KNOWLEDGE MEDIA INSTITUTE

Fusing Automatically Extracted Annotations for the Semantic Web

Technical Report kmi-06-12
July 2006

Andriy Nikolov

The Open University

Abstract

One of the necessary preconditions of the Semantic Web initiative is the availability of semantic data. The Web already contains large amounts of information intended for human users. This information is mainly stored as hypertext, which must be semantically annotated to make it accessible for software agents. The amount of information on the Web makes it impossible to solve the annotation task manually. Therefore the use of automatic information extraction algorithms is essential. These algorithms use various NLP and machine learning techniques to extract information from text. The information extracted from different sources must then be integrated in a knowledge base, so that it can be queried in a uniform way. This integration process is called *knowledge fusion*. However, performing knowledge fusion encounters a number of problems. The origins of these problems are the following:

1. Inaccuracy of existing information extraction algorithms leads to appearance of incorrect annotations.
2. Information contained on the web pages can be imprecise, incomplete or vague.
3. Multiple sources can contradict each other.

Thus, in order to perform large-scale automatic annotation it is necessary to implement a knowledge fusion procedure, which is able to deal with these problems.

Existing studies, which deal with the fusion issue, are either focused on solving separate subtasks of the problem or are only limited to a specific domain. The goal of this project is to make a contribution into the Semantic Web research by proposing a generic fusion framework based, which will make possible combining existing methods in order to perform knowledge fusion in a domain-independent way.

Table of contents

Introduction

The Web already contains a huge amount of information: according to 2003 estimates, the size of fixed web pages was 167 TB and database-driven web pages 91.850TB[1]. This information is mainly intended for human users and stored as hypertext so in a sense the Web now represents a large book. This information, however, is not machine-readable. The idea of the Semantic Web initiative is to make this information available for processing by software applications. In order to do that the data, which now is contained inside texts written in natural languages, should be stored in a strict formal machine-oriented language. Such a language, developed by W3C consortium, is RDF[2]. RDF allows information to be organised into a graph where nodes are entities and arcs are relations between them. In this way semantic information is presented explicitly. Different pieces of this graph are territorially distributed between web-servers but linked together using references. This representation format will transfer the Web into a huge database, which will make processing of information by software agents much more efficient.

RDF provides means to represent entities and relations between them in a formal way. However, in order to interpret this data we need meta-information, which explains types of objects and meanings of relations and thus provides a way to interpret these entities and relations. In analogy to databases we can divide the data itself, which describes actual objects and relations between them, and meta-data, which describes the data model. These models of data are called ontologies and are also represented in RDF-based languages, the most popular of which is OWL[3]. The factual data organised according to ontologies constitute a knowledge base.

But in order to construct such a globally distributed database, the necessary precondition of the Semantic Web initiative is the availability of semantic data. Information, which at the moment is intended for human users, must be translated into a machine-readable format (RDF). Such a translation process is called semantic annotation. The amount of information on the Web makes it impossible to solve the annotation task manually. So the usage of automatic information extraction algorithms is essential. These algorithms use various NLP and machine learning techniques to extract information from text (Ciravegna 2003; Cimiano 2005). The information extracted from different sources must then be integrated in a knowledge base, so that it can be queried in a uniform way. This integration process is called knowledge fusion.

There can be different scenarios when fusion process can help to work with information in a more efficient way than just giving all relevant information from separate sources. For instance, news taken from the sources with different point of view can be contradicting (e.g., information about the same event from Iran and Israeli web sites). In that case it may be convenient not just give the list of all news articles about the same event but also group them together according to their point of view and rank them according to estimated reliability of sources. In another situation, information from different web sites can be overlapping. In that case it makes sense to avoid redundancy and present to the user only those pieces of information which complement each other and not store twice those which are repeating. All these scenarios justify the importance of knowledge fusion.

[1] http://www.sims.berkeley.edu:8000/research/projects/how-much-info-2003/internet.htm
[2] http://www.w3.org/rdf/
[3] http://www.w3.org/2004/OWL/

For the knowledge fusion problem the following research areas are important. The first area of relevance includes existing information extraction algorithms. We need to understand of their work to make an assessment of the problems, which are likely to occur at the fusion stage. This overview will be given in the section 1 of the literature review. The next area, which is going to be considered, is knowledge fusion itself. Different communities developed methods to deal with aspects relevant to knowledge fusion. These methods have to be considered in more detail to evaluate existing techniques and their advantages and disadvantages. These techniques will be reviewed in section 2 of the literature review.

Literature review

1. Semantic annotations

A necessary precondition for the Semantic Web is the availability of semantic data. The challenge is to find an efficient way to provide the semantically annotated data. The main problem with semantic annotation is abundance of existing data in a variety of forms. Mainly information is represented as human-readable hypertext but multimedia sources are also important. The amount of work needed for annotation is too huge to be performed by human users in a reasonable time. So historically the studies in the area of semantic annotation tried to gradually minimize the human involvement and automate the process of annotation. The study (Ciravegna and Chapman 2005) overviews the requirements, which automatic information extraction algorithms must satisfy in order to be applicable on the Web scale. These include:

1. Ability to work on a large scale.
   Obviously this requirement is necessary because of abundance of information on the Web.
2. Ability to cope with heterogeneity.
   Heterogeneity here refers not only to different formats of documents but also to different degrees of structure uniformity among documents in the same format (e.g., free text vs tables/lists).
3. Sufficient quality.
   Quality of the algorithm means an obvious requirement that errors must be minimised. Both precision and recall are important, because low recall can lead to missing important pieces of data while low precision can introduce noisy annotations, which will create incorrect statements in a knowledge base.

At the current stage of research we are mostly interested in textual sources so this review will consider only information extraction from text. The literature on multimedia annotation exists and will be reviewed later (Benitez and Chang 2002; Hollink, Schreiber et al. 2003). We can distinguish following approaches to the problem of information extraction from text:

1. Manual annotations.
2. Supervised annotations.
3. Unsupervised annotations.

In the subsections we will discuss each approach in more detail and evaluate them with respect to the requirements listed above.

1.1 Manual annotations

Manual annotation was the first proposed and the most straightforward approach to semantic annotation. Its main advantage is its reliability: it is assumed that the human user's judgement about the meaning of text is the more correct one relative to any NLP algorithm. This assumption is even more justified when the author personally provides annotation to his/her data.

Different frameworks exist which allow the user to annotate materials available on the web. One such framework is Annotea (Koivunen and Swick 2003). Annotea was created primarily to support collaborative work of geographically distributed collectives. The main feature of the framework is that annotations created within Annotea are meant to be used by other human users rather than by software

agents. So strictly speaking these efforts are not aimed at performing the Semantic Web annotations. The same can be said about initiatives which propose manual annotation of resources meant for other users within the community (like http://del.icio.us).

Another annotation framework is CREAM (Handschuh and Staab 2003). Manual annotation in CREAM is meant to provide support to the authors who wish to annotate their data. CREAM distinguishes between shallow annotation (annotation of static web pages) and deep annotation (annotation of databases producing dynamic web pages). While relying on manual annotation, however, CREAM was improved to include the Amilcare information extraction module, which is able to learn from the manually annotated data and provide new annotations automatically (see below).

Because human users can better understand information given in a natural language and represented in HTML, manual annotation algorithms are accurate and flexible with respect to heterogeneity in comparison with other approaches. The main disadvantage of the manual annotation process is the effort required from the human users. This requirement violates the necessary condition for information extraction algorithms to be able to work on a large scale. So currently manual annotation tools are used rather to annotate a set of training examples for machine learning algorithms than for actual large-scale annotation. Therefore they are of limited interest for our work and are only briefly mentioned.

1.2 Supervised annotations

Supervised annotation is one step ahead of the manual annotation because it requires only limited human effort to teach the learning algorithm. After that, at least theoretically, the algorithm would be able to annotate new documents without human support.

Wrappers are one kind of supervised learning tools designed to annotate similarly structured documents such as online news stories. There are two kinds of wrappers: those, which require the user to specify extraction rules manually and those which support learning. An example of a wrapper system of the first type is Lixto Visual Wrapper (Baumgartner, Eichholz et al. 2003). Lixto provides a framework for the user to generate wrappers according to the structure of the documents in which he or she is interested. The first step requires from the user to select the pieces of documents, which contain the desired information. Then the user has to specify the conditions of the rule (e.g., that the value of a country's GDP is provided after the string token "GDP:"). Based on these conditions the system creates pattern rules, which are used later to extract information from other documents, which share the same structure. Other wrappers require the user to annotate several examples and then use machine learning techniques to create pattern rules. Obviously, a disadvantage of wrappers is their strict dependence on the documents' structure (including the HTML code), which actually makes them applicable to only one web site. In a case when the structure of the web site changes, the wrapper can become obsolete. So in this sense they do not satisfy the requirement of heterogeneity and still cannot work on a sufficiently large scale.

In order to cope with these difficulties it was proposed to make the extraction rules more flexible so that they can work in heterogeneous environments and adapt to different document types, genres etc. To achieve this it became necessary to employ techniques from NLP domain (wrappers use only statistical machine learning). An example of the system which uses this approach is Amilcare (Ciravegna 2003; Ciravegna and Wilks 2003). Amilcare requires an annotated corpus in order to learn

the extraction rules. The learning is performed using $(LP)^2$ algorithm, which is able to decide the best strategy for each type of document. The strategies vary with regard to the level of linguistic competence involved. Thus for well-structured documents the algorithm should come up with more rigid and simple rules while for the documents where the structure is not so strict the rules are more general and include more linguistic information in their conditions (e.g., the lexical category of a word or the type of a named entity). This can increase the quality of information retrieval.

The main disadvantage of supervised learning methods remains their dependence on the manually annotated training corpus. First, it requires a reasonable amount of human effort to annotate the corpus and second, the learning algorithm may have difficulties when scaling to the domains or genres which didn't appear in the training set. This means that the problems of coping with scale and heterogeneity are not solved yet, while the accuracy becomes lower than when using manual algorithms. Currently such supervised algorithms are used mostly as auxiliary tools to support the manual annotation process. For example, Amilcare is used as a part of the Melita annotation tool. In Melita the user starts with manual annotation, Amilcare learns from these annotations and later proposes new annotations to the user.

1.3 Unsupervised annotations

The latest approaches to semantic annotation problem try to extract information in a completely unsupervised way. It is argued that this can overcome the problems caused by the necessity to have a manually annotated corpus. At present most unsupervised annotation algorithms are based on pattern extraction rules (Hearst 1992). These are general patterns like "<instance name> is a <class name>" or "<class name> such as <individual names>". Obviously, such patterns are not likely to appear in each document which contains the facts to be annotated. Currently various studies try to improve the information extraction performance using various learning techniques. Some examples of such studies are given below. The most popular technique used by different systems is bootstrapping learning. In this approach the set of facts retrieved by the basic rules is used as a training set to learn new rules. In contrast to the supervised methods, which rely on the training set provided by the user, the patterns do not depend on the particular limited set of documents, which makes this approach more domain and genre-independent and thus more capable to cope with heterogeneity. The following are the tools, which are based on this technique.

One of the tools using this approach is KnowItAll (Etzioni, Cafarella et al. 2005). The system retrieves from a search engine a set of documents containing the seed instances extracted by the basic rules. Its rule learning routine tries to recognise the frequent sets of co-occurring context words and compose new extraction rules from them. Another techniques used by the KnowItAll tool are Subclass Extraction and List Extraction. Subclass Extraction tries to exploit the hypernym-hyponym relations extracted from the WordNet (e.g., instead of using just the rule "scientists such as…" use also patterns like "physicists such as…", "biologists such as..." etc.). List Extraction looks for the lists of entities in the documents. If some elements in the lists are established instances, then it is probable that other elements also represent the instances of the same type.

Another tool, which uses pattern rule extraction, is C-PANKOW (Cimiano, Ladwig et al. 2005). Its algorithm tries to approach the problem from the opposite direction: instead of finding the instances for each class it tries to find an appropriate class for each candidate instance. The algorithm deals with the possibility that

instances can belong to different concepts in different contexts and tries to assign the correct concept to each candidate instance. The algorithm performs it using the set of documents returned by the Google search in response to a query containing a basic rule initialised with the candidate entity's name (e.g., for the instance "Seville" the query will look like "such as Seville"). Thus the system gets the set of possible class names. Then the algorithm selects a subset of documents similar to the document being analysed (the cosine similarity measure is used). The candidate instance is then assigned to the class obtained from these similar documents.

Armadillo (Ciravegna, Chapman et al. 2004) also uses iterative learning approach. However, here the seed instances needed to initiate learning are taken not from language patterns but from the initial lexicon. This initial lexicon is filled using "weak" domain-dependent strategies (e.g., for the task of finding researcher's names these include searching well-known sites such as www.citeseer.com or www.informatik.uni-trier.de/~key/db/). This technique gives more correct seed instances, however it is more domain-dependent. Further instances are located using iterative training.

As we can see with regard to our requirements, the goals of developing unsupervised learning algorithms is to make information extraction less domain-dependent (and solve the problem of heterogeneity) and requiring less human effort (and improve the capability to work on the large scale). However, at the same time these advantages are compensated by lower accuracy of these algorithms because automatic choice of seed instances for learning is also not a trivial process and can introduce errors. Later on the base of these errors in the training data the algorithm may learn incorrect extraction rules.

1.4 Summary

As we can see, there exists a tendency towards minimizing the human effort in the annotation process. Thus it is possible to assume that in future the majority of semantic annotations will be extracted by unsupervised algorithms. However, unsupervised algorithms still suffer from problems, the chief of them is their low accuracy. It is hard to compare the level of precision and recall for different existing algorithms since no common testbed exists. However, the results provided in cited works allow us to estimate the accuracy of existing supervised learning algorithms as about 0.8-0.9 (F-measure). With unsupervised learning algorithms choosing seed examples for training is also a non-trivial process, which may introduce errors. This inaccuracy may introduce noise to annotated data in addition to the defects caused by incorrect information extracted from unreliable sources. It means that we cannot be fully confident about the correctness of annotations, which an algorithm produces. This uncertainty can be reasoned about if annotations are represented using a language, which supports uncertainty. The formalisms, which represent uncertainty, and ontological languages based on them are discussed in more detail in sections 3.3 and 3.4. Errors of the extraction algorithms can cause inconsistencies in extracted data, which we can attempt to resolve if we have information from different independent sources. Resolving these issues becomes a challenge of the fusion process.

2. Knowledge fusion

The problems of information integration have been considered separately in various fields of application such as database aggregation (Cochinwala, Dalal et al. 2001), expert opinion pooling (Dubois, Grabish et al. 2001), sensor fusion (Grabish

and Prade 2001) etc. However, common traits of these problems led to the emergence of a distinct research community, which addressed the integration issues in general. The term "knowledge fusion" was introduced to describe the task of integrating information from different sources. An overview of the knowledge fusion problem and methods developed within the community is provided in (Appriou, Ayoun et al. 2001). According to the authors, knowledge fusion is defined as "cojoining or merging information that stems from different sources and exploiting that cojoined or merged information in various tasks such as answering questions, making decisions, numerical estimation, etc." The task of knowledge fusion is a non-trivial one and presents a research challenge because heterogeneity of information and extraction errors in practice often lead to problems. Information gathered from different sources contains particular defective aspects. The authors (Appriou, Ayoun et al. 2001) distinguish the following aspects:

1. Ambiguity.
   In general, information which can be interpreted in several distinct ways is considered ambiguous. For example, it applies to the ambiguous identity case when it is impossible to decide what real-world item the information refers to.
2. Uncertainty.
   Uncertainty refers to the case when it is not possible to say definitely whether a particular Boolean statement is true or false.
3. Imprecision.
   Sometimes the content of the statement can be imprecise itself. For example, it is possible that a statement contains a rounded number instead of the precise one.
4. Incompleteness.
   In most cases an information source does not contain full information about the real-world item it describes. Some properties can be missing from the description.
5. Vagueness.
   Sometimes predicate of a statement can be represented by a vague term, for example, "high", "young", "fast" etc.
6. Inconsistency.
   Inconsistency occurs when several sources give a set of mutually contradicting statements.

All these issues are connected to each other so none of them can really be treated completely in isolation from others. For instance, ambiguity, when it is unknown whether two descriptions are about the same entity or not, occurs in the case when the descriptions are similar but at the same time contain some contradictions. If in this case it is impossible to make a decision with a 100% degree of confidence then the issue of uncertainty is introduced and so on. Existing approaches found in the literature do not address all aspects mentioned above with the same degree of attention but rather focus on some particular issues. Based on the studied literature we were able to identify two distinct types of approaches to fusion-related problems. One, developed within the knowledge fusion community, is based on formal logic-oriented techniques. Another, which has emerged in the database community, was later adapted by Semantic Web researchers and prefers data-driven techniques. These existing approaches will be discussed below.

Vagueness and uncertainty issues require that the chosen data representation language is able to store the information about degrees of confidence. There are

existing mathematical theories designed to deal with uncertainty and vagueness. There are studies which incorporate these formalisms into ontological languages. These studies will be considered in this review.

In the Semantic Web community the knowledge fusion problem consists of two subproblems, which so far have mostly been addressed separately. One is ontology integration, which corresponds to the generic knowledge fusion task. The second is instance fusion problem (factual data fusion). At the current stage of the project we are focused on instance fusion. Ontology integration is planned to be addressed in future.

The following subsections are organized in the following way: first, we will provide an overview of formal approaches developed within the knowledge fusion community. Then, we will consider the qualitative methods. After that we will review studies dedicated to the problems of representation of vagueness and uncertainty in ontological languages.

2.1 Formal methods

The methods developed within the knowledge fusion community consider domain-independent approaches to the problem and thus use information representation based on formal logic (in particular, first-order predicate logic).

