Ontology-Based Information Extraction

Daya C. Wimalasuriya

Towards Partial Completion of the Comprehensive Area Exam

Department of Computer and Information Science

University of Oregon

Committee:

Dr. Dejing Dou, Chair

Dr. Arthur Farley

Dr. Michal Young

Winter 2009

Abstract

Information Extraction (IE) aims to retrieve certain types of information from natural language text by processing them automatically. Ontology-Based Information Extraction (OBIE) has recently emerged as a subfield of Information Extraction. Here, ontologies - which provide formal and explicit specifications of conceptualizations - play a crucial role in the information extraction process. Because of the use of ontologies, this field is related to Knowledge Representation and has the potential to assist the development of the Semantic Web. This paper presents a survey of the current research work in this field including a classification of OBIE systems along different dimensions and an attempt to identify a common architecture among these systems. It also presents a definition for an OBIE system by taking several factors into consideration. In addition, this paper presents the details of some implementation work carried out by the author to explore the use of ontology-based information extraction. These include a project aimed at extracting information from a set of emails and a project aimed at using multiple ontologies to extract information from a set of university websites. The latter appears to be the first OBIE system to make use of multiple ontologies. Finally, the paper discusses possible directions for future research work on this field.

Contents

1 Introduction

1.1 Information Extraction

The general idea behind Information Extraction (IE) is automatically retrieving certain types of information from natural language text. According to Russell and Norvig [48], it aims to process natural language text and to retrieve occurrences of a particular class of objects or events and occurrences of relationships among them. Presenting a similar view, Riloff states that Information Extraction is a form of natural language processing in which certain types of information must be recognized and extracted from text [45].

A system that processes a set of web pages and extracts information regarding countries and their political, economic and social indicators can be given as an example for an information extraction system. Some kind of model that specifies what to look for (e.g., country name, population, capital, main cities, etc.) is needed to guide this process. The system will attempt to retrieve information matching this model and ignore other types of data.

Russell and Norvig further state that Information Extraction lies mid-way between Information Retrieval (IR) systems, which merely find documents that are related to the user's requirements, and text understanding systems (sometimes referred to as text parsers) that attempt to analyze text and extract their semantic contents [48]. Studies on information retrieval have produced many productive systems such as web-based search engines while text understanding systems have not been that successful. Since the difficulty associated with information extraction systems lies in between these two categories, their success has also been somewhere in between the levels achieved by information retrieval systems and text understanding systems.

1.2 Ontology-Based Information Extraction

Ontology-Based Information Extraction (OBIE) has recently emerged as a subfield of Information Extraction. Here, ontologies are used by the information extraction process and the output is generally presented through an ontology. It should be noted that an ontology is defined as *a formal and explicit specification of a shared conceptualization* [51, 23]. Generally, an ontology is specified for a particular domain. Since information extraction is essentially concerned with the task of retrieving information for a particular domain, formally and explicitly specifying the concepts of this domain through an ontology can be helpful to this process. For example, a geopolitical ontology that defines concepts like country, province and city can be used to guide the information extraction system described earlier. This is the general idea behind ontology-based information extraction.

It appears that the term "Ontology-Based Information Extraction" has been conceived only a few years ago. But there has been some work related to this field before that (e.g., work by Hwang [28] on constructing ontologies from text, published in 1999). Recently, there have been many publications that describe OBIE systems and even a workshop has been organized on this topic [5]. Several of these systems are related to ongoing projects. This, together with the fact that the interest on information extraction in general is on the rise, indicate that this field could experience a significant growth in the near future.

1.3 Background Areas

Because of the use of ontologies in Information Extraction, OBIE is related to Natural Language Processing (NLP) as well as Knowledge Representation (KR). This is because Information Extraction is considered a subfield of Natural Language Processing while the concept of ontologies has

originated from studies on Knowledge Representation. In addition, OBIE is also related to Text Mining because of the close relationship between NLP and Text Mining.

Natural Language Processing deals with the problem of automatically understanding and generating utterances of natural human languages. It is generally agreed that there are two branches in the study of NLP; the traditional Artificial Intelligence Natural Language Processing (AI-NLP), also known as Natural Language Understanding (NLU), which relies heavily on logic, grammar and knowledge representation and Statistical Natural Language Processing, which is based on statistical models of language. Comprehensive textbooks have been prepared for each branch. The textbook written by Charniak[11] discusses AI-NLP while the textbook prepared by Manning and Schütze [38] is dedicated to Statistical NLP.

