation.

(Roth and Yih, 2002) propose a probabilistic framework with supervised classifiers to recognize entities and relations. Classifiers are first trained separately for entities and relations, and then their output (conditional probabilities for each entity and relation) is used together with constraints induced between relations and entities, such as selectional restrictions of verbs established in terms of types of entities, in order to make global inferences for the most probable assignment for all entities and relations under consideration.

In the supervised approach proposed by (Zelenko et al., 2003), each occurrence of a pair of entities in a sentence is classified as a positive or negative instance of the relation. Instances are expressed by a pair of entities and their position in a shallow parse tree representing the sentence containing the pair. Classification uses both support vector machines and voted perceptron as learning algorithms with a specialized kernel model for that shallow parse representation.

(Zhao and Grishman, 2005) propose a supervised relation detection approach that combines clues from three different levels of syntactic processing - tokenization, sentence parsing and deep dependency analysis - using support vector machines as learning algorithm. Each source of information is represented by kernel functions. Composite kernels are then developed to integrate and extend individual kernels, allowing information from different levels to contribute to the process in an integrate way.

Focusing on complex relations, that is, *n*-ary relations that can involve *n* entities that are not in the same sentence, (McDonald et al., 2005) propose a two-stage method for extracting relations between named entities in biomedical texts. In the first stage, a supervised classifier based on maximum entropy recognizes all pairs of related entities, and a graph is built from the output of this classifier, having the entities as nodes and drawing edges between those which are related. The second stage scores maximal cliques in that graph as potential complex relation instances, that is, a subset of all the relations in which the binary classifier believes that all entities involved are pairwise related.

Similarly to our proposal, the framework presented by (Iria and Ciravegna, 2005) aims at the automation of semantic annotations according to ontologies. Several supervised algorithms can be used on the training data represented through a canonical graph-based data model. The framework includes a shallow linguistic processing step, in which corpora are analyzed and a representation is produced according to the data model, and a classification step, where classifiers run on the datasets produced by the linguistic processing step.

(Magnini et al., 2005) also present a proposal for annotating textual data according to a domain ontological, within the ONTOTEXT Project. Their idea is very similar to ours, in the sense that they also aim to identify mentions of known entities and relations in texts, as well as new entities and relations, in order to annotate a corpus of news and populate their ontology. This is an ongoing project and the particular strategy to extract relations has not been reported so far. However, the overall architecture is based on supervised learning from manually annotated texts.

Recently, many relation extraction approaches have been proposed focusing on the particular task of ontology development (learning, extension, population). These approaches aim to learn taxonomic or non-taxonomic relations between concepts, instead of lexical items. However, in essence, they can employ similar techniques as the other approaches described here to extract the relations. Additional strategies can be applied to determine whether the relations can be lifted from lexical items to concepts, as well as to determine the most appropriate level of abstraction to describe a relation. For example, (Maedche, 2002) proposes an approach for the extraction of non-taxonomic relations for ontology learning aiming at the Semantic Web. The approach uses an algorithm to discover generalised association rules and an already existent taxonomy as background knowledge in order to extract relations with the appropriate level of abstraction. The approach first uses a linguistic component to find out which pairs of lexical entries co-occur in the text, deriving statistical data indicating co-occurrences of concepts. The taxonomy is analysed to check whether taxonomic relations hold between these concepts. The generalised association rule algorithm determines confidence and support measures for the relations between the concepts and for relations at higher levels of abstraction. Finally, the algorithm determines the most appropriate level of abstraction to describe that conceptual relation by pruning the less adequate ones.

The approach proposed by (Reinberger et al., 2004) aims to extract semantic relations from text in an unsupervised way, with the ultimate goal of ontology development from scratch. Given a pre-processed corpus, SVO triples produced by a shallow parser are considered as potential relations between terms. Relevant relations are extracted by analysing simple statistics measures on the prepositional structures of the sentences. Both arguments in the triple (i.e., SO) must be part of a prepositional structure considered relevant by the statistical analysis.

Given a large amount of data, (Reinberger and Spyns, 2004) use unsupervised statistical methods based on frequency information over linguistic dependencies in order to establish relations between concepts from a corpus of the biomedical domain. Their main linguistic assumptions are the principle of selectional restrictions and the notion of co-composition. Clustering and pattern matching algorithms are used on a syntactic context to discover relations; however, these are not labelled.

(Schutz and Buitelaar, 2005) also rely on the verb as the element connection concepts. Their goal is to identify highly relevant triples over concepts from an ontology, in order to extend this ontology with such relations. The system extracts relevant verbs and their grammatical arguments from a domain-specific text collection, and then computes the corresponding relations via a combination of linguistic and statistical processing. In the linguistic phase, dependency structure analysis and named-entity/concept tagging recognition are carried out. During the statistical phase, to identify the most relevant triples, several computations are accomplished: relevance ranking, cross-reference of relevant nouns and verbs with predicate-argument pairs, and computation of co-occurrence-scores.

A comprehensive description of several other approaches for conceptual relation extraction aiming at ontology learning can be found in (Maedche, 2002) and (Gomez-Perez and Manzano-Macho, 2003). In general, these approaches only focus on learning new relations that will constitute or extend an ontology. Therefore, they do not explore already existent relations, which besides providing straightforward semantic annotations for our purposes, can be used as seeds to boost the relation learning process.

Among the other approaches previously described in this section, apart from the frameworks proposed by (Iria and Ciravegna, 2005) and (Magnini et al., 2005), both ongoing projects, none is aimed for semantic annotations. In fact, most of the relation extraction approaches are not designed for specific applications. Instead, they are designed for "general purpose" relation extraction. Therefore, it is hard to tell how well they would perform in real applications. In most of the evaluations that are presented, the systems extract usually simple and very well-known relations (such as "company-hires-person"), but there are no concerns about the relevance of these relations to a certain application and / or domain. Finally, among all the described approaches, only a few explore both knowledge-based and empirical resources and strategies, which can potentially convey both accurate and comprehensive results.

In the next section we describe our approach to relation extraction, which (1) was designed for the specific goal of producing intranet semantic annotations; and (2) merges knowledge-based and empirical features that have already shown to be effective in many of the previous works in an integrated way, in order to achieve more comprehensive and accurate results.

3. A hybrid approach for relation extraction

The proposed approach for relation extraction is hybrid in the sense that it employs knowledge-based (knowledge-intensive) and (weakly-supervised) corpus-based techniques. Our core strategy consists of mapping linguistic components with certain syntactic relationship (a linguistic triple) into their corresponding semantic components. This includes mapping not only the relations, but the linguistic terms linked by those relations. The detection of the linguistic triples involves a series of linguistic processing steps. The mapping between terms and concepts is guided by a domain ontology and a named entity recognition system. The identification of the relations relies on the knowledge available in the domain ontology, in lexical databases, and on a pattern-based classification model.