One of the approaches deals with the issue of inconsistency resolution (Gregoire 2006). It is based on the usage of semaphores – additional propositions, which describe the fact that only one of the conflicting statements is true. For instance, two conflicting formulas f_1 and f_2 are represented as $f_1 \vee \neg Sem_1$ and $f_2 \vee \neg Sem_2 = \neg f_1 \vee \neg Sem_2$. In this way two formulas can be stored in the fused knowledge base simultaneously without contradiction. If after that the resulting knowledge base will be fused with the third information source containing f_1 then it will be possible to assert that f_1 is true and deduce $\neg Sem_2$. This approach is strictly formal and allows conflicts to be resolved while remaining within the limits of Boolean logic. However, the flexibility of such an approach in representing different degrees of confidence (e.g., preferences about the reliability of sources) seems limited. In the domain of semantic annotations, as was discussed earlier, the uncertainty issue occurs inevitably because of inaccuracy of the extraction algorithms. This makes the described approach inapplicable for our case.

Another study, which constructs a formal model for fusion but also deals with uncertainty is discussed in (Cholvy 2004). It uses as a guideline the STANAG 2022 standard – NATO recommendations for evaluation of information confidence. This standard distinguishes 6 levels of reliability for information sources (A-"completely reliable" to E- "unreliable" and F-"reliability cannot be judged"). The three main notions used to evaluate the information are the number of sources that support a piece of information, their reliability and conflicts between pieces of information. When the fusion algorithm receives a new piece of information and the reliability of its source (between A and F) it checks already stored information for potential conflicts and assigns the credibility value of the new piece (between 1 and 6) according to pre-defined rules (e.g., "If a fact F is not yet stored in DB then store it with the evaluation equal to <<r>, 1>"), where <r> is the reliability of the source. Evaluation is equal to 1 if the information is not conflicting with other items and 2 otherwise. The conflicts are resolved when the systems receives a user's query. In that case it calculates the more reliable piece of information using a weighted majority vote between sources and discards all information inconsistent with it. As we can see,

this approach tries to resolve inconsistencies dealing with different degrees of uncertainty. However, the usage of discrete values to measure confidence still limits its flexibility. Also the way of resolving conflicts (discard all information except the most probable) is not appropriate for the Semantic Web domain.

The study described in (Hunter and Liu 2006) proposes to overcome this limitation and resolve inconsistencies using such well-developed formalisms for uncertainty representation as probability theory and possibility theory. This work focuses on merging structured news reports in XML format and provides formal ways of representing inconsistent information and fusing it. A degree of uncertainty can be assigned to any piece of information in the form of a probability or possibility distribution. Then the study proposes rules for merging these pieces of information and updating their degrees of uncertainty even if it is measured using different formalisms (e.g., one piece of information uses probability theory and another one possibility theory). Such an approach allows handling inconsistency and uncertainty problems in a straightforward and well-defined way. It does not discuss, however, the question of how different degrees of uncertainty are assigned in the first place. Also the authors only consider nominal attributes in their research.

Another study from the same research group (Hunter 2006) suggests a method to evaluate the importance of inconsistencies. First, it proposes using four-valued Belnap logic, which allows one of four values to be assigned: “true”, “false”, “unknown” or “both”. Then, it introduces two metrics to measure the degree of inconsistency within the knowledge base: concordance and coherence. And, finally, to evaluate the relative significance of different incoherent knowledge base parts the authors propose assigning different weights to different subsets of knowledge bases using domain knowledge. All these metrics then can be used to decide whether to ignore inconsistencies, resolve them or reject them based on a pre-defined threshold. Such quantitative measure of inconsistency can be used to measure the reliability of the sources: the source which provides more information that contradicts with other sources should be less reliable. In this way this study can be seen as complementary to the one discussed before.

As we can see, these studies mostly deal with resolving conflicts rather than identifying them. This is understandable because they all use strict formal languages to represent information, which allows unique identification of all objects mentioned in the knowledge bases. It means that the problem of ambiguity needs to be addressed using different kinds of methods.

2.2 Quantitative methods

Quantitative methods were initially developed within the database community where it was important to detect duplicate records when merging different databases. Thus, the main issue which these methods addressed was ambiguity. The method of duplicate detection based on measuring the distance between records was developed within the database research community (e.g., see (Cochinwala, Dalal et al. 2001)). The distance between records in this approach was calculated as a weighted average of distances between attributes. Later this method was adopted by the Semantic Web community. The main distinctive feature of the Semantic Web domain as opposed to the database domain is the issue of incompleteness (Guha and Garg 2004). Therefore quantitative algorithms developed within the Semantic Web community attempt to resolve the ambiguity problem given that not all information is available. One of the studies, which adopts the distance measurement approach for the instance fusion task, is (Guha and Garg 2004). It handles one particular case of the instance ambiguity

problem and tries to determine whether two instances representing people with the same names in fact refer to the same person or to different people. The algorithm described in the paper makes its decision based on the distance between instances' properties. The distance calculation process is straightforward. All property values are treated as strings. The distance between the values of the same property is assigned equal to 0 if the values are equal and 1 otherwise. The distance between instances is calculated as the weighted average of distances between properties where weights of properties are assigned using domain knowledge. The main disadvantage of this method is its indifference to various types of data: all attributes are treated as nominal.

A similar problem of the entity's identity is discussed in (Guha 2004). The study investigates the ability to identify information about the same entity from two sources based on a set of attributes constituting a key (similar to databases). All attributes in this case are also considered as nominal. The problem with determining key attributes is the fact that although a majority of objects need only a few attributes to be identified uniquely (e.g., in most cases a city's name is enough to identify it), a small number of entities require a long key (e.g., "Concord, California, USA" and "Concord, New Hampshire, USA"). So the concept of Discriminant Description (DD) was introduced, which meant a set of attributes that uniquely identify one particular entity from others. For instance, for New Orleans it will be just the name, while for Concord it will be name, state and country. After that for two datasets (world cities and IBM employees) the size of minimal DD was investigated. It was found that in most cases minimal DD had to include not more than 2 attributes and that minimal DD allowed more than 95% of entities to be identified correctly. However, the problem of incompleteness still remains, because it is not guaranteed that any source contains all attributes needed for DD.

An extension of the approach based on similarity calculation is discussed in (Doan, Lu et al. 2003). In this approach candidate pairs of instances, which are likely to describe the same entity, are selected based on the similarity calculation. But afterwards the domain knowledge in the form of profiles is applied to make a decision. The profile of a class of items contains a set of constraints, which describe the typical item of that class. An interesting idea is that constraints are imposed not only on separate properties, but also on combination of properties. For instance, when one piece of information describes a person with the name "Mike Smith" and age 9 and another one – a person with the name "Mike Smith" and salary $200000 then such a combination will be considered improbable by the profile if it contains the information that the salary $200000 is unlikely to be combined with the age 9. These constraints are applied to the candidate pairs and used to make a final decision about whether two instances in the pair are about the same instance or not. This method is particularly interesting because it applies domain knowledge to identify conflicting pieces of information. However, at this moment popular ontologies do not contain this profiling information.

Thus, quantitative approaches found in the literature were designed to address the ambiguity issue. The main task of these algorithms is to identify the pieces of information which describe the same real world entities. These approaches do not handle the question of how these pieces of information should be merged together. In this respect we can say that quantitative approaches and formal approaches complement each other.

2.3 Fuzzy ontologies

This subsection and the following one discuss existing studies dealing with representation of vagueness and uncertainty in ontological languages. Currently popular ontology languages (like OWL) do not deal with these notions. However, vagueness and uncertainty are common phenomena when dealing with real-world data and both are included into the set of defective aspects of information, which complicate the fusion process. Uncertainty during fusion originates from unreliable sources, inaccurate extraction algorithms and using contradictory data. Thus, it is necessary that information representation language used to store facts in a knowledge base is capable of representing uncertainty. At the moment two types of formalisms are used as a foundation to introduce uncertainty and vagueness into ontology languages. These are probability theory and fuzzy logic. The implementations of both approaches are discussed in more detail below.

This subsection mentions studies which propose extending ontological languages based on fuzzy logic (Novak 1989). Initially fuzzy logic was designed to deal with the problem of vagueness. However, later it was adapted to reason about different degrees of confidence.

One of the implementations of a fuzzy ontology was described in (Huang, Jian et al. 2005). The work is aimed at automatic summarization of weather forecast news in Chinese. The information retrieval system receives as its input a set of news messages and returns the summary of the weather forecast as a result. The forecast summarises each news article separately without combining them. The algorithm extracts terms from the input documents and calculates values of their membership functions with regard to ontology classes. Then fuzzy inference is used to produce summary sentences (e.g., "The typhoon will attack Taiwan") and rank them according to their membership function values. Then defuzzyfication is performed and only the most reliable sentences remain in the resulting set. Here, as we can see, fuzzy logic theory plays only an auxiliary role. Information is represented in fuzzy terms only during processing. To maintain a persistent knowledge base in which information can be changed, added and removed, however, it is necessary to have an ontological language able to represent fuzzy instances and facts.

The work provided in (Stoilos, Stamou et al. 2005) describes Fuzzy OWL – a modification of the standard ontology language OWL, which is based on fuzzy logic. The authors introduced the fuzzy variant of description logic – f-$\mathcal{SHOIN}$. This fuzzy DL was used as a foundation for Fuzzy OWL. The set of OWL axioms was translated from the basic DL into fuzzy DL thus providing the basis for Fuzzy OWL. The facts written for a fuzzy KB should be extended with the value of the fuzzy membership function showing the degree to which an instance or a fact belongs to its class/relation. With a fuzzy knowledge base it becomes possible to specify the desired degree of confidence during querying. While in the ordinary OWL it is impossible to receive a result for a complex query if the result does not satisfy at least one of the query's conditions, for a fuzzy knowledge base it is possible not only to specify the minimal degree of confidence for the resulting facts but also to rank the results according to their membership function value. This possibility makes Fuzzy OWL very relevant for the systems which must integrate information (possibly contradictory) from different sources.

2.4 Probabilistic ontologies

Current work on probabilistic ontology languages is mainly based on Bayesian Networks. One of the studies (Mitra, Noy et al. 2004) uses probability theory to

develop an algorithm which maps similar concepts from two different ontologies. Mappings link nodes of ontologies, which are supposed to refer to the same real-world notions. Thus, this is a kind of fusion problem. Both ontologies, which serve as an input, are precise. However, the mappings generated by the algorithm are probabilistic. These probabilistic mappings are organized into a Bayesian Net graph. The nodes that represent mappings between pairs of concepts, which in their turn are related with each other (e.g., a pair of ancestor nodes and a pair of child nodes), are linked by arcs in the Bayesian Net graph. Thus the probability of similarity between two classes is influenced by the probability of similarity between their ancestors, siblings and even the domains of the properties for which these classes are ranges. The probability of two classes from different ontologies being similar is measured after calculation of probabilities in the Bayesian Net graph. As we can see this approach is limited: it serves only as a technique for ontology merging and does not allow any uncertain information to be represented.

A study described in (Ding and Peng 2004) tries to extend the standard ontology language OWL. The work focuses on the ontology itself (i.e., DL TBox) instead of individuals in the knowledge base (i.e., DL ABox). The OWL modification presented in the paper allows us to define both *a priori* probabilities (e.g., probability that any object belongs to a class "Animal") and conditional probabilities (e.g., that any "Animal" is a "Male"). The resulting ontology can be translated into a Bayesian Net and be used to support such ontology-related reasoning tasks as the following:

1. Concept satisfiability (given a query, check if the resulting probability of its conditions is non-zero).
2. Concept overlapping (a probability of overlapping between individuals of two classes).
3. Concept subsumption (the most similar class to a query).

Such capabilities are of course useful but they do not provide means to actually represent uncertain data. The authors defer this to future work.

However, the work described in (da Costa, Laskey et al. 2005) allows the representation of probabilistic information and reasoning about it. The work is based on Multi-Entity Bayesian Networks (MEBN) theory, which provides full first-order expressiveness. Based on this theory the authors construct their modification of ontology language OWL – PR-OWL. This language is able to represent probabilistic data and to reason about it, which makes it a good candidate for use within the knowledge fusion framework.

2.5 Summary

Incompleteness, imprecision and vagueness are inherent features of data sources in the Semantic Web domain (web sites). Using multiple sources for fusion is supposed to help with these problems. For instance, in case of incompleteness the data missing in one source can be added from another, when one source contains imprecise or vague value it is possible that there is another one containing precise one, etc. In contrast, ambiguity and inconsistency (and uncertainty caused by them) are problems caused by the use of multiple sources and thus are the problems of the fusion step itself. So we can try to overcome them by improving the fusion algorithm. Two kinds of approaches found in the literature focus on different problems: formal methods from the knowledge fusion community specialise in resolving inconsistencies while quantitative approaches from the database and Semantic Web communities are aimed at identifying ambiguities. Thus, it makes sense to combine the two kinds of approaches within a common fusion framework and divide tasks between them. It is

often not possible to resolve ambiguity or inconsistency and be 100% confident about correctness of the decision. Thus resolving both kinds of problems can in turn add more uncertainty. Therefore representing uncertainty in the knowledge base is necessary because it allows the data to serve as both input and output for both kinds of algorithms. The algorithms will change the degrees of confidence of the input information.

In order to represent different kinds of uncertainty, fuzzy logic and probability theory were designed. Fuzzy sets usually represent vagueness (e.g., quantitative measures of such terms as "long", "short", "many" etc.). On the probabilities there are different views (Korb 2004). While classical (or frequentist) interpretation (the ratio of favourable cases to total equipossible cases) is inappropriate for representing uncertainty in knowledge bases, the subjectivist (or Bayesian) interpretation can be used. Probability is understood as the degree of belief of an ideal rational agent about hypotheses for which the truth-value is unknown. Such a view is well suited when we deal with automatically extracted annotations because we cannot be sure that the value retrieved by the algorithm is correct. So from the point of view of interpretation probabilistic approach seems more suitable. However, probability theory seems less flexible from the practical point of view. In most cases we cannot know all possible values of a property in all possible sources and therefore cannot assign the probability distribution so that $\sum p = 1$. Also we cannot be sure in all cases that different values are mutually exclusive because some properties can have multiple values. Using fuzzy logic gives more flexibility but it can cause problems when different kinds of problems appear together. For instance, it is not clear what to do if both vagueness and uncertainty are present.

To overcome these liabilities there exists another approach – possibility theory (Zadeh 1999), which was specially designed to deal with the problems of confidence. However, no mention of an existing ontological language extension based on the possibility theory was found in the literature. Probably, implementing such an extension would help to solve the problems of representing uncertainty but, since developing such a method is beyond the scope of this work, the project has to be based on existing implementations using either a fuzzy or a Bayesian approach.

3. Conclusion

In our review we considered two main topics: an overview of information extraction approaches, which construct semantic annotations from text and so provide input for knowledge fusion, and the issues of the knowledge fusion process itself. We established that the problems of knowledge fusion are caused by certain defective aspects of information, which in turn are caused by defects in the initial information and low accuracy of information extraction algorithms. We found two kinds of approaches, which deal with the fusion-related issues: quantitative approaches used for duplicate identification and formal approaches focused on resolving inconsistencies. In order to deal with uncertainty, the data format must support representing different degrees of confidences of pieces of information. Currently there are existing implementations of ontological languages able to represent uncertainty, which are based on fuzzy logic and probability theory.

As different approaches focus on different aspects of knowledge fusion-related problems, it makes sense to combine them in a common framework, which will perform fusion choosing an appropriate method depending on the kind of problem it encounters. The task of constructing such a framework has much in common with

problem solving methods in knowledge management (e.g., see (Motta 1999)). In order to design such a framework the following areas still need to be studied:

1. Semantic annotation area: multimedia annotation.
   By the moment we only considered information extraction from text. However, the Web contains also other kinds of information which must be translated into a machine-readable format. Because of this, study of these multimedia annotation algorithms is also required.
2. Knowledge fusion: integration of ontologies.
   In our review we only considered methods for knowledge base integration. However, in cases when information was annotated according to different domain ontologies we need to solve the problem of meta-data fusion as well. This is a well-established field of study in Semantic Web research community and proposed approaches need to be investigated in detail.
3. Problem solving methods in knowledge management.
   A framework for a fusion system must be as generic as possible. Generality in this case means independence from domain and task so the issues of modularity, reusability and distinction between generic and application-dependent parts are important. These issues are considered in knowledge management domain and must be studied in detail.
4. Semantic Web Services.
   From the point of view of implementation the system should consist of distributed modules in order to increase the maintenance efficiency, reusability and changeability of its components. Thus it makes sense to implement components as web-services. In order to do that the studies in the area of web services in Semantic Web must be overviewed.

Pilot study report

1. Motivation

The task of instance fusion can be divided into two main subtasks: detecting instances extracted from different sources, which potentially represent the same entities, and fusing them by combining their properties. So far these subtasks were mostly addressed separately by different research communities. Detecting matching instances was based on approaches developed within the database community while merging was mostly based on formal logic methods created by knowledge fusion community. In more details these topics are discussed in the literature review. In order to perform knowledge fusion these approaches should be used in combination.

Another issue is the applicability of these algorithms to the automatically extracted information and their capability to cope with noise in the input data caused by extraction errors. So it is important to experiment with combining these algorithms and try to perform complete instance fusion workflow. This pilot study was an attempt to study these issues and implement a model of the knowledge fusion workflow for a limited domain.