Most of the recent developments in NLP have been in the field of Statistical NLP. Although the use of statistical techniques to interpret natural language may not be very intellectually appealing, studies have shown that this approach can produce impressive results in many hard problems of NLP. Manning and Schütze [38] describe how statistical NLP can be used to tackle problems such as identifying *collocations* (an expression consisting of two or more words) and *word sense disambiguation* (identifying the meaning of a given occurrence of a word for a word that has several meanings or senses). The recent developments in information extraction have also been driven mainly by Statistical NLP. Such IE systems make heavy use of *shallow* NLP techniques such as sentence splitting and Part-Of-Speech (POS) tagging (categorizing words as nouns, verbs, determiners, adjectives, etc.) In contrast, Natural Language Understanding (NLU) systems, which follow the AI-NLP paradigm, use deeper NLP techniques such as construction of parse trees for sentences. These techniques are used to develop text understanding systems (text parsers).

Knowledge Representation (KR) studies how intelligent agents store and process knowledge. It is considered a subfield of Artificial Intelligence that is also related to cognitive science. Ontologies are widely used by KR systems. While ontologies have some similarities with modeling languages such as UML (Unified Modeling Language) they differ from modeling languages in their support of logical reasoning. Ontologies have this capability because they are based on logic, specifically description logic, which provides a formal language for constructing and combining category definitions and efficient algorithms for deciding subset and superset relationships between categories [48].

Ontologies are also related to the emerging Semantic Web. As described by Berners-Lee et al. [9], the goal of the Semantic Web is to bring *meaning* to the Web, creating an environment where software agents roaming from page to page can readily carry out sophisticated tasks for users. Since ontologies represent *knowledge* or *meaning* they are often seen as providing the backbone for the Semantic Web. As such the software agents of the Semantic Web are expected to be able to handle ontologies.

Text mining attempts to discover previously unknown knowledge from text collections [50]. It can be seen as investigating the use of data mining techniques on natural language sources. Text mining is different from NLP because its focus is on discovering hidden knowledge from text rather than understanding natural languages. However, there is a considerable overlap between the two fields. For instance, information retrieval is often considered a text mining task [25]. It has also been stated that text mining combines methodologies from various fields including computational linguistics and information extraction [50]. These factors imply that text mining is a related area for OBIE.

1.4 Potential of Ontology-Based Information Extraction

Although Ontology-Based Information Extraction is a relatively new field of study, it is generally agreed that it has a lot of potential [13, 29, 39, 53]. The following points highlight this potential.

1. *Automatically processing the information contained in natural language text:* A large fraction of the information contained in the World Wide Web takes the form of natural language text. Further, according to some estimates around 80% of the data of a typical corporation are in natural language [46]. Ontology-Based Information Extraction systems as well as general Information Extraction systems are necessary to process these information automatically. This is essential because manually processing such data is becoming increasingly difficult due to their increasing volumes.

2. *Creating semantic contents for the Semantic Web:* Although the success of the Semantic Web relies heavily on the existence of semantic contents that can be processed by software agents, the creation of such contents has been quite slow. Ontology-Based Information Extraction provides a mechanism to generate such contents by converting the information contained in existing web pages into ontologies. This has been pointed out by several authors including Wu and Weld [53] and Cimiano et al. [13]. This process is also known as *the semantic annotation* of web pages.

3. *Improving the quality of ontologies:* As pointed out by Kietz et al. [29] and Maynard et al. [39] among others, Ontology-Based Information Extraction can also be used to evaluate the quality of an ontology. If a given domain ontology can be successfully used by an OBIE system to extract the semantic contents from a set of documents related to that domain, it can be deduced that the ontology is a good representation of the domain. Further, the weaknesses of the ontology can be identified by analyzing the types of semantic content it has failed to extract.

The rest of the paper is organized as follows. Section 2 presents a definition for Ontology-Based Information Extraction. Section 3 presents a review of current research work on OBIE. This consist of identifying a common architecture among OBIE systems (section 3.1), classifying OBIE systems along several dimensions (section 3.2), and reviewing implementation details and performance metrics (section 3.3). Section 4 presents the details of the implementations carried out by the author on OBIE. These consist of a project to extract information from university websites using multiple ontologies (section 4.1) and a project to extract information from a set of emails written by mentally challenged children (section 4.2). Section 5 discusses possible future directions for research work in this area and section 6 provides concluding remarks.

2 Defining Ontology-Based Information Extraction

Although the general idea behind Ontology-Based Information Extraction is quite clear, it appears that researchers have not completely agreed on a definition for the field. This section attempts to arrive at such a definition by identifying the key characteristics of OBIE systems, concentrating on the factors that make OBIE systems different from general IE systems. These are presented below.