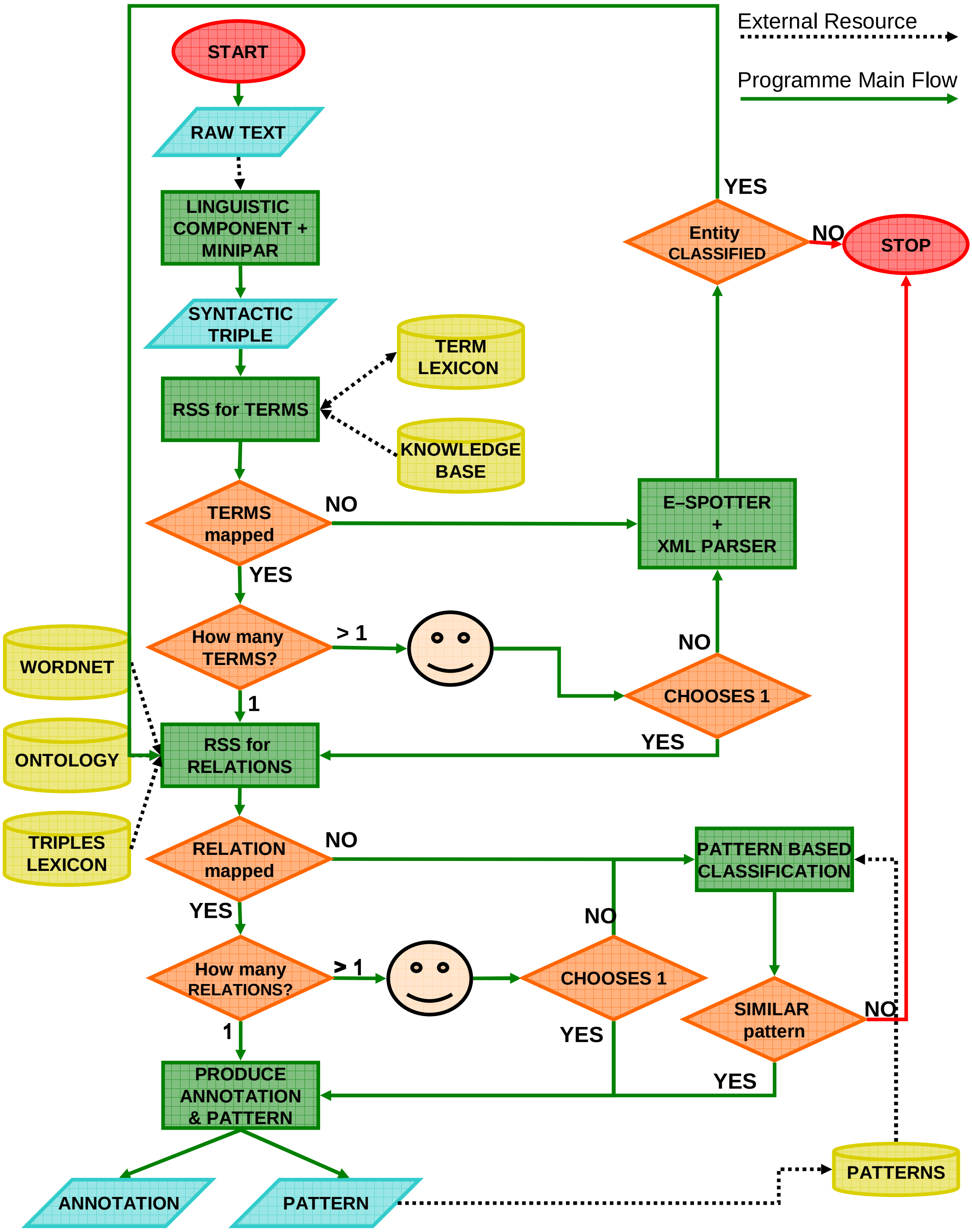


Figure 1: Architecture of the relation extraction approach

As previously mentioned, the main goal of this approach is to provide enriched information for the Semantic Web. Other potential applications include:

1) Ontology population: terms are mapped into new instances of concepts of the ontology, and instances of ontology relations between concepts are identified.

2) Ontology learning: new relations between concepts are identified, which can be used as a first step to extend an existent ontology (e.g., (Schutz and Buitelaar, 2005)). Certainly, a subsequent step to lift relations between instances to an adequate level of abstraction would be

necessary (e.g., (Maedche, 2002)). Alternatively, the relations extracted for instances could be validated by a domain expert.

The overall architecture of our system is illustrated in Figure 1. The main components are explained in details in the next sections.

3.1 Context and resources

As shown in Figure 1, the input to the system consists of electronic **Newsletter Texts** (KMi Planet[2]). These are texts describing news of several natures related to KMi members: projects, publications, events, awards, etc. The archive contains several hundreds newsletters published since 01/1997. The size of the texts can vary. For example, in a sample of 10 news texts taken from 10/12/2005 to 20/03/2006, it varied from 97 do 295 words. On average, the number of words per text in that sample was 178. The genre of texts we are addressing is, therefore, journalistic-like, while the domain is of that KMi.

Our domain **ontology** is called *KMi-basic-portal-ontology* and was designed based on the AKT reference ontology[3] to include concepts relevant to the KMi domain. All the instantiations of concepts in this ontology are stored in a knowledge base (**KB**) called *KMi-basic-portal-kb*.

The other two resources used in our architecture are the lexical database **WordNet** (Fellbaum, 1998) and the set of **Patterns**, described in Section 3.4.

3.2 Identifying linguistic triples

Given a newsletter text, the first step of the relation extraction approach is to process the natural language text in order to identify linguistic triples, that is, sets of three elements with a syntactic relationship, which can indicate potentially relevant semantic relations between two entities. In our architecture, this is accomplished by two modules, the **Linguistic Component (LC)** and the parser **Minipar**. LC is based on an adaptation of the linguistic component defined in Aqualog (Lopez et al., 2005). Aqualog is a question answering system which also uses the triple approach to identify and map linguistic triples into ontology triples. It uses GATE (Cunningham et al., 1992) infrastructure and resources, by means of GATE's API. The main processing resources from GATE employed in our architecture are:

1) ANNIE Tokenizer;
2) ANNIE Sentence splitter;
3) ANNIE Part-of-speech (POS) tagger;
4) ANNIE VP Chunker.

On top of these processing resources, which produce a series of syntactic annotations for the input text, LC uses a grammar, still under the execution of a GATE controller, in order to identify important linguistic triples. This grammar is implemented in Jape (Cunningham et al., 2000) and interpreted by a Jape transducer. A Jape grammar allows the definition of pattern of rules to recognize regular expressions using the annotations provided by GATE. One example of pattern to extract potential relations between nominal terms is illustrated in Figure 2.

```
((VERB)+ (RB)? NOUNA PREPS (VERB)?)
```

Figure 2: Example of Jape pattern

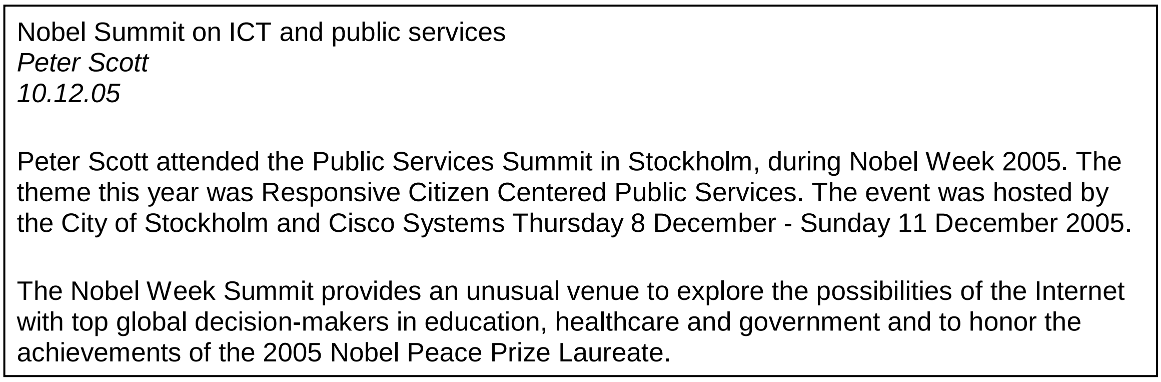
[2] http://news.kmi.open.ac.uk/kmiplanet/

[3] http://kmi.open.ac.uk/projects/akt/ref-onto/

The terms used in the patterns are macros previously defined to generalise the POS tags produced by the POS tagger. For example, "VERB" is a generic term to represent any kind of verb tag given by the POS tagger: VBN, VBZ, VBG, VBD, VBP, VB, and FVG. "RB" represents tags for adverbs, while NOUNA, tags for nouns and certain adjectives and pronouns, and PREPS, tags for certain prepositions. Thus, this pattern matches constructions with one or many verbs, optionally followed by an adverb, with a mandatory sequence of a NOUNA and a preposition, optionally followed by another verb.