2. Experimental setup

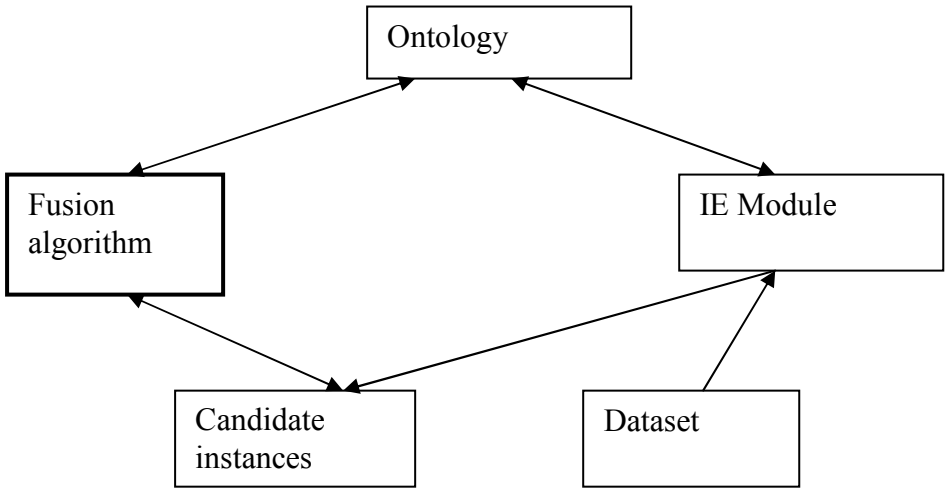

Figure 1 - Architecture of the pilot study framework

The main goals of this pilot study were:

1. To discover possible problems with fusing automatically annotated data.
2. To evaluate the applicability of quantitative matching instances detection algorithms for automatically extracted data.
3. To experiment with the parameters of the algorithm and evaluate their influence on the algorithm's performance.

The architecture of the experimental setup included the following components (Fig. 1):

1. Dataset.
   The dataset represents the corpus of documents containing information to produce candidate instances.
2. Ontology.

The ontology describes the domain and defines the structure of candidate instances.

3. Information extraction module.
   The information extraction module is necessary for two reasons. The first is the necessity to avoid manual construction of candidate instances. Also, the Semantic Web annotations in general case are going to be produced by automatic extraction algorithms. This is likely to cause the loss of quality, which can affect the fusion stage. It is useful, therefore, to model this loss of quality in the study.
4. Fusion algorithm.
   The fusion algorithm consists of two main parts responsible for two subtasks of instance fusion: matching instances detection and instance merging. Implementing and testing the algorithm was the main focus of this study.

Application of the knowledge fusion to the Semantic Web is aimed at dealing with semantic annotations automatically extracted from different web sites. Therefore as datasets for our study it was decided to use a set of web pages describing one particular domain (geography) and produce annotations from them using automatic extraction algorithms.

The ontology chosen for experiments had to describe the geographical domain with a level of detail sufficient for representing information from the selected web sites. There are several existing ontologies describing geographical data. An ontology specifically designed as a testbed for Semantic Web applications is TAP (Guha and McCool 2003). This ontology covers the geographical domain. However, it does not contain sufficient set of properties to represent all available information from the selected web sites. For instance, such free text properties as a country's culture and history description were not listed in the original ontology. Therefore it was necessary to extend an existing ontology with new properties so that it became possible to represent instances extracted from any document in the corpus. Resulting ontology is given in Appendix B.1.

Two tools of different types were selected as information extraction algorithms. First, it was Amilcare: an extractor developed in the Sheffield University (Ciravegna 2003). Another tool was Lixto Visual Wrapper (Baumgartner, Eichholz et al. 2003), which allowed the user to construct extraction rules manually. This tool was designed to extract information from web pages with a regular structure.

The fusion algorithm was based on the approach described in (Guha and Garg 2004). However, their basic algorithm uses simple comparison and treats all values as strings. The distance between attributes is set to 0 if two values are precisely the same and to 1 otherwise. Then these distances are aggregated by calculating weighted average and used to make a decision about whether two instances are referring to the same real-world entity. A potential disadvantage of this algorithm lays in the fact that it treats all types of data as strings only checking precise equality of them. While such an approach can work with nominal attributes (e.g., people's names or places of birth) it becomes inapplicable if attributes are quantitative (e.g., company's turnover) or contain large text descriptions (e.g., historical background of a country). Measuring similarity should be performed differently for different data types. Our fusion algorithm attempted to deal with this issue. In more detail it is discussed in the section 4.

3. Information extraction

As the test dataset it was decided to use various web sites containing geographic, economic and social data about countries. The main reasons for that were:

1. Variety of information types.

Different sites on the Web contain different information depending on the site's purpose. So the usage of country data for testing allows us to include into the experiments string data, free text data and numeric data.

2. Availability of information on the Web.

There are several information sources, which can be used to test different fusion techniques.

- *CIA World Fact Book*

  The CIA World Fact Book is the most popular resource containing general statistics about different countries. There exists a DAML representation of this data, which makes it convenient to use.

- *Traveldocs.com*

  This is another site providing information about various countries for US tourists. It contains a limited amount of statistical data overlapping with the CIA World Fact Book (e.g., economy) and also textual information overlapping with Lonelyplanet.com (e.g., culture&history). This makes Traveldocs data source interesting for testing of fusion techniques.

- *NationByNation.com*

  A site, which provides information about various countries, including mostly statistical data.

For the experiments it was decided to use the World Fact Book knowledge base as a benchmark. These instances were loaded into the knowledge base initially. Candidate instances from other sources were produced using information extraction tools. Two kinds of information extraction tools were chosen to be used. The first is the Amilcare (Ciravegna 2003). Amilcare is an information extraction tool based on machine learning techniques. Amilcare requires an annotated training set, from which it learns extraction rules. The training set consists of text documents into which tags were manually inserted (e.g., "GDP: <hasgdp>123 million</hasgdp>"). From these annotated documents Amilcare produces extraction rules, which after that are applied to new documents. The rules specify the conditions for insertion of a particular tag (e.g., that the opening tag <hasgdp> follows after the word "GDP:"). Conditions can be specified not only for particular words but also for word types (part of speech tags), specific data tokens (e.g., date or time), etc. Depending on the regularity of the document's structure, Amilcare adjusts its learning mechanism to employ natural language processing techniques to a greater degree. Thus, in order to use Amilcare, a part of the corpus had to be annotated manually. Then this part was provided to Amilcare as a training set to induce extraction rules. These rules were used to extract instances from the rest of the corpus.

The second approach was the usage of a wrapper induction tool, in particular, Lixto Visual Wrapper (Baumgartner, Eichholz et al. 2003). Lixto extracts information based on manually constructed rules. Each rule has in its conditions particular text tokens (e.g., "Extract from each document the token on a certain distance (in symbols) after the word "GDP""). Web pages in the chosen corpus (NationByNation) had some regularity in their structure (e.g., usually a property value was preceded by its name). This allowed us to construct induction rules for each property manually.

Amilcare was used to extract information from the Traveldocs corpus. The extraction was performed in the following way. 40% of documents (about 87

countries out 221) from the corpus were randomly selected and annotated manually. Then, to evaluate the performance of Amilcare extraction algorithm, half of these documents were used as a training set and the rest as a test set. The algorithm performed reasonably well, achieving for all tags the average F-measure of 0.75 (Table 1 contains results for economic attributes). It was found that the algorithm performed better at extracting numeric properties than strings (especially long strings). For example, the attribute "hasGDP" was extracted with an F-measure value of 0.92 while the free text "hasEconomyOverview" – only with 0.36. The most likely reason for that is the short length of numeric attributes, which allows Amilcare to induce context rules, which depend on the tags already inserted (e.g., that closing tag "</hasgdp> should be inserted one word after the opening tag <hasgdp>").

Table 1 – Attributes retrieved by Amilcare

Attribute name	F-measure
hasEconomyOverview	0.36
HasExports	0.97
hasExportPartnerships	0.61
HasGDP	0.92
hasGDPEstimationDate	1.0
hasRealGrowthRateGDP	0.85
hasRealGrowthRateGDPEstimationDate	1.0
HasPerCapitaGDP	0.96
hasPerCapitaGDPEstimationDate	1.0
HasImports	0.91
hasImportPartnerships	0.62
HasLabel	0.81
HasExternalDebt	0.83
hasExternalDebtEstimationDate	0.25
hasBudgetExpenditures	0.75
hasBudgetRevenues	1.0

The NationByNation corpus had more regular structure, which made it possible to use Lixto to extract information. In our tests we assigned to each property a degree of confidence and linked it to the source description. For Traveldocs we used F-measure as the confidence degree. For other sources it was selected arbitrarily to model the subjective belief in the source's reliability (1.0 for CIA and 0.9 for NationByNation). Later in experiments when comparing two properties with different degrees of confidence we chose to select a value with the highest degree of confidence (i.e., used fuzzy logic approach).

4. Implementation of the experimental tool

After the extraction of test corpora the next step was to perform the fusion experiments. An experimental tool was implemented in Java using the Eclipse development environment. The main functions of the tool were finding instances to be merged in the initial and candidate knowledge bases and after that merging these instances. The program included the following modules (Java packages):

1. instancecomparator.io:
   This package contains classes responsible for loading the data from the files. The data includes the domain ontology (in RDFS format), benchmark knowledge base and candidate knowledge bases (both in RDF, but

structured according to different ontologies). The data loaded from these files was used to populate internal structures defined in packages "instancecomparator.ontologydefinition" and "instancecomparator. knowledgebase".

2. instancecomparator.ontologydefinition:
   Contains classes, which describe the domain ontology (i.e., ontological classes, properties and datatypes).
3. instancecomparator.knowledgebase:
   Defines classes, which describe the knowledge base (i.e., instances of ontological classes and properties).
4. instancecomparator.knowledgebase.distancemeasurement:
   Describes routines used for comparing class and property instances, identifying matching instances and merging them.
5. instancecomparator.userinterface:
   Contains classes which describe the user interface. The user interface is built using Swing library.

The workflow for the experimental tool was the following (Fig. 2):

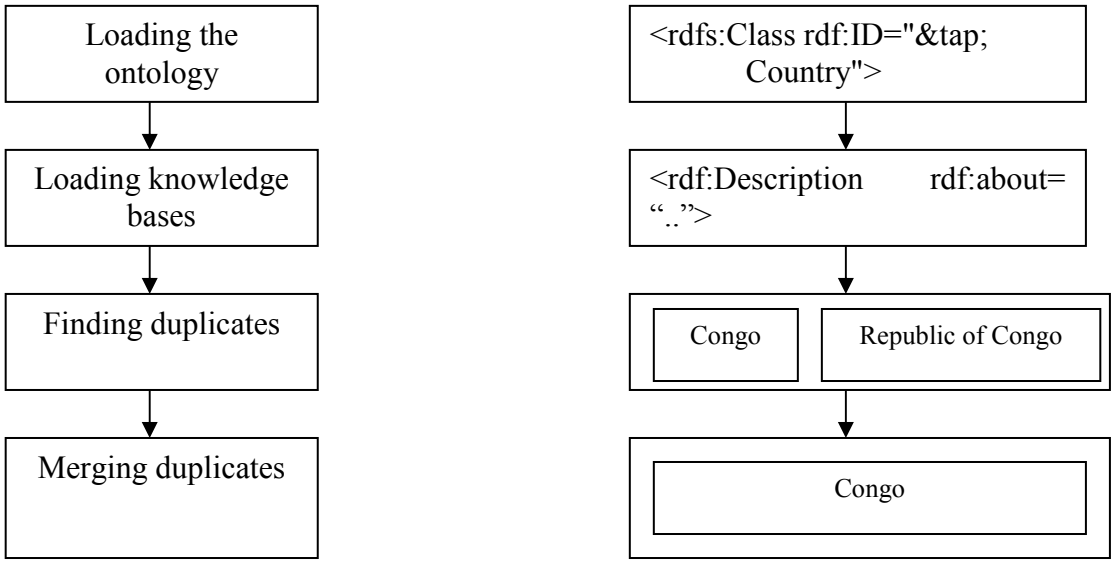

Figure 2 – Workflow of the experimental tool

4.1 Loading the ontology

The ontology used in our experiments was based on the TAP ontology and in particular on its subset which describes the class "Country" and its properties. This ontology was stored in RDFS format. The changes to the original TAP ontology were necessary because it didn't distinguish between different datatypes: the only datatype was "Literal", which was used to define all properties including numeric ones. One of the goals of the study was to experiment with different kinds of data, therefore it was necessary to distinguish between them. Thus, the property types' descriptions in the ontology were changed according to the actual data types. Loading the ontology was performed using Jena-2.3 RDF parser library.

4.2 Loading knowledge bases

The extraction algorithms used in our work did not produce their output in an ontological format (Amilcare produced plain text files and Lixto produced a set of XML documents). Therefore special wrappers were implemented to translate these files into RDF. Also, the data from CIA World Fact Book was structured according to its own ontology, while extracted data was annotated using the ontology described in the previous subsection. Therefore, in order to compare instances from two knowledge bases it was necessary to provide a mapping between two ontologies. A set of mapping rules, which specified how the properties from the benchmark knowledge base should be restructured according to the TAP ontology, was constructed manually (see Appendix B.3).

4.3 Identifying matching instances

Using data described above, we ran experiments in which series of parameters were varied. The basic algorithm involved calculation of distance between instances as an average of distances between values of each attribute. Other tests involved changing the parameters of this basic algorithm and comparing the results of experiments. The subsections explain these parameter changes in more detail.

4.3.1 Basic algorithm

The algorithm follows the approach described in (Guha and Garg 2004). The decision about whether two instances describe the same entity or not is made based on the semantic distance measurement between two instances. The distance is measured as a weighted average of distances between attributes. Each distance between attributes is calculated according to the type of the attribute. Our application had to deal with attributes of the following types:

– Continuous numeric:

The distance between two properties of this type is calculated as

$d = \frac{|v_1 - v_2|}{v_{\max} - v_{\min}}$, where

d – the distance between properties;

$v_{1,2}$ - the values of the property in two compared instances;

$v_{\min,\max}$ -minimal and maximal values of the property among the values contained in the knowledge base.

This is a normalised distance, which should put the resulting value into the interval between 0 and 1.

– Nominal:

There are two kinds of nominal properties in the knowledge base: functional (i.e. unique, like “haslabel”), which may only have one value, and non-functional (like “hasnaturalresources”), which may have multiple values. For functional properties the distance is calculated in a straightforward way: the distance is set to 0 if two values are the same and 1 otherwise. For non-functional properties the distance is calculated as cosine distance between sets:

$d = 1 - \frac{\sum_i x_{i1} x_{i2}}{\sqrt{\sum_i x_{i1}^2} \sqrt{\sum_i x_{i2}^2}}$, where

x_{ij} - number of occurrences of i-th value in j-th set (in our sets it could be 0 or 1 because sets didn't contain repeating values)

– Free text:

Free text properties are considered as sets of words. The distance between two sets is calculated using set similarity measures (e.g., cosine, overlap, or Jaccard).

From two compared instances pairs of attributes were selected, for which values were present in both instances, and then distances between them were calculated. After calculating the distances between each pair of attributes the distances were aggregated to get the distance between instances. Aggregated value was calculated as a weighted average of distances between attributes.

$d = \frac{w_i \cdot d_i}{\sum_i w_i}$, where

d – distance between instances;

d_i - distance between the i-th pair of attributes;

w_i -weight of the i-th attribute.

In the basic algorithm all weights were established equal to each other.

Pairs of instances recognised as possible matching instances were selected in the following way. A pair of instances, one from the candidate set and one from the initial set, with the smallest distance between them was considered to represent the same entity. Thus, the selection algorithm represents a variant of the k-NN machine learning algorithm with k=1. The described algorithm was used as a basic one. Then in order to improve its performance experiments with different heuristic techniques were performed. These techniques are described in the following subsections.

4.3.2 Using key field for identification

Each instance in our knowledge base represented a country. A unique identifier of a country is its name. So the most straightforward way to identify two matching instances is to compare their names. However, there are cases when the same country was identified in different knowledge bases using differently formatted names (e.g., "Palau" and "Palau islands", "Guadeloupe" and "French West-Indies", etc.). Thus, such identification still does not provide 100% accurate results in finding matching instances. Because this method is the simplest, in our experiments we used it as a baseline to compare against our algorithm. After comparison of the basic algorithm with a baseline, in order to estimate importance of other attributes for identification we then changed our basic algorithm and excluded the country name from the analysis. The last experiment we performed was a combination of two identification methods: first, the matching pairs of instances were looked for using their names. Then, for those candidate instances, which didn't have identical names in the initial database, the main algorithm, which included all fields into analysis, was used. Such a strategy (using complete algorithm only when key identification failed) prevented the cases when pairs of instances with the same name were judged as not matching because similarities of other attributes had more influence than identical names.

4.3.3 Different similarity measures for free text attributes

Some of the string attributes in the knowledge base contained long pieces of natural language text. It was not appropriate to compare these free text attributes in the same way as nominal values because it is possible that the meaning of two fragments of text is similar despite differences in phrasing. A more flexible way of comparison was to apply the approach used in natural language processing to find similarities between documents. A text was represented as a set of words. The similarity between two sets was calculated using set similarity measures. There are different existing set similarity measures, in particular cosine, overlap and Jaccard metrics.

The distance measures used in our tests were calculated as follows:

1. Cosine:

   $$d = 1 - \frac{\sum_i x_{i1} x_{i2}}{\sqrt{\sum_i x_{i1}^2} \sqrt{\sum_i x_{i2}^2}}$$, where

   x_{ij} - number of occurrences of i-th word from the word vector, containing words from two text values being compared, in the j-th text
2. Overlap

   $$d = 1 - \frac{\sum_i x_{i1} x_{i2}}{\min(\sum_i x_{i1}^2, \sum_i x_{i2}^2)}$$, where

   x_{ij} - number of occurrences of i-th word from the word vector in the j-th text
3. Jaccard

   $$d = 1 - \frac{\sum_i x_{i1} x_{i2}}{\sum_i x_{i1} + \sum_i x_{i2} - \sum_i x_{i1} x_{i2}}$$, where

   x_{ij} - number of occurrences of i-th word from the word vector in the j-th text

We ran experiments to determine whether the choice of particular metrics influenced the performance of the matching instances finding algorithm.

