

Using Multiple Ontologies in Information Extraction

Daya C. Wimalasuriya
Computer and Information Science
University of Oregon, USA
dayacw@cs.uoregon.edu

Dejing Dou
Computer and Information Science
University of Oregon, USA
dou@cs.uoregon.edu

ABSTRACT

Ontology-Based Information Extraction (OBIE) has recently emerged as a subfield of Information Extraction (IE). Here, ontologies - which provide formal and explicit specifications of conceptualizations - play a crucial role in the information extraction process. Several OBIE systems have been implemented previously but all of them use a single ontology although multiple ontologies have been designed for many domains. We have studied the theoretical basis for using multiple ontologies in information extraction and have developed information extraction systems that use them. These systems investigate the two major scenarios for having multiple ontologies for the same domain: specializing in subdomains and providing different perspectives. The domain of universities has been used for the former scenario through a corpus collected from university websites. For the latter, the domain of terrorist attacks and a corpus used by a previous Message Understanding Conference (MUC) have been used. The results from these two case studies indicate that using multiple ontologies in information extraction has led to a clear improvement in performance measures.

Categories and Subject Descriptors

H.4.0 [Information Systems Applications]: General

General Terms

Theory, Experimentation

Keywords

Information Extraction, Ontologies, Mappings

1. INTRODUCTION

1.1 Ontology-Based Information Extraction

Information extraction (IE), which is often considered a subfield of natural language processing (NLP), aims to recognize and extract certain types of information from natural

language text [20]. Generally, the types of information extracted by IE systems are related to a particular domain such as business organizations, genes or terrorist attacks. Limiting the focus to certain types of information in this manner makes information extraction different from the much harder problem of natural language understanding, which attempts to logically interpret natural language.

Ontology-based information extraction (OBIE) is a recent development in the field of information extraction. Here, the general idea is to use an ontology to *guide* the information extraction process and to present the results. The concept of ontologies comes from the field of knowledge representation. An ontology is defined as *a formal and explicit specification of a shared conceptualization* [13, 22]. Normally, an ontology is specified for a particular domain. Such an ontology, often known as a domain ontology, formally and explicitly specifies the concepts and relationships in that domain.

Ontology-based information extraction attempts to make use of the formal and explicit specifications of an ontology in the information extraction process. This is generally achieved by using information extraction to retrieve instances and values related to the classes and properties of the ontology. For example, in the domain of business organizations, the IE process might discover companies and their important features such as the number of employees, field of business and the location of the head office as identified by the class that represents companies in the ontology and its properties. The extracted information is normally presented through the ontology itself using an ontology definition language. The Web Ontology Language (OWL) [2], which has emerged as the de facto standard for defining ontologies, is widely used for this purpose. Since the software agents of the Semantic Web [7] are expected to be able to deal with languages such as OWL, the output of an OBIE system can be considered accessible from the Semantic Web.

Several OBIE systems have been implemented in the last few years and even a workshop [5] has been organized on the field. Some of these systems are described in Section 2. For information extraction, they use different techniques such as classification, linguistic extraction rules expressed as regular expressions, gazetteers and web-based search. They also extract different components of an ontology such as instances, property values and classes. The text corpora used by them are also different as some of them can process any document from a given domain while others process documents from a specific source such as Wikipedia¹.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

CIKM'09, November 2-6, 2009, Hong Kong, China.

Copyright 2009 ACM 978-1-60558-512-3/09/11 ...\$10.00.

¹<http://www.wikipedia.org>

1.2 Use of Multiple Ontologies

All these OBIE systems use only *one* ontology for the information extraction process. But there is no rule that prevents an OBIE system from using more than one ontology to guide its information extraction process. For several domains, different ontologies have been developed. For example, one ontology repository² contains more than 10 ontologies for the tourism domain. In addition, issues related to the existence of multiple ontologies such as integrating them and discovering *mappings* between the concepts of different ontologies have become an active research area as evidenced by the research papers published on these topics [9, 18].

It can be seen that multiple ontologies developed for the same domain belong to one of the following scenarios.

1. **Specializing in sub-domains:** For example, in the domain of universities, several sub-domains can be identified such as North American universities, British universities, universities with a religious background, etc. For each of these sub-domains, specific ontologies can be developed by paying special attention to the concepts unique to it.
2. **Providing different perspectives:** For example, one ontology for the domain of marriages might define two classes named “Husband” and “Wife”, while another might define an object property named “isSpouseOf”.