The main type of construction aimed to be identified by our linguistic component involves a verb as indicative of a potential relation and two nouns as terms linked by that relation. However, our patterns can also account for other types of constructions, including, for example, the use of comma to implicitly indicate a relation, as in sentence (1). In this case, having identified that "AKT" is a *project* and "Enrico Motta" is a *person*, it is possible to guess the relation indicated by the comma (for example, "work"). Some examples triples identifies by our patterns, taking the newsletter text in Figure 3 as input, are given in Figure 4.

(1) "Enrico Motta, in AKT now, ….".

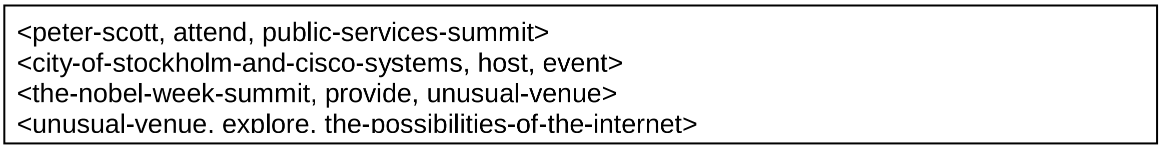


Nobel Summit on ICT and public services
Peter Scott
10.12.05

Peter Scott attended the Public Services Summit in Stockholm, during Nobel Week 2005. The theme this year was Responsive Citizen Centered Public Services. The event was hosted by the City of Stockholm and Cisco Systems Thursday 8 December - Sunday 11 December 2005.

The Nobel Week Summit provides an unusual venue to explore the possibilities of the Internet with top global decision-makers in education, healthcare and government and to honor the achievements of the 2005 Nobel Peace Prize Laureate.

Figure 3: Example of news



<peter-scott, attend, public-services-summit>
<city-of-stockholm-and-cisco-systems, host, event>
<the-nobel-week-summit, provide, unusual-venue>
<unusual-venue, explore, the-possibilities-of-the-internet>

Figure 4: Examples of linguistic triples for the text in Figure 3

Since Jape patterns are based only on shallow syntactic information, they are not able to capture certain potentially relevant triples. To overcome this limitation, we also employ a parser as a complementary resource to produce linguistic triples: **Minipar** (Lin, 1993). Minipar produces functional dependencies/relations for the components in a sentence, including subject and object relations with respect to a verb. This allows capturing some implicit relations, such as indirect objects and some long distance dependencies, which could not be identified by the Jape patterns. In fact, the use of Minipar complements the linguistic patterns, allowing a larger number of linguistic triples to be obtained.

We use the standalone version of Minipar, instead of GATE's version, since it performs differently in that framework. We then extract the syntactic relations of interest for each sentence, i.e., subject-verb-object (and modifiers) dependency triples. For example, some triples extracted for the text in Figure 3 are shown in Figure 5.

```
subject[peter_scott]+verb[attend]+verb_mod[during_nobel_week_2005]+object[public_services
_summit]+object_mod[in_stockholm]
subject[theme]+verb[be]+object[responsive]
subject[city]+subj_mod[of_stockholm]+verb[host]+object[event]
subject[nobel_week_summit]+verb[provide]+object[venue]
subject[nobel_week_summit]+verb[explore]+object[possibility]+object_mod[of_internet]
```

Figure 5: Examples of tuples extracted from Minipar's dependency trees

We convert Minipar's representation into a triple format, replicating the verb when it is related to more than one subject or object. Therefore, the intermediate representations provided both by GATE and the Jape grammars and by Minipar consist of triples of the type:

<noun_phrase, verbal_expression, noun_phrase>

3.3 Identifying ontological entities and relations

Given a linguistic triple, the next step is two verify whether the verbal expression in that triple conveys a relevant semantic relationship between entities (given by the terms) potentially belonging to an ontology. This is the most important phase of our approach and is represented by a series of modules in our architecture in Figure 1. As first step we try to map the linguistic triple into an ontology triple by using an adaptation of the Relation Similarity Service (RSS) developed in Aqualog (Lopez et al., 2005).

RSS tries to make sense of the linguistic triple by looking at the structure of the domain ontology and the information stored in its corresponding KB. Besides looking for an exact matching between the components of the linguistic triple and the components of the ontology and KB, RSS also considers partial matchings by using a set of resources in order to account for minor lexical or conceptual discrepancies between these two elements. These resources include metrics for string similarity matching, synonym relations given by WordNet, and a lexicon of previous mappings between the two types of triples.

One important feature of the RSS in Aqualog is that is meant to be used in an interactive fashion. When there is no matching between the linguistic and ontology triple, the user is expected to point out the appropriate mapping, so that the system can process the question. Additionally, the user is expected to disambiguate among several options of mappings for terms or relations. The goal in our approach, however, is to automate the annotation process as much as possible. We use RSS in a slightly different way to identify partial matching for terms (nominal phrases) and relations (verbal expressions) (modules **RSS for Terms** and **RSS for Relations**, respectively, in the architecture). As we explain in Sections 3.3.1 and 3.3.2, we first try to map terms, and if the mapping succeeds, we deal with relations. Additionally, we employ other components to find a matching with an ontology triple when there is no matching according to RSS, as discussed in Section 3.4.

3.3.1 Mapping terms

RSS for Terms checks if each of the terms, i.e., noun phrases, in the linguistic triple is an entity potentially belonging to our domain, according to the following attempts:

1) Search the lexicon of previously mapped terms (**Term Lexicon**) for exact matching of the term with any of the entries.

2) Search the KB for an exact matching of the term with any instance.

3) If there is not an exact matching in the KB and lexicon, apply string similarity metrics[4] to calculate the similarity between the given term and each instance of the KB. A hybrid scheme combining three metrics is used: jaro-Winkler, jlevelDistance and wlevelDistance. This scheme checks different combinations of threshold values for the three measures. It has proved to work reasonably well in experiments with many variations of metrics and thresholds.

3.1) If there is more that one possible matching, check whether any of them is a substring of the term. For example, the instance name for "Enrico Motta" is a substring of the term "Motta", and thus it should be preferred to any other instance.

For example, similarity values returned for the term "vanessa" with instances potentially relevant for the mapping are given in Figure 6. In this case, the combination of threshold specified for the three string metrics is met for a given instance of the KB, "Vanessa Lopez", and thus the mapping is accomplished.

```
Checking similarities for term "vanessa"

jaroDistance for "vanessa" and "vanessa-lopez" = 0.8461538461538461
wlevel for "vanessa" and "vanessa-lopez" = 1.0
jWinklerDistance for "vanessa" and "vanessa-lopez" = 0.9076923076923077
```

Figure 6: String similarity measures for the term "vanessa" and the instance "Vanessa Lopez"

3.2) If there is still more than one possible matching for the term (no substring matching or more than one substring matching), ask the user (which is assumed to be a domain expert) to choose one of the options. Here we could have assumed that there was not enough evidence to map that term, and therefore the linguistic triple should be discarded. However, initial experiments showed that in many cases we would be discarding relevant triples with valid possible matchings returned by the similarity metrics. In fact, in many cases the input linguistic term contains noise due to the use of patterns to recognize the terms (both by the LC and by a named entity recognition system, discussed later in this section). For example, a modifier appearing in the beginning of a sentence (and thus starting with a capital letter), when followed by a proper name, is usually considered part of the proper name. For example, in a sentence starting with "Director Enrico Motta …", "director-enrico-motta" can be extracted as a term. This prohibits the system to find an exact matching, but we expect the string similarity metrics to help. However, since they do not explore the semantics of the words, they capture other unrelated terms, and thus it is necessary to rely on the user to filter these terms out. The matching chosen by the user is added to the **Term Lexicon** so that it can be used in future mappings of the same term (step (1)).