4.3.4 Pruning of possibly invalid numeric values

Due to the inaccuracies of extraction algorithms it was possible that some of the numeric values were extracted incorrectly because of misclassification (e.g., “hasPerCapitaGDP” instead of “hasGDP”) or incorrectly recognised format (e.g., “$613” instead of “$613 billion”). Such incorrect values led to incorrect distance calculation, which influenced distance calculation between instances and could distort the results of matching instances identification. It was important therefore to try to exclude such values from analysis. One of the possible ways to do it was to automatically discard all values which were too great or too small for a given property. In theory, restrictions on maximal and minimal possible values for a property should be taken from the domain ontology. However, if the ontology does not contain this information (like in our case) it may be helpful at least to employ some heuristics to discard these invalid values.

The simplest technique is to compare the attribute value of a new instance against maximal and minimal values of that attribute for instances already stored in the knowledge base. In our experiment the acceptance range for a new numeric value was set to $\frac{v_{\min}}{2} \le v \le 2v_{\max}$.

4.3.5 Excluding potentially non-meaningful words from free text attributes

A natural language text necessarily contains non-content words which do not depend on the topic of the text and do not provide any useful information for determining the semantic similarity between texts. Such words are likely to be found in any text and make it harder to calculate the similarity between texts – the words present in both sets make them more similar to each other. In order to avoid this it is helpful to exclude potentially non-meaningful words from texts before comparison. The problem is to determine these words. If a pre-defined list of such words is not available then a possible technique is to exclude all non-capitalised words. We assumed that capitalised words represent named entities and therefore all of them are meaningful. Obviously with such an approach many common words which also contributed to the meaning of the text were excluded as well. Thus, we tried to find out whether information gain from excluding non-meaningful common words compensated it or not.

4.3.6 Learning weights

The basic algorithm did not take into account different importance of attributes. For example, country name, which is a unique identifier for a country, and official language, which can be the same for different countries, had the same weights and were treated as equal when calculating distance between instances. In order to avoid this different attributes should receive different weights according to their relative importance. Ideally, such weights should come from domain knowledge but if the ontology does not contain them then it can be useful to obtain them using machine learning techniques. In particular, neural networks can be trained to obtain weights if we have training data. In this experimental tool we implemented a neural network training algorithm (a perceptron using Adaline learning rule (Widrow and Hoff 1960)), which learns relative importance of attributes using training data. We used a straightforward version of the algorithm. The network consisted of one layer of neurons and each neuron corresponded to one attribute type. All weights were initially set equal to each other (0.5). The input of each neuron was the distance between two attributes of the corresponding type. On each iteration a training example consisting of two instances for which we knew whether they were matching or not was passed to the network. The output of the network was the distance between instances and the error of the network was calculated in the following way:

$$e(n) = d(n) - \frac{\mathbf{w}(n)^T \mathbf{x}(n)}{|\mathbf{w}(n)|},$$

where

$e(n)$ - error of the network on n-th iteration;

$d(n)$ - actual value of the distance between instances (0 or 1);

$\mathbf{w}(n)$ - vector of neurons' weights;

$\mathbf{x}(n)$ - vector of distances between attributes.

After calculating the network's error the weights were updated:

$\mathbf{w}(n+1) = \mathbf{w}(n) + \eta e(n)\mathbf{x}(n)$,

where

η - learning rate parameter (set to in our experiment).

The goal was to evaluate utility of the neural network and check whether it improves the quality of matching instances detection or not.

4.4 Merging matching instances

After the algorithm found instances which probably describe the same entity the next step is to merge these instances. Merging information from different sources into one knowledge base allows us to store it in a more compact way. Main advantages of information merging are reducing the amount of stored information (redundant information repeated in both knowledge bases is stored only once) and the possibility of more convenient representation (information about the same entity from different sources can be viewed together). In our experimental tool the merging process was the following. Matching algorithm described in the previous subsection produced pairs of instances, which probably described the same instance and had to be merged. Then, it was possible for the user to deselect those pairs, which were suggested for merging erroneously. After the user's confirmation all selected pairs were merged. From each pair a new instance was produced and its properties were taken from two initial instances. The rules were the following:

a) If a property had values for only one instance then it was added to the resulting instance.

b) If a property had values in both instances and these values were similar then these values were merged.

c) If a property had values in both instances and these values were not similar then both were added to the resulting instance.

The definition of "similar" varied depending on the attribute type and was the following:

- Two nominal values were considered "similar" if they were equal or one was a substring of another;
- Two numeric values were considered "similar" if the difference between them was less than 5%.
- Two free text values were considered "similar" if all words from one value were contained in another (i.e. one was a subset of another).

In case when two property values were considered similar and merged the new value was set equal to the value of one of the initial properties, which came from a more reliable source. To model the situation where sources have different reliability we assigned it manually (see section 3). When information was extracted using Amilcare we assigned the reliability of each piece equal to the F-measure of the algorithm for each particular type of property so that values of different attributes in the set were given different confidence weights.

5. Results and evaluation

As was described in the previous sections, the fusion algorithm consisted of two major steps: matching instances detection and instance merging. These steps were evaluated separately. The results are discussed in the following subsections.

5.1 Matching instances detection evaluation

Table 2 – Results of the matching instances identification stage

Section		Traveldocs (extracted)	Traveldocs (train)	NationByNation (Lixto)	Traveldocs (extracted) & NationByNation
	Number of countries compared	134	89	178	134
4.3.1 - 4.3.2	Correctly identified:				
	- without label	91 (67.9%)	84 (94.4%)	18 (10.1%)	
	- with label	127 (94.8%)	88 (98.9%)	169 (95.5%)	99 (73.9%)
	- only label	118 (88%)	78 (87.6%)	160 (89.9%)	99 (73.9%)
	- first label, then others	133 (99.3%)	88 (98.9%)	171 (96.6%)	99 (73.9%)
4.3.3	Free text distance measures:				
	- cosine	127 (94.8%)	88 (98.9%)	169 (95.5%)	99 (73.9%)
	- overlap	102 (76.1%)			
	- Jaccard	110 (82%)			
4.3.4	Pruning of potentially invalid numeric values				
	- enabled	127 (94.8%)	88 (98.9%)	169 (95.5%)	
	- disabled	117 (87.3%)	87 (97.8%)	167 (94.4%)	
4.3.5	Free text attributes				
	- all words	127 (94.8%)	88 (98.9%)	169 (95.5%)	99 (73.9%)
	- only capitalised	120 (89.5%)	88 (98.9%)		
4.3.6	Using neural network for training weights	130 (97%)	88 (98.9%)	169 (95.5%)	99 (73.9%)

Table 2 summarises the results of experiments we ran to estimate the performance of the instance matching algorithm. In each experiment we compared instances from two knowledge bases. The initial knowledge base in each case was the benchmark CIA World Fact Book knowledge base. Among other three knowledge bases two were extracted from the Traveldocs web site. One of them was extracted manually and then used as a training set for the Amilcare tool. It contained 40% of the whole Amilcare dataset (column "Traveldocs (train)"). Another knowledge base was extracted automatically using Amilcare from the remaining 60% of the dataset (column "Traveldocs (extracted)"). Experiments with the manually extracted knowledge base ("Traveldocs (train)") were done in order to evaluate the influence of errors produced by the automatic extraction algorithm on the fusion algorithm's performance. The third knowledge base was extracted from the NationByNation web site using Lixto Visual Wrapper tool. It was compared against the benchmark CIA World Fact Book (column "NationByNation (Lixto)") and against instances

automatically extracted from Traveldocs (column "Traveldocs (extracted) & NationByNation").

5.1.1 Basic algorithm vs only key values

As a baseline for our experiments we used the straightforward technique, in which each instance (country) was identified by its key field ("haslabel" attribute) and two instances were considered matching if they had the same name. Our experiments have shown that the fusion algorithm which used comparison of all attributes was able to outperform the baseline in all cases (94.8% vs 88%, 98.9% vs 87.6%, 95.5% vs 89.9% and 71.6% vs 68.7%). Other experiments were performed to evaluate the usability of techniques listed in sections 4.3.2-4.3.4. However, in some cases it gave incorrect results despite the fact that two instances had the same name if the combination of other attributes didn't match well. The best results were achieved when detection was performed in two steps: first, instances with the same name were selected as matching and then the complete algorithm was launched only for those instances which didn't have matching names (99.3%, 98.9%, 96.6% and 73.9%).

For all subsequent experiments the default parameters of the algorithm were the following:

- using whole set of properties for comparison;
- cosine distance measure;
- enabled pruning of potentially invalid values;
- considering all words in free text attributes;
- no weight learning.

In each experiment we changed one of these parameters but left all others intact.

5.1.2 Different similarity measures for free text attributes

Experiments with different distance measures for free text and nominal attributes showed that the choice of the set similarity measure influences the performance of the matching instances detection algorithm. The best performance was achieved with the cosine similarity metrics (94.8%). The difference in performance was substantial contrary to what was expected based on other studies (e.g., see (Lee, Pincombe et al. 2005)). These tests were not performed for the NationByNation corpus because NationByNationCorpus didn't contain any free text fields.

5.1.3 Pruning of possibly invalid numeric values

Pruning of the potentially erroneous numeric values in all cases had a positive influence on the algorithm's performance (94.8% vs 87.3%, 98.9% vs 97.8%, 95.5% vs 94.4%). As was expected, it didn't influence the case with the manually annotated dataset because the manually extracted knowledge base didn't contain erroneously extracted attributes.

5.1.4 Excluding potentially non-meaningful words from free text attributes

Using only capitalised words in the free text attributes comparison, however, didn't give any improvement in the algorithm's performance and even reduced its accuracy for the automatically extracted Traveldocs knowledge base. The experiments with NationByNation corpus were not performed because this knowledge base did not contain any free text attributes.

5.1.5 Learning weights

Using neural network for training the weights led to improvement in the algorithm's performance. In our test we used manually annotated Traveldocs data as a training set for weights learning. The results confirmed our expectations that assigning greater weights for more important attributes would improve the algorithm's ability to detect matching pairs of instances.

5.2 Instance merging evaluation

Table 3 – Results of the instance merging stage

	Traveldocs (extracted)	Traveldocs (train)	NationByNation	Traveldocs (extracted) & NationByNation
Number of instances merged	132	88	171	99
Number of properties in merging instances:				
- before merging	10216	8617	11198	3441
- after merging	8535	6713	10663	3280
- unique properties (only in one of the KBs)	4549 (44%)	2077 (24%)	8642 (77%)	2543 (73%)
- properties merged	1681 (16%)	1904 (22%)	535 (4%)	161 (4%)
- properties only in the candidate KB	300 (2%)	469 (5%)	206 (1%)	2136 (62%)

In this test we again used CIA World fact book knowledge base as a benchmark. As we stated before, the main advantages of merged knowledge bases are reducing the amount of stored information and the possibility of more convenient representation. The first factor can be measured quantitatively by counting the number of properties, which were merged (and thus not stored twice). The second factor (convenience of representation) can be estimated by measuring the amount of additional information provided from the candidate knowledge base. In other words, it is the number of new properties, which were added from the candidate knowledge base during merging.

In our merging tests we used the best results of the matching instances detection step: pairs of instances selected based on name (other attributes were included only if the name didn't give a candidate pair). The tests were performed with the same pairs of knowledge bases as in the previous instance.

The tests showed that the impact of the fusion step on the resulting knowledge base depends largely on the degree of overlap between knowledge bases.

The degree of overlap was high for CIA World Fact Book and Traveldocs so the amount of merged properties was relatively high (16% and 22%). On the other

hand, NationByNation corpus didn't contain many overlapping attributes so the percentage of merged properties was low (4% in both experiments).

The same reason influenced the number of properties added into the initial knowledge base as a result of fusion. CIA World fact book contained for each instance most of its properties so candidate knowledge bases didn't add much new information into it (between 1% and 5% in our experiments). On the other hand, when we merged NationByNation and Traveldocs knowledge bases, number of added properties was greater (62%) because many attributes were not overlapping.

6. Conclusion

The conclusion section is organised as follows. First, we will list notable issues and problems which were encountered during the experiments. Then, we will discuss possible ways of dealing with these issues. And the last subsection is dedicated to the discussion about the issues of experimental setup itself and its limitations.

6.1 Problems and issues found

6.1.1 Causes of conflicts

The main causes of conflicts found in the experimental data can be classified as follows:

1. Extraction errors.
   These errors in their turn can be divided into two categories:
   - misclassifications: pieces of data assigned to wrong properties (e.g., "hasGDP" and "hasPerCapitaGDP" or "hasEthnicGroups" and "hasReligions");
   - wrong formatting: attributes were determined correctly, but the data was wrongly formatted during extraction (e.g., only fragments of free text attributes instead of whole fields).
2. Different representation of the same data.
   This kind of conflict occurs with attributes of all types: nominal ("UK" and "United Kingdom"), numeric (GDP calculated using purchase power parity and GDP in absolute numbers) and free text (texts with the same meaning but differently phrased).
3. Genuine contradictions between sources.
   In our dataset these were mostly caused by different timestamps by sources referring to the same information in different time (e.g., GDP of the same country from 2001 and 2004).

Examples of each of these kinds of errors can be found in Appendix A. In order to perform high-quality fusion each of these types of conflicts must be recognised and treated in an appropriate way. Extraction errors must be discarded and not stored in the resulting knowledge base; differently represented data must be stored only once in a common format; if the sources contradict each other then all contradicting pieces should be stored in the knowledge base, possibly with confidence weights updated. But to achieve this it is necessary that the algorithm is able to determine the cause of conflict in each case.

6.1.2 Finding instances describing the same entity

The dataset we used for the experiments allowed us to use the matching instances detection procedure based on k-NN machine learning algorithm with k=1. The algorithm was looking for one instance which had the shortest distance with the instance in question. This instance was selected as potentially representing the same instance. Using such an algorithm was possible because we knew that initial and candidate knowledge bases contained instances describing the same sets of entities (countries) and so each instance from a candidate knowledge base had its pair from the initial knowledge base. However, in the general case we cannot make such assumption, which makes the usage of 1-NN algorithm problematic.

6.1.3 Instances wrongly classified as describing the same entity

A crucial step for the successful fusion is determining instances in the initial and candidate knowledge bases which describe the same entities and therefore should be merged together. However, the algorithm looking for such instances, like all machine learning algorithms, does not guarantee 100% accuracy. As our examples shown, even the algorithm with the best set of settings is capable of producing errors (e.g., see Appendix A.1). Usually such errors were caused by small number of overlapping attributes when different representation or misclassification in one of them could distort the results. A problem in this case is how to deal with such cases when the algorithm wrongly classified a pair of instances as referring to the same entity. Applying merging procedure to such pairs will produce serious and irreparable damage to the resulting knowledge base because in future it will become problematic to separate such instances.

6.1.4 Merging attributes

At the merging stage it is important to determine, which attribute values are redundant and therefore should be stored only once. If there are two equal values then the task is trivial but there can be cases when it is more complicated. For instance, if two numeric values are very close to each other it opens a question about whether they should be merged or stored separately. In case of merging it is possible to choose one value (e.g., from a more reliable source) but it can appear incorrect if, for instance, this value is rounded and a value from a less reliable source is more precise. Also an option is merging by averaging. This can be the optimal choice if the values represent the measurement of the same parameter coming from two different sensors. If, on the other hand, it is decided to keep the values separately there remains an issue of updating the confidence weights. If the attribute can only have one value then two values are necessarily contradicting each other and higher confidence in one of them decreases confidence in the other one. However, if the attribute does not have the cardinality limitation then confidence values do not necessarily need to be updated. Similar issues are important for attributes of other types too: e.g., nominal (one value is a substring of another or has slightly different spelling) or free text (one text is longer than the other and contains all words from it).

6.1.5 Different cases of fusion

In our experiments we noticed two specific cases of fusion depending on the structure of instances constituting initial and candidate knowledge bases. These two fusion scenarios are the following:

– much overlapping data (many overlapping properties; main task of the fusion – resolving conflicts) (Appendix A.3);
– small amount of overlapping data (main task of the fusion – combining information) (Appendix A.1).

The complexity of the fusion task differs in these two cases. The second case is relatively straightforward: identifying matching instances does not require comparison based on large sets of attributes and merging process involves little processing of contradictions. In an extreme case the fusion consists only of finding matching instances based on key values and simple combining of attributes as most of them are only mentioned once. In contrast, the first case requires more complex procedure of dealing with conflicts. It can be useful from the point of view of efficiency if the system is able to determine the particular case it deals with and adjust its algorithm accordingly.

6.1.6 Free text attributes

Comparison of free text attributes is a non-trivial task. Simple techniques like the one we used in our experiments are influenced by non-meaningful words, which substantially downgrade the performance, while state-of-the-art techniques for document similarity calculation (like TF-IDF (Salton and Buckley 1988)) are computationally expensive. For instance, TF-IDF algorithm when comparing two pieces of free text requires to process not only two pieces being compared but also all available documents in order to calculate word weights. A distance measure for free text values, which can be used as a part of instance matching algorithm, must not be computationally costly while at the same time should be maximally accurate.

6.1.7 Dependencies in data

In our experiments we found that one of the sources of information, which we used, was dependent on another (Traveldocs used information from the CIA World Fact Book). An evidence for this dependency was the fact that often free text attributes (like “terrain” or “climate”) were identical to each other in both knowledge bases. In real-world applications if we assume that information supported by different sources is more likely to be true these dependencies can influence the results. For instance, if we have several sources supporting a piece of information, then we can increase the confidence weight for this piece, but if these sources are just replicas of one source, we should not do that. Therefore it is important to determine, which sources are independent and which depend on other sources.