Using multiple ontologies in OBIE is interesting because it has the potential to improve the information extraction process. The following are two important opportunities on this regard.

1. Possible improvement in recall:

Recall shows the number of correctly identified items as a percentage of the total number of correct items available. Recall and *Precision*, which shows the number of correctly identified items as a percentage of the total number of items identified, are the two main performance metrics used in information extraction.

When using multiple ontologies that provide different perspectives, it can be hypothesized that information extraction processes guided by concepts of different ontologies would make more extractions together than what is possible by a single ontology, thus resulting in a higher recall. For instance, in the marriage ontologies described above, extractions made based on the “isSpouseOf” property would capture homosexual marriages in addition to some heterosexual marriages while extractions based on “Husband” and “Wife” classes are likely to be more successful in retrieving instances of heterosexual marriages. Similarly, when using ontologies that specialize on particular sub-domains, each ontology can be expected to be more successful in making extractions in its own sub-domain. Hence, a set of specialized ontologies can be expected to make more correct extractions than what is possible under a common ontology.

If the resulting multi-ontology system is more accurate as a whole than the single-ontology systems, the precision would also increase. On the other hand if there is

some loss in accuracy when making more predictions, a drop in precision can be anticipated. We expect that greater improvements in recall would offset such losses.

2. Supporting multiple perspectives:

Since each ontology directly represents a particular conceptualization or a perspective of the domain in concern, using multiple ontologies implies that the system is capable of handling the perspectives related to each of the ontologies. This means that the output of the system can be used to answer queries based on different perspectives. For example, the output of an OBIE system for the marriage domain that uses both marriage ontologies described above can be used to answer different queries such as “Is person A a husband?” and “Who is person A’s spouse?”.

1.3 Challenges in Using Multiple Ontologies in OBIE

Even though the idea of using multiple ontologies in OBIE looks promising intuitively, the following challenges are encountered in this process.

1. Figuring out the theoretical basis for using multiple ontologies in information extraction:

It is necessary to study and formally represent ontologies and the relationship between information extraction techniques and ontologies in order to correctly use multiple ontologies in information extraction. Such an analysis should also separately address the two scenarios for having multiple ontologies in the same domain described above.

2. Finding suitable ontologies and mappings:

Although several ontologies are available for most domains from ontology repositories, randomly selecting some of such ontologies for a multiple-ontology IE system would not be a good practice. Some ontologies will contain only a few concepts while others will be more detailed and some will be under construction. Hence, a careful selection will have to be made on what ontologies to use in the OBIE system. A related issue is the discovery of mappings between the concepts of the selected ontologies. This can be done manually or through the use of a mapping discovery tool. Either way, it would be necessary to verify that the mappings are correct before using them in the system.

1.4 Introduction to Our Work

The focus of the work presented in this paper is exploring the above mentioned opportunities and challenges in using multiple ontologies in information extraction. In order to achieve this objective, we first studied the theoretical basis for using multiple ontologies in information extraction. We found it interesting that there is no common agreement in the field of ontology-based information extraction on how to formally represent ontologies and the relationship between ontologies and information extraction. Therefore we started by developing such a representation for single-ontology OBIE systems based on existing ideas and then extended it to account for multiple ontologies.

Based on this theoretical framework, we then developed two OBIE systems that use multiple ontologies. One of these

²<http://www.daml.org/ontologies/keyword.html>

systems was developed for the university domain and it uses two ontologies specializing on sub-domains. The other system uses two ontologies that provide different perspectives on the terrorism domain. These two systems were compared against single-ontology IE systems for the same domains. The obtained results support our hypothesis that the use of multiple ontologies would improve the performance of information extraction systems.

The rest of the paper is organized as follows. Section 2 presents the details of some other ontology-based information extraction systems. Section 3 presents the theoretical framework for using multiple ontologies in information extraction. The details of the two case studies mentioned above, including the results, are shown in Section 4. We discuss the implications of our findings in Section 5 and provide concluding remarks in Section 6.

2. RELATED WORK

Ontology-Based Information Extraction has recently received a lot of attention from researchers mainly because of its relationship with the Semantic Web. It has been pointed out that OBIE systems can be used to create semantic contents for the Semantic Web from natural language text [10]. It should be noted that creation of such contents have been quite slow despite the fact that the success of the Semantic Web relies heavily on them. In addition, it has been stated that OBIE can be used to evaluate the quality of ontologies and to improve them [15].