- *Process unstructured or semi-structured natural language text:* Since OBIE is a subfield of Information Extraction, which is generally seen as a subfield Natural Language Processing, it is reasonable to limit the inputs to natural language text. They can be either unstructured (e.g., text files) or semi-structured (e.g., web pages using a particular template such as pages from Wikipedia[1]).

[1] `http://www.wikipedia.org`

- *Present the output using ontologies:* Li and Bontcheva. [32] identify the use of a formal ontology as one of the system inputs and as the target output as an important characteristic that distinguishes OBIE systems from IE systems. While this statement holds true for most OBIE systems, there are some OBIE systems that construct the ontology to be used through the information extraction process itself instead of treating it as an input (e.g., The Kylin system [53]). Since constructing an ontology in this manner should not disqualify a system from being an OBIE system, it is prudent to remove the requirement to have an ontology as an input for the system. However, the requirement to represent the output using ontologies appears to be reasonable.

- *Use an information extraction process guided by an ontology:* "Guide" appears to be a suitable word to describe the interaction between the ontology and the information extraction process in an OBIE system; in all OBIE systems, the information extraction process is *guided* by the ontology to extract things such as classes, properties and instances. This means that no new information extraction method is invented but an existing method is oriented to identify the components of an ontology.

  The use of the term "Ontology-Driven Information Extraction" is also relevant here. It has been used in several publications (e.g., [41, 52, 55]). Generally speaking, this can be seen as a synonym for "Ontology-Based Information Extraction", which has emerged due to the lack of a standard terminology. The term "Ontology-Based Information Extraction" is used in this paper since it appears to be the term used by a majority of publications. However, Yildiz and Miksch make a distinction between these two terms [55]. They state that in "ontology-driven" systems the extraction process is *driven* by an ontology whereas the ontology is yet another component in an "ontology-based" system. This argument is based on considering the linguistic rules that are used for information extraction as a part of the ontology. But, there seems to be no common agreement on this in the literature as described in section 3.2.1. Further, other techniques used by OBIE systems such as classification can also provide rules that can be included in an ontology in the same manner. When defining ontologies using the Web Ontology Language(OWL) [2], which has become the de facto standard for defining ontologies, these rules can be included as *annotation properties*, which are not used for reasoning, in the same manner linguistic rules are included in OWL ontologies by Yildiz and Miksch [55]. Hence, the view of the author is that categorizing systems as ontology-driven and ontology-based is essentially subjective.

Combining these factors with the definitions of Information Extraction presented by Russell and Norvig [48] and Riloff [45] results in the following definition:

Definition *An Ontology-Based Information Extraction System*: A system that processes unstructured or semi-structured natural language text through a mechanism guided by ontologies to extract certain types of information and presents the output using ontologies.

It should be noted that this definition encompasses systems that construct an ontology by processing natural language text (e.g., [28, 29]) in addition to systems that identify information related to an ontology (and present them as instances of the ontology). While ontology construction is generally not associated with information extraction, it can be seen as an important step in the ontology-based information extraction process. Further, ontology construction itself actually extracts some information because it identifies the concepts and relationships of the domain in concern. Hence, it makes sense to categorize these systems under ontology-based information extraction.