6.2 Possibilities to resolve

6.2.1 Causes of conflicts

In our experiments we employed only a simple heuristic, which tried to deal with obvious extraction errors in numeric values (pruning of too small or too large values). The results showed that even such straightforward leads to improvement in the matching instances identification algorithm’s performance. It is possible to propose other heuristics for other types of data (e.g., require that free text attributes must consist of complete sentences to prevent the cases of sentences broken in the middle).

However, it is impossible to discover all cases of extraction errors relying purely on domain-independent heuristic measures. In addition, such heuristic can

produce errors as well. For instance, when the initial knowledge base contains information only about a set of large countries and than receives an instance describing Pitcairn Island, it will judge many of its attributes as mistakenly extracted. This makes it necessary to employ domain data, which in turn means that the domain ontology must be as detailed as possible. In particular, this includes explicit description of the attributes' ranges. For instance, in our experiments domain data we could notice such cases:

- maximal and minimal values for numeric attributes;
- treating nominal attributes as object-to-object relations rather than object-to-string (e.g., specifying that "hasimportpartnerships" must be have a range "Country" instead of "string").
- in case when appropriate classes are not defined in the domain ontology the list of possible values should be given (e.g., for "hasgovernmenttype" it should include "republic", "parliamentary monarchy" etc.)

The next type of conflicts is caused by different representation of the same data. A particular complexity with this type is that domain-independent heuristics are not efficient in dealing with such conflicts. Description of possible ways of representing each property should be described as part of domain ontology. In particular, it includes:

- measurement units for numeric values;
- where different calculation methods are possible, they must be specified (e.g., that GDP can be measured not only in absolute values but also calculated based on purchase power parity);

Such representation formats tend to be source-dependent. It means that very often if the source gives a particular format for one of the values, then it is likely that the same format will be used for the value of the same property for another instance. In our case, for example, NationByNation site always gives the value of the country's territory measured in square miles. Determining undefined preferences of different sources can be addressed using machine learning methods. For example, by comparison with the values already in the knowledge base it is possible to find that country's territory (being a functional property) is likely to be measured in square miles. Then it is possible to assume the same unit of measurement for new values from the same source in case when it is not defined explicitly.

It is possible to propose implementing procedures for determining these two types of conflicts as filters, which will evaluate incoming instances and refine the data before it is passed to the merging stage. The last type of conflicts – a "genuine" conflict in this case will be determined by exclusion: all conflicts which were not eliminated at previous stages will be treated as contradictions between sources.

6.2.2 Finding instances describing the same entity

The problem with finding potential matching instances in case, when we are uncertain about whether a pair for each candidate instance exists or not, means that we can only be certain about pairs of instances identified by key value. Identifying pairs of instances which are actually referring to the same entity but do not have identical key values becomes a task, which cannot be solved with a 100% guarantee of quality. Possible ways of dealing with this include implementing machine learning techniques and storing merged instances in such a way that it is possible to represent different degrees of uncertainty about merging. Machine learning techniques can use list of pairs identified by key values as a training set and then employ built model for

new instances. Uncertainty about merging can be implemented by using <owl:sameAs> relations with confidence degrees attached.

6.2.3 Instances wrongly classified as describing the same entity

As was stated before, merging together instances which in fact do not refer to the same entity, will introduce into the knowledge base an error making impossible correct reasoning about both entities in future. After two instances are merged together it will become impossible to repair the error and to separate them even if more evidence will become available from other sources. Therefore it is sensible to perform merging in such a way that will leave a possibility to separate instances afterwards. So complete merging of instances (i.e., producing one instance instead of two) should be performed only in such cases when we can be certain about its correctness (e.g., objects are identified uniquely by keys). In other cases, for example when a pair of instances was detected using a record similarity measuring algorithm, it will be more useful to store both instances separately in the knowledge base but link them using <owl:sameAs> relation with a confidence degree.

6.2.4 Merging attributes

When determining the way to merge values of a particular property it is important to keep in mind that merging should be performed in such a way that no information that can be valuable for any application is lost. For instance, it can be the case that for most users only the average value is important but for some category of users separate values are also needed (e.g., a technician repairing sensors may need separate values from each sensor while for most users only averaged values are needed). Thus we should distinguish the generic fusion task and application-dependent fusion when the desired representation is known from the beginning (e.g., fusing industrial data from a company's intranet). In case of generic fusion task the algorithm should perform complete merging only in cases when it can be sure that information is redundant (i.e., precise equality or synonymy).

6.2.5 Different cases of fusion

In order to determine a particular case of fusion it is possible to evaluate incoming instances and compare the set of attributes with the set already in the knowledge base. After determining, which properties have overlaps, it is possible to choose appropriate methods for conflict detection and resolving. For instance, if two sources describe the same entities from a different point of view (e.g., economical and geographical description of countries), it does not make sense to run the matching instances identification algorithm based on comparison of all properties and using neural network training to determine weights, it is sufficient to identify instances by key fields (country name/capital).

6.2.6 Free text attributes

In order to compare two free text attributes with high quality we need to use additional sources of information while at the same time not to involve other values from the knowledge base (like in TF-IDF) because it will make the algorithm perform slower with the growth of the knowledge base. One of the possible types of additional sources of linguistic information are pre-defined lists of stop-words. Such lists are used in natural language process applications and contain non-meaningful words, which should not be included into comparison. Another technique is to use existing

lexicons to improve the quality of comparison by considering not only exact words but also word forms and word senses. One of such lexicons is WordNet.

6.2.7 Dependencies in data

A good evidence for inter-dependencies between two sources is their exact textual equality. Having two long pieces of free text points to the possibility that one of them is a copy of another. In this case we should take into account this dependency when determining matching instances and updating confidence weights. If one of two sources is likely to be dependent on another then the fact that they agree between themselves should not increase our belief in the data they provide.

6.3 Discussion about methodology

When constructing the experimental set for this study we made several design choices regarding the data sets and the experiments. During the experimental part of our study we found that some of these choices affected the quality of the experiments and made the evaluation of experimental results more difficult. Below we will mention these factors and discuss what impact they had on our results.

For the reasons discussed in the section 3 we selected the web sites related to geographic domain as a dataset for our experiments. However, during experiments we found that this choice also had certain negative influence on experiments. In particular, it was noticed that the sources were not independent: information from CIA World Fact Book was partially reused by other sources. The evidence for it was, for example, precise equality of certain free text fields in CIA World Fact Book and Traveldocs datasets. These equal fields influenced the results of the matching instances identification algorithm: the algorithm's performance was better than would be the case if the sources were independent.

Another issue was caused by the chosen domain of data sets. Information about countries included amount of attributes of different data types: numeric, nominal and free text. However, a disadvantage of this is a limited number of instances in our knowledge bases. This made it hard to evaluate the importance of the differences in the results of our experiments and the ability of the algorithm to scale.

However, despite these issues we were able to study the issues which were set as goals of the study. As was specified in the section 2, the main goals were:

1. To discover possible problems with fusing automatically annotated data.
   In the course of the study we encountered a number of problems and issues, which were discussed in sections 6.1.1 – 6.1.7. Future work is needed to address these issues. A preliminary set of approaches which can be taken is listed in sections 6.2.1 – 6.2.7.
2. To evaluate the applicability of quantitative matching instances detection algorithms for automatically extracted data.
   In the course of our study we found that distance finding algorithm based on aggregating distances between properties does give an improvement over simple identification by key fields if instances being compared contain overlapping properties.
3. To experiment with the parameters of the algorithm and evaluate their influence on the algorithm's performance.
   We checked different parameters of the matching instances detection algorithm and found that such techniques as pruning of potentially erroneous numeric values and learning attribute weights can lead to the improvement in the algorithm's performance.

Research proposal

1. Related work

The literature on knowledge fusion (Appriou, Ayoun et al. 2001) recognises the following 6 types of problems, which should be dealt with: ambiguity, uncertainty, incompleteness, imprecision, vagueness and inconsistency (see literature review, section 3). These defects in information have three main origins. First, existing information extraction algorithms cannot ensure 100% extraction correctness, which leads to uncertainty and incompleteness. Second, multiple sources can contradict each other, which leads to inconsistency and ambiguity and uncertainty. Third, the information itself can be imprecise, incomplete or vague. Thus, we can say that the problems of imprecision, incompleteness and vagueness are inherent for the fusion input data. The aim of the fusion step is to solve these problems using multiple sources. In contrast, the problems of ambiguity and inconsistency (and uncertainty, which is produced by them) are caused by using multiple sources and are part of fusion process itself. Therefore we can attempt to overcome them by improving the fusion algorithm.

Related work was mainly concentrated on two problems related to knowledge fusion: ambiguity and inconsistency. These two problems were mostly addressed separately by two different communities (see literature review, sections 2.1-2.2). There are also existing implementations of ontological languages able to represent uncertainty. However, in the studied literature we did not find a description of an existing domain-independent framework, which would address all these issues. Thus, we think that designing such a framework would make a contribution into the research on knowledge fusion. Another topic, which can be important, is the adaptation of the conflict resolution algorithms, which can deal with uncertain information, to the Semantic Web domain.

These issues are also important in the context of the X-Media project, which deals with management of semantic annotations produced from different media (see X-Media Technical Annex document).

2. Problem statement

In our work we plan to address the following main problems in the context of Semantic Web:

1. Developing a generic knowledge fusion framework.
2. Implementing methods for conflict resolution between uncertain statements.

The main requirements, which the framework should satisfy, are:

1. Ability to perform the fusion task in a domain-independent way (i.e. domain ontologies should be part of the input of the framework but not an inherent component of the framework).
2. Ability to select appropriate methods for solving the fusion subtasks from existing ones.

We assume that the starting conditions for knowledge fusion include availability of the knowledge base (we refer to it as the initial knowledge base), into which new annotations should be integrated. These new annotations we call the candidate knowledge base. The third component of the input data for the knowledge fusion framework includes domain ontologies, according to which the annotations

were constructed. The output of the fusion framework is the updated initial knowledge base, into which new information from the candidate knowledge base was integrated.

Major issues, which have to be resolved when constructing the framework include the following:

1. Task decomposition.
2. Organising the task-method cooperation.
3. Designing the algorithm for selecting appropriate method for each task.
4. Designing standard interfaces for data interchange between different components of the framework.

In our study we do not plan to develop novel solutions in the following areas:

1. Developing new matching instance detection methods.
2. Developing new ontological language for uncertainty representation.
3. Developing new information extraction algorithms.

For all these components we plan to use existing implementations, possibly adapting them for our needs.

3. Proposed approach

In order to design a generic framework for knowledge fusion we propose to view the knowledge fusion as a complex task, which can be decomposed into simpler subtasks. Each task can be performed using one or more methods so these methods can be organised into a library from which the system will pick appropriate methods according to the task requirements. In more detail such an approach is described in (Motta 1999). At the current stage it is possible to decompose the knowledge fusion task into the following subtasks (see Fig. 1).

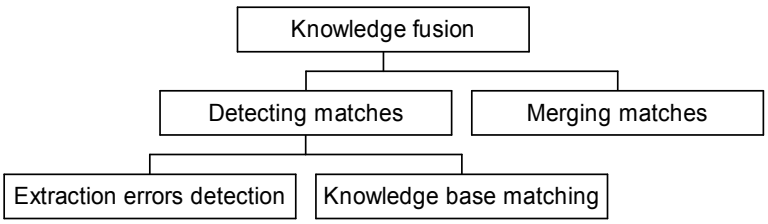

Figure 1 – Preliminary task decomposition for the knowledge fusion framework

Two main subtasks of the knowledge fusion are detecting matches and merging matches. The match detection subtask receives as its input the initial knowledge base, a candidate knowledge base and domain ontologies. Its task is to detect the subsets of two knowledge bases which refer to the same real-world entities and therefore will be effected by fusion. The output of the match detection subtask includes the set of instances and facts, which will be effected by fusion. This task itself can be decomposed into the preprocessing step aimed at eliminating spurious data caused by incorrect information extraction and knowledge base matching step, which performs actual detection of matching instances. Then this set of matching instances serves as an input to the matching instances merging task. This task includes actual merging of two knowledge bases and in particular, deals with conflicts between matching instances.

A crucial part of the framework implementation is an algorithm for selecting the best method from the library. Atomic subtasks of the generic knowledge fusion

task may have more than one method applicable to them. For instance, the following methods (algorithms) can be applied to the knowledge base matching task (e.g., see pilot study report):

- identifying instances by known key fields;
- learning a set of key fields from the initial knowledge base and identifying instances using them;
- identifying instances by calculating distances between property values using weight learning.

This list is not complete: in general, the knowledge base matching task is a kind of classification task (classifying pairs of instances referring to the same entity and pairs that do not), and there are many existing machine learning algorithms developed to solve the classification problem.

Using one method does not necessarily exclude using another method in addition. For example, a distance calculation method can be applied after the identification by key fields and deal only with pairs not discovered at the previous step. It means that in order to execute a task a knowledge fusion system must:

- select a set of methods applicable to the task (not only the best method);
- select a correct order in which they should be applied.

The system can make these selections based on the following information:

1. Reliability of the method.
   This measure can be either received from the method description (e.g., assigned by third-party experts) or found by comparing its results with the results of other methods.
2. Analysing sets of input and output roles of each method (e.g., methods based on machine learning require training sets).
3. Domain-specific information (e.g., for the law domain the methods designed for dealing with numeric data are not applicable).
4. Application-specific information (e.g., explicit knowledge about how conflicts should be dealt with).

For the conflict resolution the literature (Appriou, Ayoun et al. 2001) distinguishes the following types of approaches:

1. Unilateral resolution.
   This approach assumes that among conflicting alternatives only one is selected and the others discarded.
2. Compromise (disjunctive or averaging).
   Compromise assumes that all conflicting alternatives are in some way represented in the resulting knowledge base. The values can be aggregated with disjunctions or by averaging in cases when the values are numeric.
3. Create a new world.
   This decision assumes that the results of the conflict detection study were incorrect and conflicting statements in fact do not refer to the same entities.
4. Delaying the decision until new information arrives.
   This approach leads to increased complexity in the knowledge base because all hypotheses must be not only stored together, as in case 3, but are excluded from reasoning until the decision is made. Also in the general case there is no guarantee that the desired piece of information needed to resolve the conflict will appear. Therefore, such an approach is not promising on the large scale.

Resolving conflicts is a task, which depends on the context in which fusion is performed, in other words, on the domain and application. One domain can have more than one type of application, each of which imposes its own set of requirements to the fused data. For instance, in technical domain there can be the case of a parameter measured by several sensors with some degree of precision. Most users will be interested in a single parameter's value believed to be true. This value can be obtained by averaging all values. In contrast, a member of support staff trying to identify the broken sensor needs information coming from each sensor separately. Therefore, in the general case if we have no assumptions about the application-specific requirements we always need to keep all incoming information in the resulting knowledge base. This will lead to increasing complexity of the knowledge base. However, if there exist assumptions about what kind of information is required (e.g., a user model) then the system can decide how to treat each case of conflict.

Another component of the framework is the fusion ontology, which should explicitly describe the information related to the fusion process. This ontology has to model, among others, the following concepts:

1. Fusion set.
   This is an atomic set of interlinked instances and statements from two knowledge bases, which are matching and have to be merged together.
2. Conflict resolving preferences.
   As we specified before, there are several possible approaches to conflict resolution. Main of them are unilateral resolution and compromise: the third one is only an error handling technique. The first approach leads to more compact and consistent knowledge base but there is more risk of losing important information while the second allows keeping all possible information but makes the resulting knowledge base large and less consistent. The preferences should model the preferred way of conflict resolution for each particular conflict.
3. Fusion solution.
   Fusion solution represents a set of instances and relations to be added into the knowledge base as a result of fusion.

A necessary part of the project after the framework will be implemented is its evaluation. We envisage two possible ways of evaluation:

1. Gold standard evaluation.
   This approach assumes availability of a test set consisting of an initial knowledge base and a candidate knowledge base, for which the fusion procedure was performed manually. Then the resulting knowledge base after manually performed fusion can be compared with the results of fusion performed by the system. The comparison can be based on quantitative metrics, which measure the quality of fusion (e.g., percentage of correctly identified matching instances, percentage of correctly merged properties etc.). As test beds for the gold standard evaluation it is possible to use existing evaluation datasets used in the database community (e.g., those stored in RIDDLE repository[4])
2. Task-based evaluation.
   This type of evaluation involves user studies and is based on comparing the users' experiences with solving some task using fused and non-fused

[4] http://www.cs.utexas.edu/users/ml/riddle/

data (e.g., information search in the fused knowledge base and in the separate sources).

4. Risk analysis

The main risks of the study are related to its interdependencies with other work packages of the X-Media project. In particular, in order to implement and evaluate the framework the following pre-requisites are necessary:

1. Availability of an extraction tool to provide the input of the fusion framework.
2. Availability of an evaluation dataset and domain ontology.
3. Availability of a representation language supporting uncertainty.
4. Availability of implemented methods for solving fusion subtasks.

All these parts are planned to be developed in the course of the X-Media project. However, it is possible that some of these implementations will not become available in time to be included into this PhD project. In this case it is necessary to envisage possible alternative solutions.

There are several available existing extraction tools. Particularly interesting is a set of tools developed in the University of Sheffield (e.g., Armadillo (Ciravegna and Chapman 2005)) . If necessary, these tools can be used to provide input data for the framework. Test datasets and ontologies used by the authors of the tools can be reused in our study.

In case the uncertainty representation language is not developed early enough in the course of X-Media project a possible alternative is to use the existing Fuzzy OWL implementation (Stoilos, Stamou et al. 2005).

Methods for detecting matching instances and resolving conflicts should be developed as parts of the knowledge fusion work package in the X-Media project. While it is possible to reuse existing modules for detecting matching instances stage (e.g., SimMetrics library[5], developed in the University of Sheffield, or the modification of the algorithm used for pilot study), for resolving conflicts new methods have to be implemented.

Another complication which may arise from the interconnection with the X-Media project is the difficulty in clarifying the research contribution. In order to prevent this, it is necessary to establish the distribution of tasks within the knowledge fusion work package of the X-Media project at the early stage.