Ontology-Based Information Extraction systems whose details have been published recently include Kylin [24], C-PANKOW [11], SOBA [8] and the implementations by Saggion et al. [21] and Li and Bontcheva [17]. We briefly describe the functionality of two such systems that are representative of many OBIE systems below.

The Kylin system [24] extracts information from a set of Wikipedia pages. The ontology used by the system is constructed by combining the information in infoboxes of Wikipedia pages (which present a tabular summary of the object described in a page) with concepts from WordNet lexical semantic database [4]. This task is performed by a component named Kylin Ontology Generator (KOG). Information extraction is basically performed as a two step process that relies on classification. The first classifier predicts which attribute values are contained in a given sentence. This classifier uses Maximum Entropy model using a variety of features including bag of words and Part-Of-Speech (POS) tags. The second classifier uses the CRF model with a wide variety of features to extract the attribute values from sentences. Kylin has performed well when there are enough training examples but has not worked well for “sparse classes” for which there are very few examples. The authors have applied three independent techniques to rectify this situation including adding training examples from the Web.

The implementation by Saggion et al. [21] uses linguistic extraction rules and gazetteers to extract information from a set of documents in the domain of business intelligence. Linguistic extraction rules use patterns expressed as regular expressions to make extractions. For example, the expression `(watched|seen) <NP>`, where `<NP>` denotes a noun phrase, might capture the names of movies (represented by the noun phrase) in a set of documents. Gazetteers on the other hand simply list the individual entities of a particular category and strings matching this list are recognized as

instances of the respective class. Saggion et al. [21] have used an ontology that has been developed as a part of the “Multi-Industry Semantic-Based Next Generation Business Intelligence (MUSING)” project. The system has been implemented using the General Architecture for Text Engineering (GATE) [1] and it has shown impressive results in terms of precision and recall.

3. THEORETICAL BASIS

3.1 Using a Single Ontology

An ontology consists of several components such as classes, properties (including both datatype properties and object properties), individuals (also known as instances and objects), property values of individuals and constraints. The W3C specification [2] defines the components supported by OWL. OWL is based on description logic.

In OBIE systems, information extraction techniques are normally used to extract individuals of classes and property values for individuals. For example, an OBIE system that uses a geopolitical ontology might identify “France” as an individual of the “Country” class and extract “Paris” as its property value for the object property “capital” (and identify “Paris” as an individual for the “City” class). Hence an OBIE system can be defined as a set of extractors each attempting to identify individuals of a given class or property values of a given property. Formally, this can be presented as follows.

Definition *Ontology*: An ontology O is a quintuple, $O = (C, P, I, V, A)$ where $C, P, I, V,$ and A are the sets of classes, properties, individuals, property values and other axioms (such as constraints) respectively.

Definition *Ontology-Based Information Extraction System*: An OBIE for the ontology O (as defined above), $I(O)$ is a set of n extractors as follows.

$$I(O) = \{E(O, X_1), E(O, X_2), \dots, E(O, X_n)\}$$

where $\forall i (1 \leq i \leq n)$, $X_i \in C$ or $X_i \in P$. For a given corpus D , each extractor $E(O, X_i)$ would make a set of extractions $R(E(O, X_i), D)$, which according to its predictions are either individuals or property values. (It should be noted that some of these predictions may be incorrect.)

We denote the actual individuals and property values found in D (often known as the *gold standard* or the *key*) by $k(I, D)$ and $k(V, D)$ respectively. It is assumed that all these actual individuals and property values are included in I and V . Formally,

$$k(I, D) \subset I \text{ and } k(V, D) \subset V$$

Based on these definitions, we can obtain formulae for *precision* and *recall* of the information extraction system (denoted by $P(I(O))$ and $R(I(O))$ respectively).

$$P(I(O)) = \frac{|\bigcup_{i=1}^n R(E(O, X_i), D) \cap \{k(I, D) \cup k(V, D)\}|}{|\bigcup_{i=1}^n R(E(O, X_i), D)|}$$

$$R(I(O)) = \frac{|\bigcup_{i=1}^n R(E(O, X_i), D) \cap \{k(I, D) \cup k(V, D)\}|}{|\{k(I, D) \cup k(V, D)\}|}$$

There appears to be no consensus on whether information extractors ($E(O, X_i)$ s) should be a part of the ontology or not. Some authors have argued that these should be considered a part of the ontology when linguistic rules are used as the information extraction technique [12, 19]. The terms

extraction ontology [12] and concrete ontology [19] have been proposed for an ontology that contains such rules.