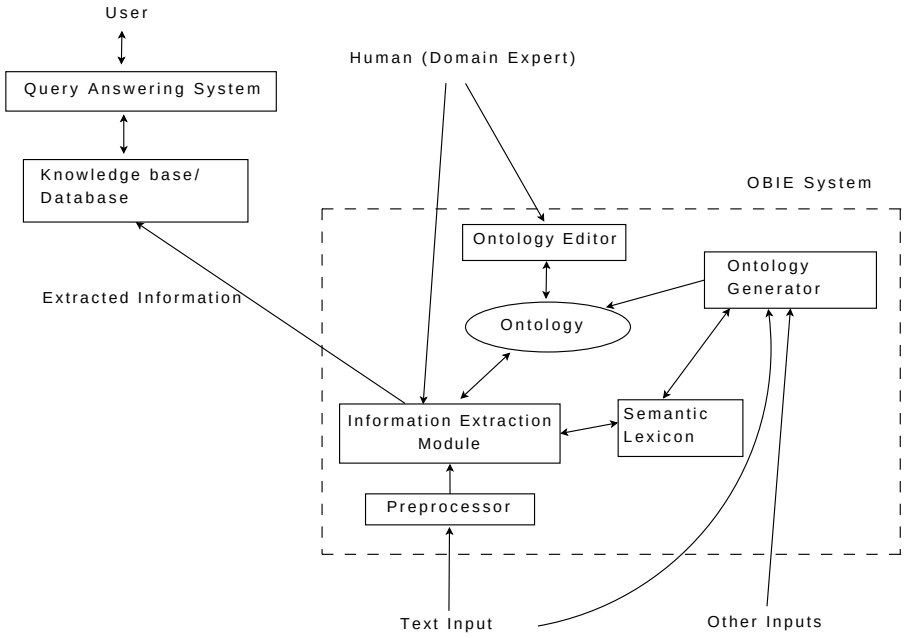

Figure 1: General Architecture of an OBIE system

3 A Survey of Current Research Work on Ontology-Based Information Extraction

This section presents a survey of current research work on Ontology-Based Information Extraction. First, an attempt is made to identify a common architecture for these systems. Then existing OBIE systems are classified along several dimensions with the objective of obtaining a better understanding of their operation. Then the implementation details of these systems are reviewed including the tools widely used by these systems and the performance metrics used to compare their performance.

3.1 Common Architectures and General Functionality

Even though the implementation details of individual OBIE systems are different from each other, a common architecture of such systems can be identified from a higher level. Figure 1 schematically represents this common architecture. This figure represents the union of different components found in different OBIE systems. As such, many OBIE systems do not contain all the components of this architecture. For example, the systems that use an ontology defined by others instead of constructing an ontology internally (as discussed in section 3.2.2) do not have the "Ontology Generator" component. In addition, slight variation from this architecture can be observed in some systems.

It should also be noted that in some implementations, the OBIE system is a part of a larger system that answers user queries based on the information extracted by the OBIE system. Figure 1 shows these components as well. But they should not be recognized as parts of an OBIE system.

As represented by Figure 1, the textual input of an OBIE systems first goes through a preprocessor component, which converts the text to a format that can be handled by the Information Extraction Module. For example, this might involve removing tags from an html file and converting it into a pure text file.

The Information Extraction Module is where actual information extraction takes places. This can be implemented using several techniques as described in section 3.2.1. No matter what information extraction technique is used, it is *guided* by an ontology. A semantic lexicon for the language in concern is often used to support this purpose. For example, the WordNet [21] toolkit is widely used for the English language. It groups English words into sets of synonyms called synsets and provides semantic relationships between them including a taxonomy.

The ontology that is used by the system may be generated internally by an Ontology Generator component. This process too might make use of a semantic lexicon. In addition, humans may assist the system in the ontology generation process. This is typically carried out through an ontology editor such as Protégé [3]. Humans may also be involved in the information extraction process in some systems that operate in a semi-automatic manner.

The output of the OBIE system consists of the information extracted from the text. They can be represented using an ontology definition language such as OWL [2]. In addition, the output might also include links to text documents from which the information was extracted. This is useful in providing a justification for an answer given to a user relying on the extracted information. (Berners-Lee speaks of an "Oh yeah?" button that provides such explanations [8].)

As mentioned earlier, the OBIE system is a part of a larger query-answering system in some implementations. In such systems, the output of the OBIE process is often stored in a database or a knowledge base. An approach such as SOR [34] can be used to store ontologies in databases. The query answering system makes use of the extracted information, stored either in a knowledge base or a database, and answers user queries. This many also include a reasoning component. The nature of the interface provided by the query answering system to the users depends on the particular implementation.

It is insightful to analyze how some OBIE systems fit into this architecture. For example, the OBIE system implemented by Saggion et al. [49] operates as a part of the larger EU MUSING project. The output of the OBIE system is stored in a knowledge base, which is then used by MUSING applications that constitute the query answering system in this case. The ontology to be used by the system is manually defined by domain experts and as such this system does not have an ontology generation component. The information extraction module of this system uses linguistic rules and gazetteer lists (which are described in section 3.2.1). Information extraction operates in a semi-automatic manner, where incorrect extractions made by the process are corrected by humans. Note that the general architecture accommodates this.

For the Kylin OBIE system [53], the input consists of a set of web pages of Wikipedia. These files go through a preprocessor before being used by the information extraction module. In this case, the information extraction module employs classification the IE technique. The ontology is constructed by a special component named "Kylin Ontology Generator". This ontology generator makes use of WordNet, which is a semantic lexicon for the English language as mentioned earlier. Kylin is not a part of a larger system and as such its output is not stored in a data store. Instead the accuracy of the extractions is evaluated using precision and recall as described in section 3.3.