[5] http://sourceforge.net/projects/simmetrics/

5. Work plan

Table 1 – Major work stages

	Name	Start	End	Comments
1	Design	Aug 2006	Jan 2007	- Choosing existing techniques, which can be reused; - Outlining necessary modules to be implemented
2	Implementation	Sep 2006	Jan 2008	- Constructing a library of methods solving fusion subtasks (including adaptation of existing ones); - Implementing the framework
3	Evaluation	Sep 2007	Apr 2008	
4	Writing-up	Oct 2007	Aug 2008	

Table 2 – Milestones

	Date	Comments
1	Oct 2006	Review of the design draft v0.1
2	Jan 2007	Review of the design (complete) Review of the implementation progress
3	Jul 2007	Review of the implementation demo Review of the evaluation plan
4	Nov 2007	Review of the evaluation status Review of the thesis plan (chapter headings)
5	Apr 2008	Final review of experimental work Review of the thesis status (methodology + skeleton of implementation)
6	Jul 2008	Review of the first draft

Table 3 – Suggested papers

	Date	Comments
1	Mid-2007	Description of the demo implementation
2	1st half 2008	Description of work & evaluation

References

X-Media Annex 1 - Description of Work.

Appriou, A., A. Ayoun, et al. (2001). "Fusion: General concepts and characteristics." International Journal of Intelligent Systems **16**(10): 1107-1134.

Baumgartner, R., S. Eichholz, et al. (2003). Semantic Markup of News Items with Lixto. Annotation for the Semantic Web. S. Handschuh and S. Staab. Amsterdam, IOS Press. **96:** 63-78.

Benitez, A. B. and S.-F. Chang (2002). Semantic Knowledge Construction From Annotated Image Collections. International Conference on Multimedia and Expo, Lausanne, Switzerland.

Cholvy, L. (2004). Information Evaluation in fusion: a case study. 10th International Conference on Information Processing and management of Uncertainty in Knowledge-Based Systems, Perugia, Italy.

Cimiano, P. (2005). Ontology Learning and Population. Dagstuhl Seminar "Machine Learning for the Semantic Web".

Cimiano, P., G. Ladwig, et al. (2005). Gimme' the context: context-driven automatic semantic annotation with C-PANKOW. 14th international conference on World Wide Web, Chiba, Japan, ACM Press.

Ciravegna, F. (2003). (LP)2: Rule Induction for Information Extraction Using Linguistic Constraints. Sheffield.

Ciravegna, F. and S. Chapman (2005). Mining the Semantic Web: Requirements for Machine Learning. Dagstuhl seminar: "Machine learning for the Semantic Web".

Ciravegna, F., S. Chapman, et al. (2004). Learning to Harvest Information for the Semantic Web. 1st European Semantic Web Symposium, Heraclion, Crete, Greece.

Ciravegna, F. and Y. Wilks (2003). Designing Adaptive Information Extraction for the Semantic Web in Amilcare. Annotation for the Semantic Web. S. Handschuh and S. Staab. Amsterdam, IOS Press. **96:** 112-127.

Cochinwala, M., S. Dalal, et al. (2001). "Record matching: Past, present and future."

da Costa, P. C. G., K. B. Laskey, et al. (2005). PR-OWL: A Bayesian Ontology Language for the Semantic Web. Workshop on Uncertainty Reasoning for the Semantic Web, International Semantic Web Conference.

Ding, Z. and Y. Peng (2004). A Probabilistic Extension to Ontology Language OWL. 37th Hawaii International Conference On System Sciences (HICSS-37), Big Island, Hawaii.

Doan, A., Y. Lu, et al. (2003). Object matching for information integration: A profiler-based approach. IJCAI-03 Workshop on Information Integration on the Web.

Dubois, D., M. Grabish, et al. (2001). "Using the transferable belief model and a qualitative possibility theory approach on an illustrative example: The assessment of the value of a candidate." International Journal of Intelligent Systems **16**(11): 1245-1272.

Etzioni, O., M. Cafarella, et al. (2005). "Unsupervised Named-Entity Extraction from the Web: An Experimental Study." Artificial Intelligence.

Grabish, M. and H. Prade (2001). "The correlation problem in sensor fusion in a possibilistic framework." International Journal of Intelligent Systems **16**(11): 1273-1283.

Gregoire, E. (2006). "An unbiased approach to iterated fusion by weakening." Information Fusion
Logic-based Approaches to Information Fusion **7**(1): 35-40.
Guha, R. (2004). Object Co-Identification on the Semantic Web. 13th World Wide Web Conference, New York, USA.
Guha, R. and A. Garg (2004). Disambiguating People in Search. 13th World Wide Web Conference, New York, USA.
Guha, R. and R. McCool (2003). "TAP: An Semantic Web Test-bed." Journal of Web Semantics **1**(1).
Handschuh, S. and S. Staab (2003). Annotation of the Shallow and the Deep Web. Annotation for the Semantic Web. S. Handschuh and S. Staab. Amsterdam, IOS Press. **96:** 25-45.
Hearst, M. (1992). Automatic Acquisition of Hyponyms from Large Text Corpora. Fourteenth International Conference on Computational Linguistics, Nantes, France.
Hollink, L., G. Schreiber, et al. (2003). Semantic Annotation of Image Collections. KCAP'03 Workshop on Knowledge Markup and Semantic Annotation, Florida, USA.
Huang, L.-K., Z.-W. Jian, et al. (2005). "A fuzzy ontology and its application to news summarization." IEEE Transactions on Systems, Man and Cybernetics **35**(5): 859-880.
Hunter, A. (2006). "How to act on inconsistent news: Ignore, resolve, or reject." Data & Knowledge Engineering **57**(3): 221-239.
Hunter, A. and W. Liu (2006). "Fusion rules for merging uncertain information." Information Fusion
Logic-based Approaches to Information Fusion **7**(1): 97-134.
Koivunen, M.-R. and R. R. Swick (2003). Collaboration through Annotations in the Semantic Web. Annotation for the Semantic Web. S. Handschuh and S. Staab. Amsterdam, IOS Press. **96:** 46-60.
Korb, K. B. (2004). Bayesian artificial intelligence. Boca Raton, Chapman&Hall/CRC.
Lee, M. D., B. Pincombe, et al. (2005). An empirical evaluation of models of text document similarity. 27th Annual Meeting of the Cognitive Science Society, Austin, TX, USA.
Mitra, P., N. F. Noy, et al. (2004). OMEN: A Probabilistic Ontology Mapping Tool. Workshop on Meaning coordination and negotiation at the Third International Conference on the Semantic Web (ISWC-2004), Hiroshima, Japan.
Motta, E. (1999). Reusable components for knowledge modelling. Amsterdam, IOS Press.
Novak, V. (1989). Fuzzy sets and their applications. Bristol, Adam Hilger.
Salton, G. and C. Buckley (1988). "Term-weighting approaches in automatic text retrieval." Information Processing & Management **24**(5): 513-523.
Stoilos, G., G. Stamou, et al. (2005). Fuzzy OWL: Uncertainty and the Semantic Web. International Workshop of OWL: Experiences and Directions, Galway, Ireland.
Widrow, B. and M. E. Hoff (1960). "Adaptive switching circuits." IRE WESCON convention record **4**: 96-104.
Zadeh, L. (1999). "Fuzzy sets as a basis for a theory of possibility." Fuzzy Sets and Systems **100**(1999 supp.): 9-34.

Appendix A – Examples of merged instances

1. Turkey/Ivory Coast – wrongly merged instances

Turkey

haslanguages	Turkish	CIA
	Greek	CIA
	Armenian	CIA
	Arabic	CIA
	Kurdish	CIA
populationhasreligions	Other	CIA D://Study//Dataset//testrdf_whole.rdf
	Muslim	CIA D://Study//Dataset//testrdf_whole.rdf
	Indigenous	D://Study//Dataset//testrdf_whole.rdf
	and Christian Catholic	D://Study//Dataset//testrdf_whole.rdf
	five principal language groups Education Years compulsoryschool is not compulsory at this time Attendance (*)	D://Study//Dataset//testrdf_whole.rdf
hasbudgetexpenditures	6.91E10	CIA
hasbudgetrevenues	4.24E10	CIA
hasethnicgroups	Turkish	CIA
	Kurdish	CIA
	Akan east and center (*)	D://Study//Dataset//testrdf_whole.rdf
	including Lagoon peoples of the southeast (*)	D://Study//Dataset//testrdf_whole.rdf
	Krou southwest (*)	D://Study//Dataset//testrdf_whole.rdf
	Southern Mande west (*)	D://Study//Dataset//testrdf_whole.rdf
	Northern Mande northwest (*)	D://Study//Dataset//testrdf_whole.rdf
	Senoufo Lobi north center and northeast The Baoules (*)	D://Study//Dataset//testrdf_whole.rdf
	in the Akan division (*)	D://Study//Dataset//testrdf_whole.rdf
	probably comprise the single largest subgroup with of the population They are based in the central region around Bouake and Yamoussoukro The Betes in the Krou division (*)	D://Study//Dataset//testrdf_whole.rdf
	the Senoufos in the north (*)	D://Study//Dataset//testrdf_whole.rdf
	and the Malinkes in the northwest and the cities are the next largest groups (*)	D://Study//Dataset//testrdf_whole.rdf
	with each of the national population Most of the principal divisions have a significant presence in neighboring countries (*)	D://Study//Dataset//testrdf_whole.rdf

hasarableland	34.53	CIA
hasnaturalresources	Borate	CIA
	Antimony	CIA
	Copper	CIA
	Chromium	CIA
	iron ore	CIA
	Coal	CIA
	arable land	CIA
	Sulphur	CIA
	Hydropower	CIA
	Mercury	CIA
haschildmortalityperthousand	44.2	CIA
haseconomyoverview	Turkey's dynamic economy is a complex mix of modern industry and commerce along with a traditional agriculture sector that in 2001 still accounted for 40% of employment. It has a strong and rapidly growing private sector, yet the state still plays a major role in basic industry, banking, transport, and communication. The most important industry - and largest exporter - is textiles and clothing, which is almost entirely in private hands. In recent years the economic situation has been marked by erratic economic growth and serious imbalances. Real GNP growth has exceeded 6% in many years, but this strong expansion has been interrupted by sharp declines in output in 1994, 1999, and 2001. Meanwhile, the public sector fiscal deficit has regularly exceeded 10% of GDP - due in large part to the huge burden of interest payments, which account for more than 50% of central government spending; inflation has remained in the high double-digit range. Perhaps because of these problems, foreign direct investment in Turkey remains low - less than $1 billion annually. In late 2000 and early 2001 a growing trade deficit and serious weaknesses in the banking sector plunged the economy into crisis - forcing Turkey to float the lira and pushing the country into recession. Results in 2002 were much better, because of strong financial support from the IMF	CIA

	and tighter fiscal policy. Continued slow global growth and serious political tensions in the Middle East cast a shadow over prospects for 2003.	
hascapitalcity	Ankara	CIA
haspopulation	6.8109469E7	CIA
	1.87E7	D://Study//Dataset//testrdf_whole.rdf
hasnationalitynounform	Turk(s)	CIA
	Ivoirian	D://Study//Dataset//testrdf_whole.rdf
terrain	high central plateau (Anatolia); narrow coastal plain; several mountain ranges	CIA
hasexportpartnerships	Germany	CIA D://Study//Dataset//testrdf_whole.rdf
	United Kingdom	CIA
	Russia	CIA
	France	CIA D://Study//Dataset//testrdf_whole.rdf
	United States (**)	CIA
	Italy	CIA
	US (**)	D://Study//Dataset//testrdf_whole.rdf D://Study//Dataset//testrdf_whole.rdf
	Netherlands Total imports of GDP (*)	D://Study//Dataset//testrdf_whole.rdf
haspopulationgrowthrate	1.16	CIA
hasrealgrowthrategdp	7.8	CIA
hasarea	780580.0	CIA
	322500.0	D://Study//Dataset//testrdf_whole.rdf
hasimportpartnerships	Germany	CIA D://Study//Dataset//testrdf_whole.rdf
	United States (**)	CIA
	France	CIA D://Study//Dataset//testrdf_whole.rdf
	Russia	CIA
	United Kingdom	CIA
	Italy	CIA D://Study//Dataset//testrdf_whole.rdf
	Nigeria	D://Study//Dataset//testrdf_whole.rdf
	US (**)	D://Study//Dataset//testrdf_whole.rdf
hascoastlinekilometers	7200.0	CIA
haslabel	Turkey	CIA
	Ivory Coast	D://Study//Dataset//testrdf_whole.rdf
hasgdp	4.68E11	CIA
	1.74E10	D://Study//Dataset//testrdf_whole.rdf
haspercapitagdp	7000.0	CIA
hasbirthrateperthousand	17.59	CIA
hasfemaleliteracyrate	78.7	CIA
hasdeathrateperthousand	5.95	CIA

hasmaleliteracyrate	94.3	CIA
hasexternaldebt	1.183E11	CIA
hasareacomparitive	slightly larger than Texas	CIA
	slightly larger than New (*)	D://Study//Dataset//testrdf_whole.rdf
haspermanentcropland	3.36	CIA
climate	temperate; hot, dry summers with mild, wet winters; harsher in interior	CIA
haslocation	southeastern Europe and southwestern Asia (that portion of Turkey west of the Bosporus is geographically part of Europe), bordering the Black Sea, between Bulgaria and Georgia, and bordering the Aegean Sea and the Mediterranean Sea, between Greece and Syria	CIA
hasliteracyrate	86.5	CIA
longformname	Republic of Turkey	CIA
haslaborforce	2.38E7	CIA

2. Burkina Faso – small number of overlapping properties

Burkina Faso

haslabel	Burkina Faso	D://Study//Dataset//nationbynation.rdf D://Study//Dataset//testrdf_whole.rdf
haspercapitagdp	1100.0	D://Study//Dataset//nationbynation.rdf
	300.0	D://Study//Dataset//testrdf_whole.rdf
hasgdp	1.21E10	D://Study//Dataset//nationbynation.rdf
	4.5E9	D://Study//Dataset//testrdf_whole.rdf
haspopulationgrowthrate	2.4	D://Study//Dataset//nationbynation.rdf
hasarea	105869.0	D://Study//Dataset//nationbynation.rdf
	274200.0	D://Study//Dataset//testrdf_whole.rdf
haslifeexpectancy	44.0	D://Study//Dataset//nationbynation.rdf
haschildmortalityperthousand	99.0 (***)	D://Study//Dataset//nationbynation.rdf
	83.0 (***)	D://Study//Dataset//testrdf_whole.rdf
hasexportpartnerships	European Union	D://Study//Dataset//testrdf_whole.rdf
	Taiwan	D://Study//Dataset//testrdf_whole.rdf
haspopulation	1.22E7	D://Study//Dataset//testrdf_whole.rdf
hasliteracyrate	32.25	D://Study//Dataset//testrdf_whole.rdf
hasexports	2.64E8	D://Study//Dataset//testrdf_whole.rdf
hasagriculturepercentlaborforce	92.0	D://Study//Dataset//testrdf_whole.rdf
hasimports	2.64E8	D://Study//Dataset//testrdf_whole.rdf
hasnationalitynounform	Burkinabe	D://Study//Dataset//testrdf_whole.rdf

3. Germany – large number of overlapping properties

Germany

haslanguages	German	CIA D://Study//Dataset//testrdf_whole.rdf
populationhasreligions	Muslim	CIA
	unaffiliated or other	CIA
	Roman Catholic	CIA
	Protestant	CIA
hasbudgetexpenditures	8.25E11	CIA
hasbudgetrevenues	8.02E11	CIA
hasethnicgroups	Other	CIA
	German	CIA D://Study//Dataset//testrdf_whole.rdf
	Turkish	CIA
	Sorbian Slavic minority in the east (*)	D://Study//Dataset//testrdf_whole.rdf
	million foreign residents Religions Protestants million slightly outnumber Roman Catholics million (*)	D://Study//Dataset//testrdf_whole.rdf
hasarableland	33.88	CIA
hasnaturalresources	iron ore	CIA D://Study//Dataset//testrdf_whole.rdf
	arable land	CIA
	Coal	CIA D://Study//Dataset//testrdf_whole.rdf
	Nickel	CIA D://Study//Dataset//testrdf_whole.rdf
	Potash	CIA D://Study//Dataset//testrdf_whole.rdf
	Timber	CIA D://Study//Dataset//testrdf_whole.rdf
	Uranium	CIA D://Study//Dataset//testrdf_whole.rdf
	Salt	CIA D://Study//Dataset//testrdf_whole.rdf
	natural gas	CIA D://Study//Dataset//testrdf_whole.rdf
	Copper	CIA D://Study//Dataset//testrdf_whole.rdf
	lignite	CIA D://Study//Dataset//testrdf_whole.rdf
haschildmortalityperthousand	4.23 (***)	CIA
	5.0 (***)	D://Study//Dataset//testrdf_whole.rdf
haseconomyoverview	Germany's affluent and technologically powerful economy turned in a relatively weak performance throughout much of the 1990s. The modernization and integration of the eastern German economy continues to be a costly long-term problem, with annual transfers from west to east amounting to roughly $70 billion.	CIA