At this stage, given a linguistic triple with 2 terms, <term$_1$, verbal_expression, term$_2$>, we will have four possible situations for the terms, representing total or partial or total matching with instances of the ontology, or no matching at all:

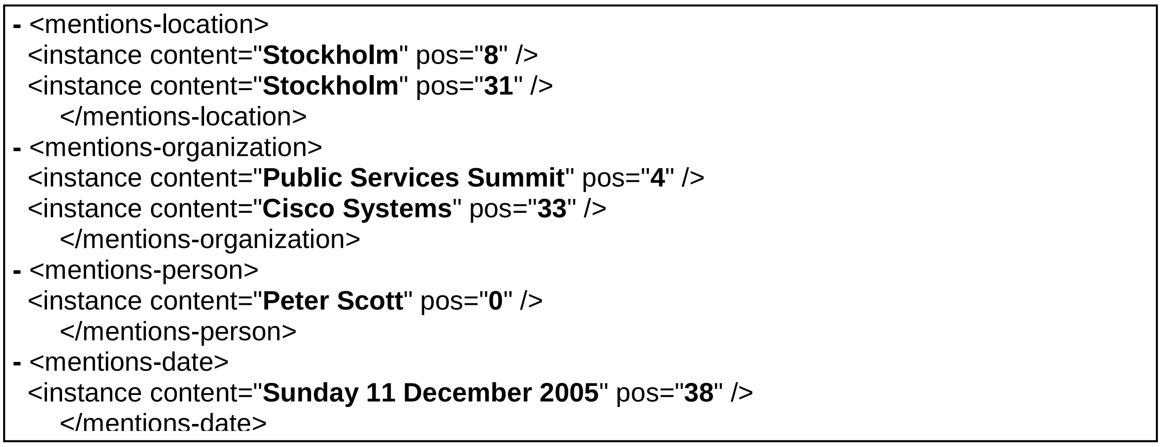
<instance$_1$, _ , instance$_2$>
<instance$_1$, _ , ?>
<?, _, instance$_2$>
<?, _ , ?>

4) If none of the previous attempts succeeds for a certain term (i.e., 3 last cases presented above), it can be the case that the term in the linguistic triple expresses a new entity, which is not

[4] Available in http://sourceforge.net/projects/simmetrics/.

part of the KB. In order to check if it in fact conveys a new entity that is relevant to our domain, or if it instead has no corresponding conceptual representation in our domain, we use a named entity recognition system to identify the class of the term (if any), according to the classes of the domain ontology. The system, **ESpotter++**, is an extension of the named entity recognition system ESpotter (Zhu et al., 2005).

ESpotter is an adaptive named entity recognition system based on a mixture of lexicon (gazetteers) and patterns. It can be adapted to particular domains by means of the customization of its lexicon and patterns of entities. In our approach, we extended ESpotter by including new entities (extracted from several gazetteers), new types of entities, and a small set of efficient patterns for the new types of entities. In Espotter++, all types of entities correspond to generic classes of the domain ontology under consideration. These types include: *person*, *organization*, *event*, *publication*, *location*, *project*, *research-area*, *technology*, *date*, *time*, etc. For example, for the text in Figure 3, part of the output produced by ESpotter++ is shown in Figure 7.



```
- <mentions-location>
  <instance content="Stockholm" pos="8" />
  <instance content="Stockholm" pos="31" />
     </mentions-location>
- <mentions-organization>
  <instance content="Public Services Summit" pos="4" />
  <instance content="Cisco Systems" pos="33" />
     </mentions-organization>
- <mentions-person>
  <instance content="Peter Scott" pos="0" />
     </mentions-person>
- <mentions-date>
  <instance content="Sunday 11 December 2005" pos="38" />
     </mentions-date>
```

Figure 7: Example of entities recognised by ESpotter++ for the news in Figure 3

If ESpotter++ is not able to identify the types of a term, it might be either the case that the term does not have any conceptual mapping (for example “it”), or the case that the conceptual mapping is not part of our domain ontology. Since we are not concerned with potential new concepts, which represent a complex issue on ontology learning, we do not attempt to map such term, and thus the process stops without any annotation.

It is important to emphasize that each of the two terms in the linguistic triple can be mapped according to different strategies (exact or partial matching or ESpotter++). However, if at least one of the terms is not mapped according to the mentioned steps, the process stops and no annotation is produced.

3.3.2 Mapping relations

In order to map the verbal expression of the linguistic triple into a conceptual relation, we assume that the terms of that triple have already been mapped either into instances of certain classes in the KB, by the component RSS for Terms, or into potential new instances of ontology classes, by ESpotter++, as explained in the last section. **RSS for Relations** then checks if the verbal expression in the triple matches any already existent possible relation between the classes (and superclasses) of those instances in the ontology. The following attempts are then made:

1) Search the ontology for an exact matching of the verbal expression with any existent relation between the classes (and superclasses) of the instances under consideration.

2) If there is not an exact matching with the ontology classes, apply the string similarity metrics to calculate the similarity between the given verbal expression and the possible relations between classes corresponding to the terms in the linguistic triple (as explained for terms).

3) If there is not an acceptable similarity or there is more than one matching considered similar, search for similar relation mappings for the classes under consideration in a lexicon of mappings previously accomplished according to users' choices in the on-line version of the question answering system Aqualog[5]. This lexicon contains a set of complete ontology triples with the original verbal expression which was mapped to the conceptual relation (**alias**). The use of this lexicon represents a simplified form of pattern matching, in which only exact matching is considered. Examples of entries in the lexicon are shown in Table 1.

class$_1$	alias (relation)	conceptual relation	class$_2$
project	works	has-project-member	person
project	cite	has-publication	publication

Table 1: Examples of the lexicon of patterns

4) If there is not an exact matching with entries in the lexicon of patterns and / or there are multiple possibilities of matchings coming from the string similarity metrics (step (2)), search for synonyms of the given verbal expression in WordNet, in order to verify if there is a synonym which matches (complete or partially, using string similarity metrics) with any existent relation for the classes (or superclasses) of the terms under consideration.

Three situations may arise from these attempts to map the linguistic triple into an ontology triple: (1) matching with one single relation of the ontology; (2) matching with more than one conceptual relation; and (3) no matching at all. That is:

<entity$_1$, (conceptual_relation)*, entity$_2$>.

If the matching attempt succeeds with only one conceptual relation, then the triple can be formalized into a semantic annotation. This allows the annotation of an already possible relation (according to the ontology) for two instances of the KB or new instances identified by ESpoter++. The produced triple generalized to the classes of the entities, i.e., <*class, conceptual_relation, class*>, is added to the set of **Patterns** of ontology triples, which is used to identify new relations (see Section 3.4).

If it there is more than one possible matching for the conceptual relation, the system gives the options to the user (assumed to be a specialized user). At a previous stage, we tried to use a disambiguation module to choose among multiple possible relations. This module, SenseLearner (Mihalcea and Csomai, 2005), is a supervised word sense disambiguation (WSD) system already trained on a corpus tagged with WordNet senses. Therefore, the only way it could be effectively used in our system was when the WordNet component used by RSS returned more than one synonym for the verb in a linguistic triple and these would match different conceptual relations. In that case, the WSD module could identify in which sense the verb was being used in the sentence and therefore allow the system to choose for one among the possible matchings.

Additionally, since the disambiguation system had been trained in another corpus, of a different domain, in most of the cases, the relation to be disambiguated was not present in the set of examples, and thus no sense could be retrieved to it. In order to benefit from this system (or any effective disambiguation system), we would have to train it on our corpus of newsletter texts, but this would require "sense" tagging this corpus with the appropriate senses, an effort that we consider would not pay off.

[5] In http://plainmoor.open.ac.uk:8080/aqualog/index.html.

Finally, if there is no matching for the relation, or if, given multiple "possible" relations, the user states that none of them is appropriate, the system triggers an alternative strategy to find out whether the verbal expression in the triple represents a relevant relation not covered by the ontology (or expressed in a different way): the **Pattern-based Classification** model (Section 3.4).