However, we do not subscribe to the view that such linguistic rules should be considered a part of an ontology. Since these rules are essentially approximations, which are *known* to contain errors, objections can be raised that they are not accurate or formal enough to be included in an ontology. Moreover, we do not see any special feature in linguistic extraction rules that support their inclusion in ontologies. Other information extraction techniques such as classification and gazetteers perform the same task and they too could be included in an ontology in the same manner. But these techniques would also suffer from inaccuracies.

Therefore, we propose that information extractors should be treated as lying outside the ontology. But a developer of an ontology might find it useful to denote that extractors have been developed for some of its classes and properties. One way to accommodate this is to store the details of the information extractor in a URI (e.g., the linguistic extraction rules or the details of the classifiers used) and provide the URIs in the ontology. For classes this can be easily accommodated as a specific datatype property but more complex axioms would be needed in the case of extractors developed for properties.

Now we can move on to explore the use of multiple ontologies for information extraction. We do this separately for the two scenarios of using multiple ontologies for the same domain.

3.2 Multiple Ontologies Specializing on Subdomains

In this case, we have a generic (common) ontology O_c and a set of m specialized ontologies S given by,

$$S = \{O_1, O_2, \dots, O_m\}.$$

Let $O_c = (C_c, P_c, I_c, V_c, A_c)$ and

$$\forall i (1 \leq i \leq m), O_i = (C_i, P_i, I_i, V_i, A_i)$$

In performing information extraction on a given corpus D , the single-ontology IE system would use the common ontology O_c . The multiple-ontology system would use the set of specialized ontologies. For each of the K documents of the corpus D_j ($1 \leq j \leq K$), this system would have to determine which ontology to use. It would try to use the most suitable ontology for each document. This selection would be performed by an *ontology selector* component and it can be represented by a function os , which returns the number of the selected specialized ontology.

$$\forall j (1 \leq j \leq K), os(D_j) \in \{1, 2, \dots, m\}$$

The single-ontology system $I_s(O_c)$, consisting of n_s extractors, can be presented as follows using the definition of Section 3.1.

$$I_s(O_c) = \{E_s(O_c, X_1), E_s(O_c, X_2), \dots, E_s(O_c, X_{n_s})\}$$

where $\forall i (1 \leq i \leq n_s), X_i \in C_c$ or $X_i \in P_c$

The multiple-ontology system $I_m(S)$ would contain extractors for each ontology. Therefore, it can be defined as follows.

Definition Multiple-Ontology IE System: A multiple ontology IE system for the set S of ontologies

$$(S = \{O_1, O_2, \dots, O_m\}), I_m(S) \text{ is given by,}$$

$$I_m(S) = \{I_{m_1}(O_1), I_{m_2}(O_2), \dots, I_{m_m}(O_m)\}$$

where each $I_{m_i}(O_i)$ contains a set of r_i extractors for a single

ontology O_i as shown by,

$$\forall i (1 \leq i \leq m), I_{m_i}(O_i) = \{E_{m_i}(O_i, X_1), \dots, E_{m_i}(O_i, X_{r_i})\}$$

where $\forall j (1 \leq j \leq r_i), X_j \in C_i$ or $X_j \in P_i$

The multiple-ontology system makes extractions for each document with respect to the ontology assigned to it by the ontology selector. Based on this we can obtain expressions for precision and recall of the multiple-ontology system.

Using $\alpha = os(D_j), 1 \leq j \leq K$,

$$Precision(I_m(S)) =$$

$$\frac{\sum_{j=1}^K |\bigcup_{i=1}^{r_\alpha} R(E_{m_\alpha}(O_\alpha, X_i), D_j) \cap \{k(I_\alpha, D_j) \cup k(V_\alpha, D_j)\}|}{\sum_{j=1}^K |\bigcup_{i=1}^{r_\alpha} R(E_{m_\alpha}(O_\alpha, X_i), D_j)|}$$

$$Recall(I_m(S)) =$$

$$\frac{\sum_{j=1}^K |\bigcup_{i=1}^{r_\alpha} R(E_{m_\alpha}(O_\alpha, X_i), D_j) \cap \{k(I_\alpha, D_j) \cup k(V_\alpha, D_j)\}|}{\sum_{j=1}^K |\{k(I_\alpha, D_j) \cup k(V_\alpha, D_j)\}|}$$