	Germany's ageing population, combined with high unemployment, has pushed social security outlays to a level exceeding contributions from workers. Structural rigidities in the labor market - including strict regulations on laying off workers and the setting of wages on a national basis - have made unemployment a chronic problem. Business and income tax cuts introduced in 2001 did not spare Germany from the impact of the downturn in international trade, and domestic demand faltered as unemployment began to rise. Growth in 2002 again fell short of 1%. Corporate restructuring and growing capital markets are setting the foundations that could allow Germany to meet the long-term challenges of European economic integration and globalization, particularly if labor market rigidities are addressed. In the short run, however, the fall in government revenues and the rise in expenditures has brought the deficit close to the EU's 3% debt limit.	
hascapitalcity	Berlin	CIA
haspopulation	8.2398326E7	CIA D://Study//Dataset//testrdf_whole.rdf
hasnationalitynounform	German	CIA D://Study//Dataset//testrdf_whole.rdf
terrain	lowlands in north, uplands in center, Bavarian Alps in south	CIA
hasexportpartnerships	France	CIA D://Study//Dataset//testrdf_whole.rdf
	Netherlands	CIA D://Study//Dataset//testrdf_whole.rdf
	Spain	CIA
	Switzerland	CIA
	United States (**)	CIA
	United Kingdom	CIA
	US (**)	D://Study//Dataset//testrdf_whole.rdf
haspopulationgrowthrate	0.04	CIA
hasrealgrowthrategdp	0.4	CIA
hasarea	357021.0	CIA
hasimportpartnerships	France	CIA D://Study//Dataset//testrdf_whole.rdf
	Belgium	CIA
	Italy	CIA
	Austria	CIA

	Japan	CIA
	Netherlands	CIA D://Study//Dataset//testrdf_whole.rdf
	United Kingdom	CIA
	United States (**)	CIA
	US (**)	D://Study//Dataset//testrdf_whole.rdf
hascoastlinekilometers	2389.0	CIA
haslabel	Germany	CIA D://Study//Dataset//testrdf_whole.rdf
hasgdp	2.184E12 (***)	CIA
	1.8 (***)	D://Study//Dataset//testrdf_whole.rdf
haspercapitagdp	26600.0 (***)	CIA
	22900.0 (***)	D://Study//Dataset//testrdf_whole.rdf
hasbirthrateperthousand	8.6	CIA
hasdeathrateperthousand	10.34	CIA
hasunemployedpercentage	9.8	CIA
hasareacomparitive	Slightly smaller than Montana	CIA
haspermanentcropland	0.65	CIA
climate	temperate and marine; cool, cloudy, wet winters and summers; occasional warm foehn wind	CIA D://Study//Dataset//testrdf_whole.rdf
haslocation	Central Europe, bordering the Baltic Sea and the North Sea, between the Netherlands and Poland, south of Denmark	CIA D://Study//Dataset//testrdf_whole.rdf
hasliteracyrate	99.0	CIA
longformname	Federal Republic of Germany	CIA
haslaborforce	4.19E7	CIA
	22900.0 (*)	D://Study//Dataset//testrdf_whole.rdf
hasimports	5.94E11	D://Study//Dataset//testrdf_whole.rdf
hasexports	6.28E11	D://Study//Dataset//testrdf_whole.rdf
hasindustrypercentlaborforce	100.0	D://Study//Dataset//testrdf_whole.rdf
haslifeexpectancy	80.0	D://Study//Dataset//testrdf_whole.rdf

Notes:
* - incorrectly extracted values.
** - contradictions because of different format.
*** - genuine contradictions between sources.

Appendix B –Ontologies

1. TAP Ontology

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- TAP KB Fragment. Made available under the TAP KB Open License. Please see http://tap.stanford.edu/license.html -->
<!DOCTYPE uridef [
  <!ENTITY tap "http://tap.stanford.edu/data/">
  <!ENTITY xsd "http://www.w3.org/2001/XMLSchema#">
  <!ENTITY ftmeta "http://kmi.open.ac.uk/fusiontest/meta/#">
]>
<rdf:RDF
   xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
   xmlns:dc="http://purl.org/dc/elements/1.1/"
   xmlns:a="http://tap.stanford.edu/2002/05/GetDataSchema#"
   xmlns:daml="http://www.daml.org/2001/03/daml+oil#"
   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
   xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
   xmlns:ftmeta="&ftmeta;"
   xmlns:tap="&tap;">
 <rdfs:Class rdf:ID="&tap;Resource">
  <rdfs:label xml:lang="en">Resource</rdfs:label>
 </rdfs:Class>
 <rdfs:Class rdf:ID="&tap;Tangible">
  <rdfs:subClassOf rdf:resource="&tap;Resource"/>
 </rdfs:Class>
 <rdfs:Class rdf:ID="&tap;Place">
  <rdfs:label xml:lang="en">Place</rdfs:label>
  <rdfs:subClassOf rdf:resource="&tap;Tangible"/>
 </rdfs:Class>
 <rdfs:Class rdf:ID="&tap;GeoPoliticalRegion">
  <rdfs:subClassOf rdf:resource="&tap;Place"/>
 </rdfs:Class>
 <rdfs:Class rdf:ID="&tap;Country">
  <rdfs:label xml:lang="en">Country</rdfs:label>
  <rdfs:subClassOf rdf:resource="&tap;GeoPoliticalRegion"/>
 </rdfs:Class>
 <rdf:Property rdf:ID="hasEconomyOverview">
  <rdfs:label xml:lang="en">has Economy Overview</rdfs:label>
  <rdfs:domain rdf:resource="&tap;Country"/>
  <rdfs:range rdf:resource="&ftmeta;FreeText"/>
 </rdf:Property>
 <rdf:Property rdf:ID="hasGDP">
  <rdfs:label xml:lang="en">has GDP</rdfs:label>
  <rdfs:domain rdf:resource="&tap;Country"/>
  <rdfs:range rdf:resource="&xsd;decimal"/>
 </rdf:Property>
 <rdf:Property rdf:ID="hasRealGrowthRateGDP">
  <rdfs:label xml:lang="en">has Real Growth Rate GDP</rdfs:label>
```

```
  <rdfs:domain rdf:resource="&tap;Country"/>
  <rdfs:range rdf:resource="&xsd;decimal"/>
</rdf:Property>
<rdf:Property rdf:ID="hasPerCapitaGDP">
  <rdfs:label xml:lang="en">has Per Capita GDP</rdfs:label>
  <rdfs:domain rdf:resource="&tap;Country"/>
  <rdfs:range rdf:resource="&xsd;decimal"/>
</rdf:Property>
<rdf:Property rdf:ID="hasBudgetExpenditures">
  <rdfs:label xml:lang="en">has Budget Expenditures</rdfs:label>
  <rdfs:domain rdf:resource="&tap;Country"/>
  <rdfs:range rdf:resource="&xsd;decimal"/>
</rdf:Property>
<rdf:Property rdf:ID="hasBudgetRevenues">
  <rdfs:label xml:lang="en">has Budget Revenues</rdfs:label>
  <rdfs:domain rdf:resource="&tap;Country"/>
  <rdfs:range rdf:resource="&xsd;decimal"/>
</rdf:Property>
<rdf:Property rdf:ID="hasExternalDebt">
  <rdfs:label xml:lang="en">has External Debt</rdfs:label>
  <rdfs:domain rdf:resource="&tap;Country"/>
  <rdfs:range rdf:resource="&xsd;decimal"/>
</rdf:Property>
<rdf:Property rdf:ID="hasExportPartnerships">
  <rdfs:label xml:lang="en">has Export Partnerships</rdfs:label>
  <rdfs:domain rdf:resource="&tap;Country"/>
  <rdfs:range rdf:resource="&tap;Literal"/>
</rdf:Property>
<rdf:Property rdf:ID="hasExports">
  <rdfs:label xml:lang="en">has Exports</rdfs:label>
  <rdfs:domain rdf:resource="&tap;Country"/>
  <rdfs:range rdf:resource="&xsd;decimal"/>
</rdf:Property>
<rdf:Property rdf:ID="hasImports">
  <rdfs:label xml:lang="en">has Imports</rdfs:label>
  <rdfs:domain rdf:resource="&tap;Country"/>
  <rdfs:range rdf:resource="&xsd;decimal"/>
</rdf:Property>
<rdf:Property rdf:ID="hasImportPartnerships">
  <rdfs:label xml:lang="en">has Import Partnerships</rdfs:label>
  <rdfs:domain rdf:resource="&tap;Country"/>
  <rdfs:range rdf:resource="&tap;Literal"/>
</rdf:Property>
<rdf:Property rdf:ID="hasGDPEstimationdate">
  <rdfs:label xml:lang="en">has GDPEstimationdate</rdfs:label>
  <rdfs:domain rdf:resource="&tap;Country"/>
  <rdfs:range rdf:resource="&xsd;integer"/>
</rdf:Property>
<rdf:Property rdf:ID="hasRealGrowthRateGDPEstimationDate">
```

<rdfs:label xml:lang="en">has Real Growth Rate GDPEstimation Date</rdfs:label>
<rdfs:domain rdf:resource="&tap;Country"/>
<rdfs:range rdf:resource="&xsd;integer"/>
</rdf:Property>
<rdf:Property rdf:ID="hasPerCapitaGDPEstimationDate">
<rdfs:label xml:lang="en">has Per Capita GDPEstimation Date</rdfs:label>
<rdfs:domain rdf:resource="&tap;Country"/>
<rdfs:range rdf:resource="&xsd;integer"/>
</rdf:Property>
<rdf:Property rdf:ID="hasBudgetEstimationDate">
<rdfs:label xml:lang="en">has Budget Estimation Date</rdfs:label>
<rdfs:domain rdf:resource="&tap;Country"/>
<rdfs:range rdf:resource="&xsd;integer"/>
</rdf:Property>
<rdf:Property rdf:ID="hasExternalDebtEstimationDate">
<rdfs:label xml:lang="en">has External Debt Estimation Date</rdfs:label>
<rdfs:domain rdf:resource="&tap;Country"/>
<rdfs:range rdf:resource="&xsd;integer"/>
</rdf:Property>
<rdf:Property rdf:ID="hasLabel">
<rdfs:label xml:lang="en">has Label</rdfs:label>
<rdfs:domain rdf:resource="&tap;Country"/>
<rdfs:range rdf:resource="&tap;Literal"/>
</rdf:Property>
<rdf:Property rdf:ID="hasNationalityNounForm">
<rdfs:label xml:lang="en">has Nationality Noun Form</rdfs:label>
<rdfs:domain rdf:resource="&tap;Country"/>
<rdfs:range rdf:resource="&tap;Literal"/>
</rdf:Property>
<rdf:Property rdf:ID="hasPopulation">
<rdfs:label xml:lang="en">has Population</rdfs:label>
<rdfs:domain rdf:resource="&tap;Country"/>
<rdfs:range rdf:resource="&xsd;decimal"/>
</rdf:Property>
<rdf:Property rdf:ID="hasBirthRatePerThousand">
<rdfs:label xml:lang="en">has Birth Rate Per Thousand</rdfs:label>
<rdfs:domain rdf:resource="&tap;Country"/>
<rdfs:range rdf:resource="&xsd;decimal"/>
</rdf:Property>
<rdf:Property rdf:ID="hasDeathRatePerThousand">
<rdfs:label xml:lang="en">has Death Rate Per Thousand</rdfs:label>
<rdfs:domain rdf:resource="&tap;Country"/>
<rdfs:range rdf:resource="&xsd;decimal"/>
</rdf:Property>
<rdf:Property rdf:ID="hasEthnicGroups">
<rdfs:label xml:lang="en">has Ethnic Groups</rdfs:label>
<rdfs:domain rdf:resource="&tap;Country"/>
<rdfs:range rdf:resource="&tap;Literal"/>

```
        </rdf:Property>
        <rdf:Property rdf:ID="populationHasReligions">
          <rdfs:label xml:lang="en">population Has Religions</rdfs:label>
          <rdfs:domain rdf:resource="&tap;Country"/>
          <rdfs:range rdf:resource="&tap;Literal"/>
        </rdf:Property>
        <rdf:Property rdf:ID="hasLanguages">
          <rdfs:label xml:lang="en">has Languages</rdfs:label>
          <rdfs:domain rdf:resource="&tap;Country"/>
          <rdfs:range rdf:resource="&tap;Literal"/>
        </rdf:Property>
        <rdf:Property rdf:ID="hasLiteracyRate">
          <rdfs:label xml:lang="en">has Literacy Rate</rdfs:label>
          <rdfs:domain rdf:resource="&tap;Country"/>
          <rdfs:range rdf:resource="&xsd;decimal"/>
        </rdf:Property>
        <rdf:Property rdf:ID="hasMaleLiteracyRate">
          <rdfs:label xml:lang="en">has Male Literacy Rate</rdfs:label>
          <rdfs:domain rdf:resource="&tap;Country"/>
          <rdfs:range rdf:resource="&xsd;decimal"/>
        </rdf:Property>
        <rdf:Property rdf:ID="hasFemaleLiteracyRate">
          <rdfs:label xml:lang="en">has Female Literacy Rate</rdfs:label>
          <rdfs:domain rdf:resource="&tap;Country"/>
          <rdfs:range rdf:resource="&xsd;decimal"/>
        </rdf:Property>
        <rdf:Property rdf:ID="hasChildMortalityPerThousand">
          <rdfs:label xml:lang="en">has Child Mortality Per Thousand</rdfs:label>
          <rdfs:domain rdf:resource="&tap;Country"/>
          <rdfs:range rdf:resource="&xsd;decimal"/>
        </rdf:Property>
        <rdf:Property rdf:ID="hasChildMortalityPerThousandEstimationDate">
          <rdfs:label xml:lang="en">has Child Mortality Per Thousand Estimation
Date</rdfs:label>
          <rdfs:domain rdf:resource="&tap;Country"/>
          <rdfs:range rdf:resource="&xsd;integer"/>
        </rdf:Property>
        <rdf:Property rdf:ID="hasLifeExpectancy">
          <rdfs:label xml:lang="en">has Life Expectancy</rdfs:label>
          <rdfs:domain rdf:resource="&tap;Country"/>
          <rdfs:range rdf:resource="&xsd;decimal"/>
        </rdf:Property>
        <rdf:Property rdf:ID="hasLifeExpectancyEstimationDate">
          <rdfs:label xml:lang="en">has Life Expectancy Estimation
Date</rdfs:label>
          <rdfs:domain rdf:resource="&tap;Country"/>
          <rdfs:range rdf:resource="&xsd;integer"/>
        </rdf:Property>
        <rdf:Property rdf:ID="hasFemaleLifeExpectancy">
          <rdfs:label xml:lang="en">has Female Life Expectancy</rdfs:label>
```

```
            <rdfs:domain rdf:resource="&tap;Country"/>
            <rdfs:range rdf:resource="&xsd;decimal"/>
          </rdf:Property>
          <rdf:Property rdf:ID="hasMaleLifeExpectancy">
            <rdfs:label xml:lang="en">has Male Life Expectancy</rdfs:label>
            <rdfs:domain rdf:resource="&tap;Country"/>
            <rdfs:range rdf:resource="&xsd;decimal"/>
          </rdf:Property>
          <rdf:Property rdf:ID="hasAgriculturePercentLaborForce">
            <rdfs:label xml:lang="en">has Agriculture Percent Labor
Force</rdfs:label>
            <rdfs:domain rdf:resource="&tap;Country"/>
            <rdfs:range rdf:resource="&xsd;decimal"/>
          </rdf:Property>
          <rdf:Property rdf:ID="hasServicesPercentLaborForce">
            <rdfs:label xml:lang="en">has Services Percent Labor Force</rdfs:label>
            <rdfs:domain rdf:resource="&tap;Country"/>
            <rdfs:range rdf:resource="&xsd;decimal"/>
          </rdf:Property>
          <rdf:Property rdf:ID="hasLaborForceEstimationDate">
            <rdfs:label xml:lang="en">has Labor Force Estimation Date</rdfs:label>
            <rdfs:domain rdf:resource="&tap;Country"/>
            <rdfs:range rdf:resource="&xsd;integer"/>
          </rdf:Property>
          <rdf:Property rdf:ID="hasIndustryPercentLaborForce">
            <rdfs:label xml:lang="en">has Industry Percent Labor Force</rdfs:label>
            <rdfs:domain rdf:resource="&tap;Country"/>
            <rdfs:range rdf:resource="&xsd;decimal"/>
          </rdf:Property>
          <rdf:Property rdf:ID="hasLaborForce">
            <rdfs:label xml:lang="en">has Labor Force</rdfs:label>
            <rdfs:domain rdf:resource="&tap;Country"/>
            <rdfs:range rdf:resource="&xsd;decimal"/>
          </rdf:Property>
          <rdf:Property rdf:ID="hasUnemployedPercentage">
            <rdfs:label xml:lang="en">has Unemployed Percentage</rdfs:label>
            <rdfs:domain rdf:resource="&tap;Country"/>
            <rdfs:range rdf:resource="&xsd;decimal"/>
          </rdf:Property>
          <rdf:Property rdf:ID="hasUnemployedPercentageEstimationDate">
            <rdfs:label xml:lang="en">has Unemployed Percentage Estimation
Date</rdfs:label>
            <rdfs:domain rdf:resource="&tap;Country"/>
            <rdfs:range rdf:resource="&xsd;integer"/>
          </rdf:Property>
          <rdf:Property rdf:ID="hasPopulationEstimationDate">
            <rdfs:label xml:lang="en">has Population Estimation Date</rdfs:label>
            <rdfs:domain rdf:resource="&tap;Country"/>
            <rdfs:range rdf:resource="&xsd;integer"/>
          </rdf:Property>
```