3.4 Identifying new relations – the pattern-based classification (PBC) model

The process described in Section 3.3 for the identification of relations accounts only for the relations already predicted as possible in the domain ontology. However, we are also interested in the additional information that can be provided by the text, in the form of new types of relations for known or new entities. In order to discover these relations, we employ a pattern matching strategy to identify relevant relations between classes of entities.

As previously mentioned, pattern matching strategies constitute the basis of many relation extraction approaches. This strategy has proved to be an efficient way to extract semantic relations, but in general has the drawback of requiring the possible relations to be previously defined. In order to overcome this limitation, we employ a **Pattern-based Classification** model, which can identify similar patterns based on a small initial number of patterns.

To identify the relations, we rely on the patterns already produced by the linguistic component, which are mostly SVO triples. Although this is not a highly expressive representation, it covers a significant number of relations on the text, with a low complexity. We consider patterns of relations between classes of entities, instead of the entities themselves, since we believe that it would not be possible to accurately judge the similarity between patterns of the kinds of entities we are addressing (names of people, locations, etc). Thus, our patterns consist of triples of the type <*class$_1$*, *conceptual_relation*, *class$_2$*>. These are compared against a given triple, also using the classes of its terms, in order to classify the relation in that triple as *relevant* or *non-relevant*.

Our pattern-based classification model is similar to the one proposed by (Stevenson, 2004) (Section 2). It is a weakly-supervised corpus-based module which takes as examples a small set of relevant SVO patterns, called seed patterns, and compares the pattern to be classified against all the relevant ones, using a WordNet-based semantic similarity metric. In our case, the initial seed patterns mixes patterns extracted from the lexicon generated by Aqualog's users, as described in Section 3.3.2, and a small number of manually defined relevant patterns. Currently, the set contain 65 patterns. This set of patterns is then enriched with new patterns as our system annotates relevant relations for given entities: the system adds new ontology triples to the initial set of **Patterns**. Some examples of seed patterns are illustrated in Table 2: each pattern consists of the ontology triple **<class$_1$, conceptual_relation, class$_2$>**, while **alias (relation)** is the relation used in the linguistic triple, which was mapped into the **conceptual_relation**.

class$_1$	alias (relation)	conceptual relation	class$_2$
project	has-member	has-project-member	person
project	publish	has-publication	publication
person	publish	has-publication	conference
person	work	work-for	project
person	implement	develop	technology
person	participate	attend-by	event
organization	hire	has-employee	person

Table 2: Examples of seed patterns

Our similarity metric requires that the terms and relations in both the pattern and the triple being mapped belong to WordNet. Since WordNet is not meant to contain multi-word expressions, such as “has-project-member”, in order to compute the similarity for these multi-word expressions, we defined a set of simple heuristics to identify the most representative word in the expression. If the multi-word expression is the conceptual relation of the pattern, we use the **alias** of relation as most representative word. For example, for the multi-word conceptual relation “work-for”, we use the alias “work”. However, if the multi-word expression is one of the classes, e.g., “organization-unit”, or is the verbal expression in the linguistic triple we want to map, i.e., cases for which there is no alias, the heuristics use the governance relations given by Minipar to identify the syntactic head of the expression. They also state that the head has to be a verb for the relation, but a noun for the terms. For example, the head of the verbal expression “has-publication” is “has”, while the heat of the term “organization-unit” is “unit”. Having identified the most significant word of each multi-word expression, we use them to calculate the similarity score as if they were single words in the pattern/triple.

Likewise Stevenson, we use the semantic similarity metric proposed by (Lin, 1998), which has shown to be suitable for this purpose. This metric assigns numerical values to each node in WordNet hierarchy representing the amount of information it contains, *IC*, which is derived from corpus probabilities: *IC(s) = -log(Pr(S))*. To calculate the similarity between two senses s_1 and s_2, it takes the lowest common subsumer, $lcs(s_1,s_2)$, which is defined as the sense with the highest information content which subsumes both senses in the WordNet hierarchy, that is:

$$\frac{sim(s_1,s_2) = 2\times IC(lcs(s_1,s_2))}{IC(s_1)+IC(s_2)}$$

Stevenson adapts this metric to compute the similarity between two words, w_1 and w_2, instead of two senses, by analysing the set of senses for each of the words, *S(w)*, and choosing the pair of senses which maximizes the similarity score, according to the formula:

$$word\_sim(w_1,w_2) = \underset{\substack{1\le i\le|S(w_1)|\\ 1\le j\le|S(w_2)|}}{MAX}\; sim(s_{1i},s_{2j})$$

Additionally, the author extends the metric to compute the similarity between two patterns, p_1 and p_2, consisting of *m* and *n* elements, respectively (in our case, *m* and *n* are always 3):

$$psim(p_1,p_2) = \frac{\sum_{i=1}^{n} word\_sim(p_{1i},p_{2i})}{MAX(m,n)}$$

We consider relevant the patterns for which the score is greater than the threshold of 0.90 for the formula above. In that case, a new annotation is produced for the entities in the linguistic triple and the new relation. Additionally, as previously mentioned, if the triple is not part of the set of patterns, it is added to it, for future use.

The level of novelty of the extracted (new) relations is determined as a consequence of the value of the threshold for the formula above (*psim*). A high threshold means that the relation is considered relevant only if triple is very similar to at least one of the existent patterns. Is also guarantees that even if the two classes in the triple completely match classes in a pattern (score = 1 for each of the classes), the relation will not be considered relevant unless it has some similarity with the relation in the pattern (if the relation similarity is = 0, *psim* = 0.67). A slightly smaller value for the threshold could make other also relevant relations to be acquired, but since we add each the mapped triple to the set of patterns, relaxing the threshold to allow less closely related relations to be learned can be tricky: it will possibly result in noise, which will be propagated and increased as the system is used.

One example of relation "learned" by the pattern-based classification model for the given sentence is shown in Figure 8. The figure also shows the linguistic triple, partial ontology triple (with terms mapped only), triple submitted to the system (with the classes of the mapped terms), most similar pattern, similarity value achieved for that pattern, and finally the new pattern added to the set of patterns for future use.

Sentence: KMI is headed by Enrico Motta
Linguistic triple: <kmi, headed, enrico-motta>
Partial ontology triple: <knowledge-media-institute-at-the-open-university, ?, kmi-director-enrico-motta>
Pattern given to PBC: <r-and-d-institute-within-educational-organization, headed, person>
Most similar pattern: <organization, led-by, affiliated-person>
Similarity value: 1
Relation learned: headed-by
New pattern: <r-and-d-institute-within-educational-organization, headed-by, person>

Figure 8: Example of use of the PBC model

It is important to notice that, although WordNet is also used in the RSS component, in that case only synonyms are being checked, while in this case the similarity metric explores deeper information in WordNet, considering the meaning (senses) of the words and the hierarchical structure of WordNet. It is also important to distinguish the semantic similarity metrics employed here from the string metrics used in RSS. String similarity metrics simply try to capture minor variations on the strings representing terms/relations, they do not account for the meaning of those strings. Alternative semantic similarity metrics exploiting other information in WordNet (Pedersen et al., 2004) can be investigated in future work.

3.5 Annotating relevant relations

In order to formalize the relations extracted, we use OWL representations, as specified by the Semantic Web framework. For example, the representation of the entity "John Domingue" and relations extracted from the sentences in Figure 9 is given in Figure 10.