The precision and recall for the single-ontology system can be obtained using the formulae given in Section 3.1. When calculating the recall, these formulae will only consider the instances and property values found with respect to the common ontology. However, more instances and property values will exist with respect to the specialized ontologies used by the multiple-ontology system. It is possible to compute a separate measure of recall with respect to these. We call this measure *global recall* (and refer to the recall computed with respect to the common ontology, which is the standard measure of recall, as *local recall*).

$$Global\ Recall(I_s(O_c)) =$$

$$\frac{|\bigcup_{i=1}^{n_s} R(E(O_c, X_i), D) \cap \{k(I_c, D) \cup k(V_c, D)\}|}{\sum_{j=1}^K |\{k(I_\alpha, D_j) \cup k(V_\alpha, D_j)\}|}$$

Here it is assumed that,

$$k(I_c, D) \subset \bigcup_{j=1}^K k(I_\alpha, D_j) \text{ and } k(V_c, D) \subset \bigcup_{j=1}^K k(V_\alpha, D_j).$$

In other words, the common ontology would only contain classes and properties common to all the specialized ontologies. This is what one would normally expect from a common ontology.

3.3 Multiple Ontologies Providing Different Perspectives

In this case we have a set of m ontologies,

$$S = (O_1, O_2, \dots, O_m)$$

which have the same definitions as in Section 3.2. Since none of these ontologies can be seen as a “common ontology”, there will be a set of single-ontology systems

$$I_s(O_i), 1 \leq i \leq m.$$

The multiple-ontology system can be denoted by $I_m(S)$ as in Section 3.2.

$$I_m(S) = \{I_{m_1}(O_1), I_{m_2}(O_2), \dots, I_{m_m}(O_m)\}$$

where each $I_{m_i}(O_i)$ contains a set of r_i extractors for a single ontology O_i as shown by,

$$\forall i (1 \leq i \leq m), I_{m_i}(O_i) = \{E_{m_i}(O_i, X_1), \dots, E_{m_i}(O_i, X_{r_i})\}$$

where $\forall j (1 \leq j \leq r_i), X_j \in C_i$ or $X_j \in P_i$

However, there will be no ontology selector component in this case because the ontologies are not specialized towards sub-domains. Therefore, for a given corpus D , each document of the corpus will be processed with respect to each and every ontology.

Assuming that single-ontology IE systems have already been developed for individual ontologies, the multiple-ontology system can make use of these extractors instead of developing new ones. In developing an extractor to identify individuals of a given class or property values of a given property, the multiple-ontology system can use the extractors of more than one single-ontology system. The intuition behind this approach is implementing a better information extractor by combining a set of different information extractors. For example, for the marriage ontologies we have discussed earlier, the extractor for the “Spouse” class in the multiple-ontology system can use not only the results for the “Spouse” class but also the results of the “Husband” and “Wife” classes of a different ontology. Here, the general idea is to use the information extractors for all the concepts that have some *mapping* with the concept in concern.

Definition Mapping: A mapping $M(X_a, X_b)$ exists between two concepts X_a and X_b of two different ontologies O_a and O_b ($X_a \in C_a \cup P_a$ and $X_b \in C_b \cup P_b$ with usual definitions for O_a and O_b), if and only if, $val(X_a) \equiv val(X_b)$ or $val(X_a) \subset val(X_b)$ or $val(X_a) \supset val(X_b)$, where $val(X_a)$ and $val(X_b)$ represent the sets of individuals/property values of X_a and X_b respectively.

For example, if X_a and X_b are classes, all the individuals of X_a may also be individuals of X_b which means that $val(X_a) \subset val(X_b)$.

Let $X_j \in \{C_i \cup P_i\}$, $1 \leq i \leq m$ and $1 \leq j \leq r_i$ be a class or a property of ontology O_i . (m is the number of ontologies and r_i is the number of extractors for ontology O_i)

Let $\overline{X} = \{\overline{X}_1, \overline{X}_2, \dots, \overline{X}_n\}$ be the set of n properties or classes of other ontologies, which have a mapping $M(X_j, \overline{X}_l)$, $1 \leq l \leq n$ and let $o(\overline{X}_l) \in \{1, 2, \dots, m\}$, $1 \leq l \leq n$ denote the number of the ontology for \overline{X}_l .