```
<rdf:Property rdf:ID="hasPopulationGrowthRate">
  <rdfs:label xml:lang="en">has Population Growth Rate</rdfs:label>
  <rdfs:domain rdf:resource="&tap;Country"/>
  <rdfs:range rdf:resource="&xsd;decimal"/>
</rdf:Property>
<rdf:Property rdf:ID="hasPopulationGrowthRateEstimationDate">
  <rdfs:label xml:lang="en">has Population Growth Rate Estimation Date</rdfs:label>
  <rdfs:domain rdf:resource="&tap;Country"/>
  <rdfs:range rdf:resource="&xsd;integer"/>
</rdf:Property>
<rdf:Property rdf:ID="hasAreaComparitive">
  <rdfs:label xml:lang="en">has Area Comparitive</rdfs:label>
  <rdfs:domain rdf:resource="&tap;Country"/>
  <rdfs:range rdf:resource="&ftmeta;FreeText"/>
</rdf:Property>
<rdf:Property rdf:ID="hasArea">
  <rdfs:label xml:lang="en">has Area</rdfs:label>
  <rdfs:domain rdf:resource="&tap;Country"/>
  <rdfs:range rdf:resource="&xsd;decimal"/>
</rdf:Property>
<rdf:Property rdf:ID="terrain">
  <rdfs:label xml:lang="en">terrain</rdfs:label>
  <rdfs:domain rdf:resource="&tap;Country"/>
  <rdfs:range rdf:resource="&ftmeta;FreeText"/>
</rdf:Property>
<rdf:Property rdf:ID="climate">
  <rdfs:label xml:lang="en">climate</rdfs:label>
  <rdfs:domain rdf:resource="&tap;Country"/>
  <rdfs:range rdf:resource="&ftmeta;FreeText"/>
</rdf:Property>
<rdf:Property rdf:ID="hasArableLand">
  <rdfs:label xml:lang="en">has Arable Land</rdfs:label>
  <rdfs:domain rdf:resource="&tap;Country"/>
  <rdfs:range rdf:resource="&xsd;decimal"/>
</rdf:Property>
<rdf:Property rdf:ID="hasPermanentCropLand">
  <rdfs:label xml:lang="en">has Permanent Crop Land</rdfs:label>
  <rdfs:domain rdf:resource="&tap;Country"/>
  <rdfs:range rdf:resource="&xsd;decimal"/>
</rdf:Property>
<rdf:Property rdf:ID="longFormName">
  <rdfs:label xml:lang="en">long Form Name</rdfs:label>
  <rdfs:domain rdf:resource="&tap;Country"/>
  <rdfs:range rdf:resource="&tap;Literal"/>
</rdf:Property>
<rdf:Property rdf:ID="hasCapitalCity">
  <rdfs:label xml:lang="en">has Capital City</rdfs:label>
  <rdfs:domain rdf:resource="&tap;Country"/>
  <rdfs:range rdf:resource="&tap;Literal"/>
```

```
  </rdf:Property>
  <rdf:Property rdf:ID="hasLocation">
    <rdfs:label xml:lang="en">has Location</rdfs:label>
    <rdfs:domain rdf:resource="&tap;Country"/>
    <rdfs:range rdf:resource="&ftmeta;FreeText"/>
  </rdf:Property>
  <rdf:Property rdf:ID="hasGeographyDescription">
    <rdfs:label xml:lang="en">has Geography Description</rdfs:label>
    <rdfs:domain rdf:resource="&tap;Country"/>
    <rdfs:range rdf:resource="&ftmeta;FreeText"/>
  </rdf:Property>
  <rdf:Property rdf:ID="hasCoastlineKilometers">
    <rdfs:label xml:lang="en">has Coastline Kilometers</rdfs:label>
    <rdfs:domain rdf:resource="&tap;Country"/>
    <rdfs:range rdf:resource="&xsd;decimal"/>
  </rdf:Property>
  <rdf:Property rdf:ID="hasNaturalResources">
    <rdfs:label xml:lang="en">has Natural Resources</rdfs:label>
    <rdfs:domain rdf:resource="&tap;Country"/>
    <rdfs:range rdf:resource="&tap;Literal"/>
  </rdf:Property>
  <rdf:Property rdf:ID="image">
    <rdfs:label xml:lang="en">image</rdfs:label>
    <rdfs:domain rdf:resource="&tap;Country"/>
    <rdfs:range rdf:resource="&xsd;anyUri"/>
  </rdf:Property>
</rdf:RDF>
```

2. CIA World Fact Book DAML ontology (fragment)

```
<?xml version='1.0' encoding='ISO-8859-1'?>
<!DOCTYPE rdf:RDF [
  <!ENTITY rdf "http://www.w3.org/1999/02/22-rdf-syntax-ns#">
  <!ENTITY rdfs "http://www.w3.org/2000/01/rdf-schema#">
  <!ENTITY daml "http://www.w3.org/2002/07/owl#">
  <!ENTITY xsd "http://www.w3.org/2001/XMLSchema#">
  <!ENTITY fips-countries-ont "http://www.daml.org/2001/09/countries/fips-10-4-ont">
  <!ENTITY factbook "http://www.daml.org/2003/09/factbook/factbook-ont#">
]>

<rdf:RDF
  xmlns:rdf = "&rdf;"
  xmlns:rdfs ="&rdfs;"
  xmlns:owl ="&daml;"
  xmlns:factbook="&factbook;"
  xml:base="http://www.daml.org/2003/09/factbook/factbook-ont">

<owl:Ontology rdf:about="">
  <owl:versionInfo>$Id: factbook-ont.owl,v 1.9 2005/09/20 16:29:22 mdean Exp $</owl:versionInfo>
  <owl:imports rdf:resource="&fips-countries-ont;"/>
  <rdfs:comment>CIA World Fact Book (2003) Ontology</rdfs:comment>
  <rdfs:comment>see http://www.cia.gov/cia/publications/factbook/</rdfs:comment>
</owl:Ontology>

<rdf:Description rdf:about="&fips-countries-ont;#Country">
  <rdfs:comment>we add properties to existing Country objects rather than creating new ones</rdfs:comment>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#background"/>
      <owl:allValuesFrom rdf:resource="&xsd;string"/>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#location"/>
      <owl:allValuesFrom rdf:resource="&xsd;string"/>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#geographicCoordinates"/>
      <owl:allValuesFrom rdf:resource="#LatLon"/>
```

```
  </owl:Restriction>
</rdfs:subClassOf>
<rdfs:subClassOf>
  <owl:Restriction>
    <owl:onProperty rdf:resource="#mapReferences"/>
    <owl:allValuesFrom rdf:resource="&xsd;string"/>
  </owl:Restriction>
</rdfs:subClassOf>
<rdfs:subClassOf>
  <owl:Restriction>
    <owl:onProperty rdf:resource="#totalArea"/>
    <owl:allValuesFrom rdf:resource="&xsd;decimal"/>
    <factbook:units>sq km</factbook:units>
  </owl:Restriction>
</rdfs:subClassOf>
<rdfs:subClassOf>
  <owl:Restriction>
    <owl:onProperty rdf:resource="#landArea"/>
    <owl:allValuesFrom rdf:resource="&xsd;decimal"/>
    <factbook:units>sq km</factbook:units>
  </owl:Restriction>
</rdfs:subClassOf>
<rdfs:subClassOf>
  <owl:Restriction>
    <owl:onProperty rdf:resource="#waterArea"/>
    <owl:allValuesFrom rdf:resource="&xsd;decimal"/>
    <factbook:units>sq km</factbook:units>
  </owl:Restriction>
</rdfs:subClassOf>
<rdfs:subClassOf>
  <owl:Restriction>
    <owl:onProperty rdf:resource="#comparativeArea"/>
    <owl:allValuesFrom rdf:resource="&xsd;string"/>
  </owl:Restriction>
</rdfs:subClassOf>
<rdfs:subClassOf>
  <owl:Restriction>
    <owl:onProperty rdf:resource="#landBoundaries"/>
    <owl:allValuesFrom rdf:resource="&xsd;decimal"/>
    <factbook:units>km</factbook:units>
  </owl:Restriction>
</rdfs:subClassOf>
<rdfs:subClassOf>
  <owl:Restriction>
    <owl:onProperty rdf:resource="#border"/>
    <owl:allValuesFrom rdf:resource="#Border"/>
  </owl:Restriction>
</rdfs:subClassOf>
<rdfs:subClassOf>
  <owl:Restriction>
```

```
    <owl:onProperty rdf:resource="#coastline"/>
    <owl:allValuesFrom rdf:resource="&xsd;decimal"/>
    <factbook:units>km</factbook:units>
  </owl:Restriction>
</rdfs:subClassOf>
<rdfs:subClassOf>
  <owl:Restriction>
    <owl:onProperty rdf:resource="#maritimeClaim"/>
    <owl:allValuesFrom rdf:resource="&xsd;integer"/>
    <factbook:units>nautical miles</factbook:units>
  </owl:Restriction>
</rdfs:subClassOf>
<rdfs:subClassOf>
  <owl:Restriction>
    <owl:onProperty rdf:resource="#climate"/>
    <owl:allValuesFrom rdf:resource="&xsd;string"/>
  </owl:Restriction>
</rdfs:subClassOf>
<rdfs:subClassOf>
  <owl:Restriction>
    <owl:onProperty rdf:resource="#terrain"/>
    <owl:allValuesFrom rdf:resource="&xsd;string"/>
  </owl:Restriction>
</rdfs:subClassOf>
<rdfs:subClassOf>
  <owl:Restriction>
    <owl:onProperty rdf:resource="#highestPoint"/>
    <owl:allValuesFrom rdf:resource="#ElevationExtreme"/>
  </owl:Restriction>
</rdfs:subClassOf>
<rdfs:subClassOf>
  <owl:Restriction>
    <owl:onProperty rdf:resource="#lowestPoint"/>
    <owl:allValuesFrom rdf:resource="#ElevationExtreme"/>
  </owl:Restriction>
</rdfs:subClassOf>
<rdfs:subClassOf>
  <owl:Restriction>
    <owl:onProperty rdf:resource="#naturalResource"/>
    <owl:allValuesFrom rdf:resource="#NaturalResource"/>
  </owl:Restriction>
</rdfs:subClassOf>
<rdfs:subClassOf>
  <owl:Restriction>
    <owl:onProperty rdf:resource="#landUse"/>
    <owl:allValuesFrom rdf:resource="&xsd;float"/>
    <factbook:units>percent</factbook:units>
  </owl:Restriction>
</rdfs:subClassOf>
<rdfs:subClassOf>
```

```
  <owl:Restriction>
    <owl:onProperty rdf:resource="#irrigatedLand"/>
    <owl:allValuesFrom rdf:resource="&xsd;float"/>
    <factbook:units>sq km</factbook:units>
  </owl:Restriction>
</rdfs:subClassOf>
<rdfs:subClassOf>
  <owl:Restriction>
    <owl:onProperty rdf:resource="#naturalHazard"/>
    <owl:allValuesFrom rdf:resource="&xsd;string"/>
  </owl:Restriction>
</rdfs:subClassOf>
<rdfs:subClassOf>
  <owl:Restriction>
    <owl:onProperty rdf:resource="#environmentalIssue"/>
    <owl:allValuesFrom rdf:resource="&xsd;string"/>
  </owl:Restriction>
</rdfs:subClassOf>
<rdfs:comment>environment - international agreements</rdfs:comment>
<rdfs:subClassOf>
  <owl:Restriction>
    <owl:onProperty rdf:resource="#population"/>
    <owl:allValuesFrom rdf:resource="&xsd;integer"/>
    <factbook:units>people</factbook:units>
  </owl:Restriction>
</rdfs:subClassOf>
<rdfs:comment>age structure</rdfs:comment>
<rdfs:subClassOf>
  <owl:Restriction>
    <owl:onProperty rdf:resource="#medianAgeTotal"/>
    <owl:allValuesFrom rdf:resource="&xsd;float"/>
    <factbook:units>years</factbook:units>
  </owl:Restriction>
</rdfs:subClassOf>
<rdfs:subClassOf>
  <owl:Restriction>
    <owl:onProperty rdf:resource="#medianAgeMale"/>
    <owl:allValuesFrom rdf:resource="&xsd;float"/>
    <factbook:units>years</factbook:units>
  </owl:Restriction>
</rdfs:subClassOf>
<rdfs:subClassOf>
  <owl:Restriction>
    <owl:onProperty rdf:resource="#medianAgeFemale"/>
    <owl:allValuesFrom rdf:resource="&xsd;float"/>
    <factbook:units>years</factbook:units>
  </owl:Restriction>
</rdfs:subClassOf>
<rdfs:subClassOf>
  <owl:Restriction>
```

```
    <owl:onProperty rdf:resource="#populationGrowthRate"/>
    <owl:allValuesFrom rdf:resource="&xsd;float"/>
    <factbook:units>percent/year</factbook:units>
  </owl:Restriction>
</rdfs:subClassOf>
<rdfs:subClassOf>
  <owl:Restriction>
    <owl:onProperty rdf:resource="#birthRate"/>
    <owl:allValuesFrom rdf:resource="&xsd;float"/>
    <factbook:units>births/1000 population</factbook:units>
  </owl:Restriction>
</rdfs:subClassOf>
<rdfs:subClassOf>
  <owl:Restriction>
    <owl:onProperty rdf:resource="#deathRate"/>
    <owl:allValuesFrom rdf:resource="&xsd;float"/>
    <factbook:units>deaths/1000 population</factbook:units>
  </owl:Restriction>
</rdfs:subClassOf>
<rdfs:subClassOf>
  <owl:Restriction>
    <owl:onProperty rdf:resource="#netMigrationRate"/>
    <owl:allValuesFrom rdf:resource="&xsd;float"/>
    <factbook:units>migrants/1000 population</factbook:units>
  </owl:Restriction>
</rdfs:subClassOf>
<rdfs:subClassOf>
  <owl:Restriction>
    <owl:onProperty rdf:resource="#sexRatio"/>
    <owl:allValuesFrom rdf:resource="#SexRatioBreakdown"/>
  </owl:Restriction>
</rdfs:subClassOf>
<rdfs:subClassOf>
  <owl:Restriction>
    <owl:onProperty rdf:resource="#infantMortalityRateTotal"/>
    <owl:allValuesFrom rdf:resource="&xsd;float"/>
    <factbook:units>deaths/1000 live births</factbook:units>
  </owl:Restriction>
</rdfs:subClassOf>
<rdfs:subClassOf>
  <owl:Restriction>
    <owl:onProperty rdf:resource="#infantMortalityRateFemale"/>
    <owl:allValuesFrom rdf:resource="&xsd;float"/>
    <factbook:units>deaths/1000 live births</factbook:units>
  </owl:Restriction>
</rdfs:subClassOf>
<rdfs:subClassOf>
  <owl:Restriction>
    <owl:onProperty rdf:resource="#infantMortalityRateMale"/>
    <owl:allValuesFrom rdf:resource="&xsd;float"/>
```

```
    <factbook:units>deaths/1000 live births</factbook:units>
  </owl:Restriction>
</rdfs:subClassOf>
<rdfs:comment>life expectancy at birth</rdfs:comment>
<rdfs:subClassOf>
  <owl:Restriction>
    <owl:onProperty rdf:resource="#totalFertilityRate"/>
    <owl:allValuesFrom rdf:resource="&xsd;float"/>
    <factbook:units>children born/woman</factbook:units>
  </owl:Restriction>
</rdfs:subClassOf>
<rdfs:comment>HIV/AIDS</rdfs:comment>
<rdfs:subClassOf>
  <owl:Restriction>
    <owl:onProperty rdf:resource="#nationalityNoun"/>
    <owl:allValuesFrom rdf:resource="&xsd;string"/>
  </owl:Restriction>
</rdfs:subClassOf>
<rdfs:subClassOf>
  <owl:Restriction>
    <owl:onProperty rdf:resource="#nationalityAdjective"/>
    <owl:allValuesFrom rdf:resource="&xsd;string"/>
  </owl:Restriction>
</rdfs:subClassOf>
```

……

3. Translation rules from CIA World Fact Book DAML ontology to TAP ontology

TAP attribute name	CIA DAML attribute name (path)	Data type
haseconomyoverview	economyOverview	free_text
hasexports	exports	double
hasgdp	grossDomesticProduct	double
hasgdpestimationdate		
hasimports	imports	double
haslabel	conventionalShortCountryName	nominal
hasexportpartnerships	exportPartner->country->conventionalShortCountryName	nominal
haspercapitagdp	grossDomesticProductPerCapita	double
hasrealgrowthrategdp	grossDomesticProductRealGrowth	double
hasimportpartnerships	importPartner->country->conventionalShortCountryName	nominal
haspercapitagdpestimationdate		
hasrealgrowthrategdpestimationdate		
hasexternaldebt	externalDebt	double
hasexternaldebtestimationdate		
hasbudgetexpenditures	budgetExpenditures	double
hasbudgetrevenues	budgetRevenues	double
hasagriculturepercentlaborforce		
haschildmortalityperthousand	infantMortalityRateTotal	double
haschildmortalityperthousandestimationdate		
hasethnicgroups	ethnicGroup->ethnicGroup->name	nominal
haslaborforce	laborForce	double
haslaborforceestimationdate	double	
haslanguages	language->language->name	nominal
haslifeexpectancy		
haslifeexpectancyestimationdate		
hasnationalitynounform	nationalityNoun	nominal
haspopulation	population	double
haspopulationestimationdate		
haspopulationgrowthrate	populationGrowthRate	double
haspopulationgrowthrateestimationdate		
hasservicespercentlaborforce		
populationhasreligions	religion->religion->name	nominal
hasindustrypercentlaborforce		
hasliteracyrate	literacyTotal	double
hasfemalelifeexpectancy		
hasmalelifeexpectancy		
hasfemaleliteracyrate	literacyFemale	double
hasmaleliteracyrate	literacyMale	double
hasbirthrateperthousand	birthRate	double
hasdeathrateperthousand	deathRate	double
hasunemployedpercentage	unemploymentRate	double
hasunemployedpercentageestimationdate		
climate	climate	free_text
hasarableland	arableLand	double
hasarea	totalArea	double
hasareacomparitive	comparativeArea	free_text
hascapitalcity	capital->name	nominal

hascoastlinekilometers	coastline	double
hasgeographydescription		
haslocation	location	free_text
hasnaturalresources	naturalResource->name	nominal
haspermanentcropland	permanentCrops	double
longformname	conventionalLongCountryName	nominal
Terrain	terrain	free_text

Appendix C – Work plan Gantt diagram