John Domingue is member of AKT
John Domingue is the Scientific Director for DIP

Figure 9: Example of input sentences

```
<rdf:Description rdf:ID="john-domingue">
  <rdf:type>
    <owl:Restriction>
      <owl:hasValue rdf:resource="#dip"/>
      <owl:onProperty rdf:resource="#has-project-leader"/>
    </owl:Restriction>
  </rdf:type>
  <rdf:type>
    <owl:Restriction>
      <owl:hasValue rdf:resource="#akt"/>
      <owl:onProperty rdf:resource="#has-project-member"/>
    </owl:Restriction>
  </rdf:type>
</rdf:Description>
```

Figure 10: OWL annotations produced for the news in Figure 9

4. Evaluation and discussion

In order to evaluate our system, we manually selected a set of 25 sentences from the newsletters that mentioned entities belonging to our domain (people, technologies, projects, etc.), i.e., sentences from where instances and relations could be extracted. We restricted the selection to relatively simple sentences, having one potential triple per sentence. At this stage, we are more concerned about the precision of the system, instead of its coverage. Complex sentences cannot be processed by the system, due to limitations inherited from the components extracting the linguistic triples (linguistic component and Minipar parser). For example, Minipar is not able to process sentences with more than 1000 characters, and cannot capture very long distance dependencies (involving subject and object elements, in this case). These are, however, expected limitations even for the state of the art linguistic processing tools. Examples of sentences used in our evaluation are shown in Figure 11.

(1) Adam Ingram visited the Open University today.
(2) Sun Microsystems hosts an internal KMi Stadium Webcast, on March 12th 1997.
(3) Hon Roy MacLaren visited Milton Keynes today.
(4) Dr Hans Geiser visited the Open University on Wednesday 21st May, 1997.
(5) Dr Geiser, Director of the UN Staff College in Turin, visited the OU.
(6) Roxana is currently working in KMi on the SUPER project.
(7) KMi's Deputy Director John Domingue is the Scientific Director for DIP.
(8) The DIP team at KMi incorporates Roxana Belecheanu.
(9) KMi hosts the 9th Computational Linguistics UK Research Colloquium (CLUK 06).
(10) Peter Scott attended the Public Services Summit in Stockholm, during Nobel Week 2005.
(11) Simon Buckingham Shum has been working with the ILO's HIV/AIDS Education in the Workplace programme.
(12) Dr Dawei Song has joined KMi as a Senior Lecturer on September 12th, 2005.

Figure 11: Examples of news sentences used in the evaluation

As a measure of overall performance, we calculated coverage, precision and recall for the produced ontology triples, given the input sentences, as shown in Table 3. Since we have two parallel modules processing the input sentences to produce the intermediate linguistic triples (i.e., Linguistic Component – LC – and Minipar), we can have as twice triples as input sentences. However, as we discuss later, when both modules are able to produce correct linguistic triples, the corresponding resulting ontology triples are consistent, i.e., they are exactly the same for linguistic triples generated by both LC and Minipar. On the other hand, when LC or Minipar produce incorrect linguistic triples, they are not mapped into ontology triples. Therefore, here we consider the intersection of the produced ontology triples, which amounts to 17.

# Sentences	Coverage	Recall	Precision
25	(17/25) = 0.68	(15/25) = 0.60	(15/17) = 0.88

Table 3: Overall figures for ontology triples generated from input sentences

Since we have different components processing the various steps of the system, we also present, in Table 4 – Table 7, the results of submitting our test set of 25 sentences to the system according to each of these components. Essentially, we have:

- 2 modules producing linguistic triples: Linguistic Component (LC) and Minipar;
- 2 modules identifying terms: Relation Similarity Service (RSS) and ESpotter++;

- 2 modules identifying relations: RSS and Pattern-based Classification (PBC).

In Table 4 we focus on the performance of the modules in identifying linguistic triples from the sentences. Given the total of input sentences *n* (*n* = 25), with one potential linguistic triple each, the column "**# Ling. triples**" shows the number (*m*) of linguistic triples actually identified by the modules. The column "**Coverage**" shows the proportion of identified triples given the total of potential linguistic triples *n*, i.e., expresses the coverage (*m/n*) of LC and Minipar in identifying linguistic triples (correctly or incorrectly). The column "**Recall**" shows the proportion of *correctly* identified linguistic triples given the total *n*. Finally, the last column, "**Precision**", shows the proportion of *correctly* identified linguistic triples given the number of identified linguistic triples *m*.

	# Ling. triples (*m*)	**Coverage**	**Recall**	**Precision**
LC	12	(12/25) = 0.48	(11/25) = 0.44	(11/12) = 0.92
Minipar	21	(21/25) = 0.84	(19/25) = 0.76	(19/21) = 0.90

Table 4: Figures for linguistic triples extracted by the LC and Minipar

Regarding the overlapping in the triples identified by both components in Table 3, looking at the linguistic triples correctly identified by each of them, we see that each identifies a different subset of triples. In this case, LC identifies 2 triples that were not covered by Minipar, while Minipar identifies other 11 triples that were not covered by LC. Therefore, taking the non-overlapping set of linguistic triples provided by both components, we obtain 23 triples.

In Table 5 we show the performance of the system on mapping the *correct* linguistic triples identified by each of the modules (LC and Minipar) into ontology triples, without going into details about which internal modules were involved in this mapping. Ideally, each linguistic triple should produce an ontology triple. The column "**# Ont. triples**" gives the number of ontology triples created from correct linguistic triples. The column "**Coverage**" shows the percentage of mapped triples, over the total of correct linguistic triples given by LC and Minipar (11 by LC and 19 by Minipar), i.e., expresses the coverage of the system in identifying ontology triples from correct linguistic triples (correctly or incorrectly). The column "**Recall**" shows the percentage of correctly identified ontology triples over the total of correct linguistic triples, i.e, expresses the recall of the system in identifying ontology triples. Finally, the column "**Precision**" shows the ratio of correctly identified ontology triples by the number of identified ontology triples.

	# Ont. triples	**Coverage**	**Recall**	**Precision**
Linguistic triple by LC	10	(10/11) = 0.91	(9/11) = 0.82	(9/10) = 0.90
Linguistic triple by Minipar	14	(14/19) = 0.74	(12/19) = 0.63	(12/14) = 0.86

Table 5: Figures for ontology triples produced from the LC's and Minipar's linguistic

The fact that we are using two modules to identify linguistic triples allowed us to identify more ontology triples, since, as previously mentioned, LC and Minipar were able to recognize linguistic triples for different sentences. In fact, triples produced by Minipar allowed 6 different ontology triples to be mapped, when compared to the mappings for linguistic triples produced by LC. On the other hand, linguistic triples produced by LC gave origin to 3 additional ontology triples. Regarding the consistency of the ontology triples produced from linguistic triples provided by both components, it is important to say that when the same linguistic triple was mapped by both components, the resulting ontology triples were identical. The resulting incorrect mappings were due to incorrect mapping of either any of the terms or the verbal expression in the linguistic triple. Similarly, null mappings were due to the lack of mapping for either one of the terms or the verbal expression.

In Tables 6 and 7 we analyze each of these elements, i.e., terms and verbal expressions, individually, also taking into account which module was used to map each of them. Therefore, still focusing on the performance of the system to map linguistic triples into ontology triples, now we analyze first the performance of the system in mapping terms into conceptual entities (Table 6), and then the performance in mapping verbal expressions into conceptual (Table 7). Here we do not distinguish if the triple were identified by LC or Minipar; instead, we consider the set of all 23 (non-overlapping and correct) linguistic triples. Therefore, we can have at most 46 terms and 23 verbal expressions mapped. In Table 6, given the total of terms in the correctly identified linguistic triples (each linguistic triple has two terms), the column "**# Terms**" shows the number of terms that were mapped to a conceptual representation (instance or new entity), by both RSS and ESpotter++. The column "**Coverage**" shows the proportion of (correctly or incorrectly) mapped terms given the total of terms to be mapped. The column "**Recall**" shows the proportion of correctly mapped terms given the total of terms to be mapped. Finally, the column "**Precision**" shows the proportion of correctly mapped terms given the number of mapped terms.