The extractor for X_j in the multiple-ontology system can make use of not only the extractor for X_j but also of the extractors for elements of \overline{X} in the single-ontology systems.

This means that the extractions made by the extractor $E_{m_i}(O_i, X_j)$ for the corpus D depends on a set of single-ontology extractors as follows.

$$R(E_{m_i}(O_i, X_j), D) = f_j(R(E_s(O_i, X_j), D), R(E_s(O_{o(\overline{X}_1)}, \overline{X}_1), D), R(E_s(O_{o(\overline{X}_2)}, \overline{X}_2), D), \dots, R(E_s(O_{o(\overline{X}_n)}, \overline{X}_n), D))$$

Here function f_j is based on set operators. It presents the operation of the multiple-ontology system for the given class or a property. For the “Spouse” class of the marriage ontology mentioned earlier, f_j may be the union the results for the “Spouse”, “Husband” and “Wife” classes.

For the multiple-ontology system, precision and recall can be computed as follows.

$$Precision(I_m(S)) = \frac{\sum_{i=1}^m |\bigcup_{j=1}^{r_i} R(E_{m_i}(O_i, X_j), D) \cap \{k(I_i, D) \cup k(V_i, D)\}|}{\sum_{i=1}^m |\bigcup_{j=1}^{r_i} R(E_{m_i}(O_i, X_j), D)|}$$

$$Recall(I_m(S)) = \frac{\sum_{i=1}^m |\bigcup_{j=1}^{r_i} R(E_{m_i}(O_i, X_j), D) \cap \{k(I_i, D) \cup k(V_i, D)\}|}{\sum_{i=1}^m |\{k(I_i, D) \cup k(V_i, D)\}|}$$

For the single ontology systems, precision and recall can be defined using the formulae of Section 3.1. We can also define a formula for global recall as follows.

For the single-ontology IE system $I_s(O_i)$, $1 \leq i \leq m$, $Global\ Recall(I_s(O_i)) =$

$$\frac{|\bigcup_{j=1}^{r_i} R(E_s(O_i, X_j), D) \cap \{k(I_i, D) \cup k(V_i, D)\}|}{\sum_{i=1}^m |\{k(I_i, D) \cup k(V_i, D)\}|}$$

Here the number of extractors in the single-ontology system $I_s(O_i)$ is r_i because both the multiple-ontology system and this system has the same number of extractors for ontology O_i (corresponding to the total number of classes and properties for which extractors are developed).

4. EXPERIMENTS

4.1 Multiple Ontologies Specializing on Subdomains

4.1.1 Corpus and Ontologies

As mentioned earlier, the domain of universities was used in this case study. The corpus consisted of web pages of 100 universities, 50 from North America and 50 from other parts of the world. From each group, 30 were selected for the training set and 20 were used as the test set. Since the set of all documents of a university website is typically very large and contains many pages irrelevant to the task of extracting information about the university (such as personal websites), only a selected set of webpages was included in the corpus. A programming interface to the Google search engine was used for this purpose. This program takes the domain name of a university as the input and selects a set of webpages from that domain by searching for certain key words.

The ontologies used were developed by studying the documents of the training set and other university ontologies. An ontology developed by the Simple HTML Ontology Extensions (SHOE) project³ was helpful in developing the North American ontology. The development of the non-North American ontology was primarily based on documents of the training set. A common ontology (to be used by the single-ontology IE system) was designed by identifying the concepts common to the two specialized ontologies.

The ontologies were defined in OWL using Protégé [3] ontology editor. Figure 1 shows a section of class hierarchy of the common ontology. Figures 2 and 3 show the section of the class hierarchy related to employees of a university in the ontologies for North American and non-North American universities respectively.

4.1.2 Design and Implementation

In this case study, linguistic extraction rules were used as the information extraction technique. As described in Section 2, this technique is based on the use of regular expressions that capture certain types of information. Specific words, phrases and linguistic features such as Part-Of-Speech (POS) tags can be used by these regular expressions, which are often known as rules. We used the General Architecture for Text Engineering (GATE) [1], which is Java-based *shallow* natural language processing tool to implement these rules. When using this tool, extraction rules can be specified using a format known as JAPE (Java Annotations Patterns Engine).