	# Terms	Coverage	Recall	Precision
Mapped by RSS	35	(35/46) = 0.76	(32/46) = 0.70	(32/35) = 0.91
Mapped by ESpotter++	8	(8/11) = 0.72	(8/11) = 0.72	(8/8) = 1

Table 6: Figures for conceptual terms mapped by RSS and ESpotter++

It is important to notice that ESpotter++ is only triggered when RSS does not find any possible direct mapping for the term and the user is not satisfied with the options of mappings given by RSS and chooses to use ESpotter++. Therefore, here the figures reflect the use of ESpotter++ in those cases only, that is, for the 11 terms out of the total of 46, since 35 were mapped by RSS.

Finally, in Table 7, given the total of verbal expressions in the 23 correctly identified linguistic triples, the column "**# Relations**" shows the number of verbal expressions that were mapped into conceptual relations (known or new), by both RSS and PBC. The column "**Coverage**" shows the proportion of (correctly or incorrectly) mapped verbal expressions given the total of verbal expressions to be mapped, i.e., 23. The column "**Recall**" shows the proportion of correctly mapped verbal expressions given the total of verbal expressions to be mapped. The column "**Precision**" shows the ratio of correctly mapped verbal expressions by the number of mapped verbal expressions. It is worth recalling that the system only attempts to map verbal expressions once both terms in the linguistic triple are mapped. So, in some cases, the relation was not found because at least one of the terms was not mapped. Out of the possible number of relations to be mapped (23), 2 were not mapped because of this reason, while 1 case was not mapped because one of the terms was incorrectly mapped (by RSS). Other 3 cases were not mapped because of the low coverage of our resources (KB, set of patterns). For the remaining 17 cases, a relation was found. In only 2 of the cases, the relation found (by PBC) was incorrect, and this always due to incorrect mappings of the terms.

	# Relations	Coverage	Recall	Precision
Mapped by RSS	6	(6/23) = 0.26	(6/23) = 0.26	(6/6) = 1
Mapped by PBC	11	(11/17) = 0.65	(9/17) = 0.53	(9/11) = 0.81

Table 7: Figures for conceptual relations mapped by RSS and PBC

Again, it is worth noticing that the PBC model is only triggered when RSS cannot find any relation to directly map the verbal phrase (the threshold is not achieved) and the user is not satisfied with any of the options of mappings given by RSS and chooses to "learn" the relation. Therefore, here we are only taking into account the use of PBC in those cases, and not to map all the verbal phrases (i.e., for the 17 cases out of the total of 23, since 6 were mapped by RSS).

The coverage (and consequently, recall) of RSS was low particularly due the lack of knowledge in our KB. In this case, we should rely on the PBC module. However, here the coverage of this module was also not very high, due to the very high threshold specified to determine whether a relation is similar to the ones in the set of patterns. Therefore, so far the "new" relations being learned are very closely related to the ones already existing in our ontology or manually defined seed patterns. In order to capture relevant relations which are not so close to the ones already existent, we can try different thresholds (and metrics) to learn these relations. Alternatively, we can increase the set of seed patterns.

5. Conclusions and future work

We presented a hybrid approach for the extraction of semantic relations from texts. It is mainly aimed to produce enriched semantic annotations for the Semantic Web, but can be also used for ontology population and ontology learning. Initial evaluation experiments on a small dataset of news texts yielded promising results. We believe these are due to the use of multiple components to tackle certain problems, mainly, the use of both shallow (Jape) patterns and syntactic parser to identify linguistic triples, the use of several resources to accomplish the mapping of these triples into ontology concepts and relations, and the use of a machine learning-based technique (PBC) to capture relations that are not explicit in the ontology. In fact, the evaluation showed that these different modules complement each other in producing correct annotations.

In future work we intend to improve the Linguistic Component, so that it can cover more non-conventional (SVO-type) relations. Particularly, we are interested addressing complex (n-ary) relations, instead of relations between two elements only. This will improve the linguistic coverage of the approach.

In order to improve the linguistic-conceptual mapping coverage, we plan to add more metadata and their corresponding ontologies to RSS (both RSS for Terms and RSS for Relations), and possibly use Swoogle (Ding et al., 2004) to gather this ontologies and metadata from the web. We also plan to add other lexical resources to improve the mapping of relations, such as FrameNet (Baker et al., 1998), which has a richer and more structured description of lexical items. The inclusion of these resources in our architecture is illustrated in Figure 12. This will also decrease the need of user interaction when mapping terms and relations, since more instances and classes will be analyzed, potentially covering the terms and relations in the linguistic triple with exact matching. Certainly, problems may arise such as having the same term/verbal expression in more than one ontology, representing different concepts / relations. We can use other textual sources (Wikipedia, web) to gather more information to disambiguate entities (Bunescu and Pasca, 2006, e.g.).

Even with the use of multiple resources to provide a broader mapping coverage, certainly we will encounter linguistic triples with new relations, which do not belong to any of our ontologies. We intend to carry on using the pattern-based classification model to cope with these new relations; however, we plan to extend the set of seed patterns, and also to play with different threshold for the semantic similarity metric. In order to avoid the system to learn non-relevant relations, we will have to perform some additional reasoning on the learned relations in order to verify whether they are really relevant to our domain.

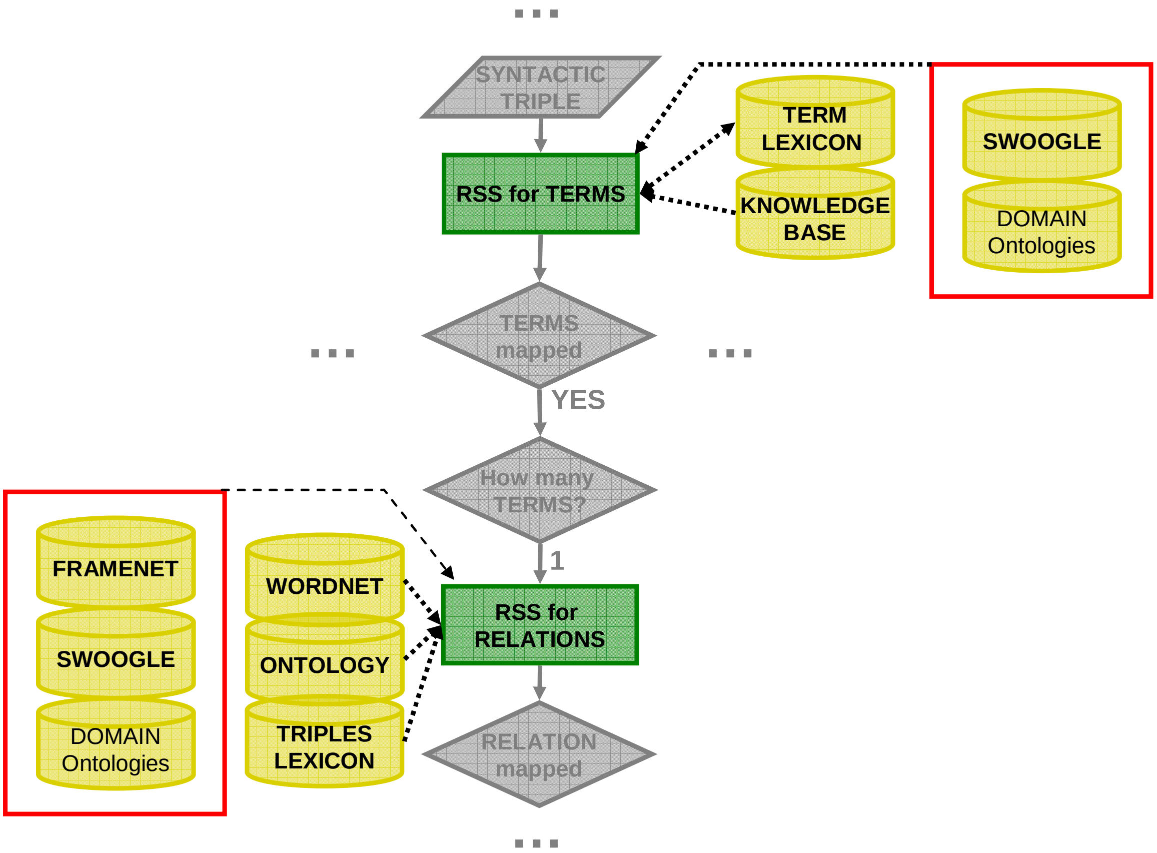


Figure 12: New resources to be added to our architecture

Acknowledgments

This work was funded by the Advanced Knowledge Technologies (AKT) Interdisciplinary Research Collaboration (IRC), sponsored by the UK Engineering and Physical Sciences Research Council under grant number GR/N15764/01, and the Open Knowledge project sponsored by the European Commission as part of the Information Society Technologies (IST) program under EC grant numbers IST-FF6-027253.

References

C. F. Baker, C. J. Fillmore, and J. B. Lowe. 1998. The Berkeley FrameNet Project. *COLING-ACL-1998*, Montreal, pp. 86-90.

K. Bontcheva and H. Cunningham. 2003. The Semantic Web: A New Opportunity and Challenge for Human Language Technology. Workshop on Human Language Technology for the Semantic Web and Web Services at International Semantic Web Conference, Florida.

R. Bunescu and M. Pasca. 2006. Using Encyclopaedic Knowledge for Named Entity Disambiguation. *11th Conference of the European Chapter of the Association for Computational Linguistics (EACL-2006)*, Trento, pp. 9-16

Catala, N. Castell, M. Martin. 2000. Essence: A Portable Methodology for Acquiring Information Extraction *Patterns. 14th European Conference on Artificial Intelligence (ECAI*), Berlin, pp. 411-415.

J. Chai, and A. Biermann. 1999. The Use of Word Sense Disambiguation in an Information Extraction System. *16th National Conference in Artificial Intelligence and Eleventh Annual Conference on Innovative Applications of Artificial Intelligence*, Orlando, pp. 850-855.

H. L. Chieu and H. T. Ng. 2002. A Maximum Entropy Approach to Information Extraction from Semi-Structured and Free Text. *18th AAAI/IAAI*, Edmonton, Alberta, pp. 786-791.

H. Cunningham, D. Maynard, K. Bontcheva, and V. Tablan. 2002. GATE: A Framework and Graphical Development Environment for Robust NLP Tools and Applications. *40th Annual Meeting of the Association for Computational Linguistics (ACL-2002),* Philadelphia.

H. Cunningham, D. Maynard, and V. Tablan. 2000. JAPE: a Java Annotation Patterns Engine. *Tech. Report CS--00--10,* University of Sheffield, Department of Computer Science.

L. Ding, T. Finin, A. Joshi, R. Pan, R. Scott Cost, Y. Peng, P. Reddivari, V. C. Doshi, and J. Sachs. 2004. Swoogle: A Search and Metadata Engine for the Semantic Web. *13th ACM Conference on Information and Knowledge Management*, Washington DC.

C. D. Fellbaum (ed). 1998. *WordNet: An Electronic Lexical Database*. The MIT Press.

P. Gamallo, M. Gonzalez, A. Agustini, G. Lopes, and V. S. de Lima. 2002. Mapping syntactic dependencies onto semantic relations. *ECAI Workshop on Machine Learning and Natural Language Processing for Ontology Engineering,* Lyon, France.

A. Gomez-Perez and D. Manzano-Macho. 2003. *A Survey of Ontology Learning Methods and Techniques*. Deliverable 1.5, OntoWeb Project.

J. Iria and F. Ciravegna. 2005. Relation Extraction for Mining the Semantic Web. *Dagstuhl Seminar on Machine Learning for the Semantic Web, Dagstuhl,* Germany.

Y. Lei, M. Sabou, V. Lopez, J. Zhu, V. Uren, and E. Motta. 2006. An infrastructure for Acquiring High Quality Semantic Metadata. *3rd European Semantic Web Conference (ESWC 2006),* Budva, Montenegro.

D. Lin. 1993. Principle based parsing without overgeneration. *31st Annual Meeting of the Association for Computational Linguistics (ACL-1993),* Columbus, pp. 112-120.

D. Lin. 1998. An information-theoretic definition of similarity. International Conference on Machine Learning.

V. Lopez, M. Pasin, and E. Motta. 2005. AquaLog: An Ontology-portable Question Answering System for the Semantic Web. *2nd European Semantic Web Conference (ESWC 2005),* Creete, Grece.

B. Magnini, M. Negri, E. Pianta, L. Romano, M. Speranza, and R. Sprugnoli. 2005. From Text to Knowledge for the Semantic Web: the ONTOTEXT Project. SWAP 2005, Semantic Web Applications and Perspectives, Trento.

A. D. Maedche. 2002. *Ontology Learning for the Semantic Web*, Kluwer Academic Publishers, Norwell, MA.

R. McDonald, F. Pereira, S. Kulick, S. Winters, Y. Jin, and P. White. 2005. Simple Algorithms for Complex Relation Extraction with Applications to Biomedical IE. *43rd Annual Meeting of the Association for Computational Linguistics (ACL-2005),* Ann Arbour, Michigan, pp. 491-498.

S. Miller, H. Fox, L. Ramshaw, and R. Weischedel. 2000. A novel use of statistical parsing to extract information from text. *6th ANLP-NAACL*, Seattle, pp. 226-233.

R. Mihalcea and A. Csomai. 2005. SenseLearner: Word Sense Disambiguation for All Words in Unrestricted Text. *43rd Annual Meeting of the Association for Computational Linguistics (ACL-2005),* Ann Arbor.

T. Pedersen, S. Patwardhan, and J. Michelizzi. 2004. WordNet::Similarity - Measuring the Relatedness of Concepts. *5th Annual Meeting of the North American Chapter of the Association for Computational Linguistics (NAACL-04)*, Boston, pp. 38-41.

M. Reinberger and P. Spyns. 2004. Discovering knowledge in texts for the learning of DOGMA inspired ontologies. *ECAI 2004 Workshop on Ontology Learning and Population*, Valencia, pp. 19-24.

M.L. Reinberger, P. Spyns, and A.J. Pretorius. 2004. Automatic initiation of an ontology. *On the Move to Meaningful Internet Systems 2004: CoopIS, DOA, and ODBASE,* LNCS 3290, Napa, Cyprus, pp. 600-617.

D. Roth and W. Yih. 2002. Probabilistic reasoning for entity & relation recognition. *19th COLING*, Taipei, Taiwan, pp. 1-7.

A. Schutz and P. Buitelaar. 2005. RelExt: A Tool for Relation Extraction from Text in Ontology Extension. *4th International Semantic Web Conference (ISWC-2005)*, Galway, pp. 593-606.

S. Soderland. 1999. Learning information extraction rules for semi-structured and free text. *Machine Learning*, 34

M. Stevenson. 2004. An Unsupervised WordNet-based Algorithm for Relation Extraction. *4th LREC Workshop Beyond Named Entity: Semantic Labeling for NLP Tasks*, Lisbon.

M. Stevenson and M. Greenwood. 2005. A Semantic Approach to IE Pattern Induction. *43rd Meeting of the Association for Computational Linguistics (ACL-05),* Ann Arbour, Michigan, p. 379-386.

D. Zelenko, C. Aone, and A. Richardella. 2003. Kernel Methods for Relation Extraction. *Journal of Machine Learning Research*, (3):1083-1106.

S. Zhao and R. Grishman. 2005. Extracting Relations with Integrated Information Using Kernel Methods. *43d Annual Meeting of the Association for Computational Linguistics (ACL-2005)*, Ann Arbor.

J. Zhu, V. Uren, and E. Motta. 2005. ESpotter: Adaptive Named Entity Recognition for Web Browsing. *3rd Conf. on Professional Knowledge Management*, Kaiserslautern, pp. 518-529.

R. Yangarber, R. Grishman and P. Tapanainen, 2000. Unsupervised Discovery of Scenario-Level Patterns for Information Extraction. *6th ANLP*, Seattle, pp. 282-289.