³<http://www.cs.umd.edu/projects/plus/SHOE/onts/univ1.0.html>

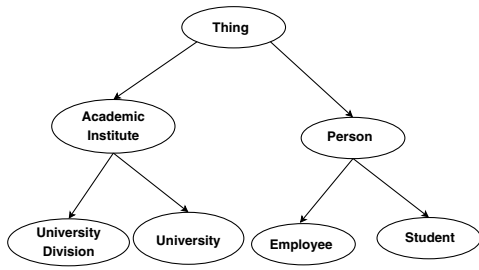


Figure 1: A section of the class hierarchy of the ontology for the common university ontology

As mentioned in Section 3.2, the multiple-ontology system should have an *ontology selector* component which assigns a specialized ontology for each document. We designed this component to make use of the URLs of university web-sites. The documents from domains .edu, .ca and .us were assigned to the North American ontology while the others were assigned to the non-North American ontology. The single-ontology systems uses the common ontology only and does not need an ontology selector.

Apart from the difference on the use of an ontology selector, the architectures of the single and multiple ontology IE systems are the same. The webpages of the corpus are processed by GATE using the JAPE rules that specify the linguistic extraction rules. It writes the output of each document into a separate file. These files are then processed by another program called *ontology language handler* that adds instances or property values to the ontology in concern based on the extractions specified in files. This program uses a popular Java OWL API⁴. The OWL files produced by the ontology language handler constitute the final output of the system. These are then compared against a gold standard for the same document specified by a human to compute the performance measures for the IE system.

In order to get some results within a limited time frame, we decided to restrict the implementation to a set of classes and properties instead of attempting to make extractions for all the concepts of the ontologies. Some classes and properties selected for information extraction at this stage are shown below. For each class or property, the ontologies in which it is found, either directly or by a concept directly mapped into it, are shown within paranthesis (using the symbols NA, NNA and C to denote North American, non-North American and Common ontologies respectively). Note that some concepts are only found in specialized ontologies.

- Classes *University* (NA,NNA,C) and *UniversitySystem* (NA)
- Object properties *hasFunctionalHead* (NA,NNA,C) and *hasCeremonialHead* (NA,NNA,C)
- Datatype properties *isFoundedOn* (NA,NNA,C) and *isReligiousUniversity* (NA,NNA,C)

The regular expressions used for information extraction were manually written by studying the documents of the training set. In some cases, different regular expressions

⁴<http://owlapi.sourceforge.net/index.html>

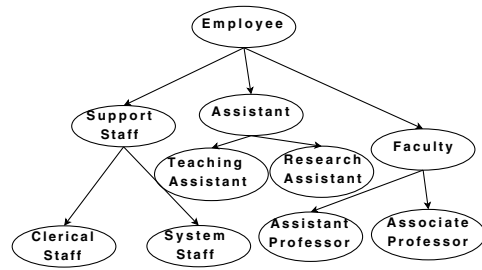


Figure 2: A section of the class hierarchy related to employees in North American ontology

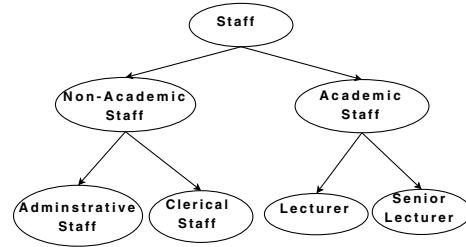


Figure 3: A section of the class hierarchy related to employees in non-North American ontology

were used for concepts of different ontologies that were directly mapped to each other. For example, for the *isReligiousUniversity* datatype property, patterns based on the words “Christian” and “Catholic” were used for North American universities while patterns based on word “Islamic” were also used for non-North American universities.

4.1.3 Results

Table 1 shows the summary of the results obtained. It shows the precision, recall and F1 measure (weighted harmonic mean between precision and recall, giving equal weights for precision and recall) for each sub-domain as well as for the entire domain. Note that the figures for the entire domain are not the averages of the corresponding figures for the two sub-domains because the number of extractions made for the two sub-domains are different. It can be seen that the multiple-ontology system has shown improvements in all three measures. The improvement in recall is somewhat higher than the improvement in precision. Altogether, the multiple ontology system has shown an improvement of about 5% in F1 measure for the entire corpus.

We have also computed the global recall of each system according to the definitions presented in Section 3.2. It can be seen that the global recall is slightly lower than the standard recall (local recall) for North American universities in the single-ontology system. This is because some concepts specific to the North American university ontology (such as the class for university systems) were used by the multiple-ontology system. No such concepts were used for non-North American ontology and as such the figure for global recall is the same as local recall for these universities in the single-ontology system.