A similar analysis can be carried out for other OBIE systems as well.

3.2 Classification of Current OBIE Systems

This section provides a classification of current OBIE systems along different dimensions.

3.2.1 Information Extraction Method

Over the years, several types of Information Extraction techniques have been developed. Moens has presented a comprehensive categorization and analysis of these techniques in the form of a

textbook [42]. Most of the these techniques have been adopted by OBIE systems. In OBIE systems, they are guided by ontologies to extract information related to ontologies such as instances and property values.

The following are the main Information Extraction methods employed by the OBIE systems studied in this work.

Linguistic Rules Represented by Regular Expressions: The general idea behind this technique is specifying regular expressions that capture certain types of information. 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. By specifying a set of rules like this, it is possible to extract a significant amount of information. The set of regular expressions are often implemented using finite-state transducers which consist of a series of finite-state automata. In practice, they are combined with NLP tools such as Part-Of-Speech (POS) taggers and noun phrase chunkers enabling the use of a wide variety of rules.

Despite its simplicity, experiments have shown that this technique produces surprisingly good results. The FASTUS Information Extraction system [7], implemented in 1993, appears to be one of the earliest systems to use this method. The General Architecture for Text Engineering (GATE) [1], which is a widely used NLP framework, provides an easy to use platform to employ this technique.

Embley's work back in 2004 [20] appears to be one of the first OBIE systems to use this technique. He has combined linguistic rules that use regular expressions with the elements of ontologies such as classes and properties, resulting in "extraction ontologies". Following Embley, Yildiz and Miksch have employed a similar technique in their ontoX system [55]. Such regular expressions are also used by the Textpresso system for biological literature [43], which is more of an information retrieval system but can be seen as an OBIE system as well. The same principle is employed to construct an ontology in the implementation by Hwang [28]. In addition, the OBIE systems that use the GATE architecture, such as the implementation by Saggion et al. [49] rely at least partly on this method.

Embley considers the linguistic rules used for information extractions a part of his *extraction ontologies*. Presenting a similar view Maedche et al. define a *concrete ontology* as the combination of an *abstract ontology* and the *lexicon* for that abstract ontology [36]. The argument raised by Yildiz and Miksch [55] on the difference between ontology-driven and ontology-based information extraction methods is also based on this understanding. However, it appears that there is no general consensus on this issue in the literature. For instance, this approach appears to go against the generally accepted definition of an ontology provided by Gruber, who states that an ontology is a formal specification of the conceptualization [23]. Since linguistic rules are never 100% accurate, including them in the ontology definition appears to violate this condition.

Gazetteers: This technique relies on finite-state automata just like linguistic rules but recognizes individual words or phrases instead of patterns. The words or phrases to be recognized are provided to the system in the form of a list, known as a gazetteer. This technique is widely used in the named-entity recognition task, which can be seen a component of information extraction. It is concerned with identifying individual entities of a particular category. For example, gazetteers can be used to recognize states of the US or the countries of the world.

This technique is used by several OBIE systems. These systems often use gazetteers that list all the instances of some classes of the ontology. They have been used in the SOBA system [10] to get details about soccer games and in the implementation by Saggion et al. [49] to get details about countries and regions.

It is clear that one has to be careful in using gazetteers in an OBIE system or an IE system. For example, if designing an IE system to get information on terrorist organizations includes preparing a

gazetteer of such organizations by reading a large number of news wires, it is clear that something is out of place. To avoid the misuse of this technique, it can be proposed that the following conditions should be satisfied when using it.

1. Specify exactly what is being identified by the gazetteers.

2. Specify where the information for the gazetteers were obtained from: These should be valid public references and should involve little or no processing. For example a list of all departments and agencies of the US government is available from the official web site of the US government[2].

Classification Techniques: Different classification techniques such as Support Vector Machines (SVM), maximum entropy models and decision trees have been used in information extraction. Moens provides a comprehensive review of these techniques and categorizes them as "Supervised Classification" techniques [42].

Different linguistic features such as POS tags, capitalization information and individual words are used as input for classification. It is also a common practice to convert an information extraction task into a set of binary classification tasks. For example, the IE system implemented by Li et al. [33], which uses uneven margins SVM and perceptron techniques, uses one binary classifier to decide whether a word token is a start of an entity and uses another to detect the end token.

When using classification for OBIE, classifiers are trained to identify different components of an ontology such as objects and property values. The Kylin [53] OBIE system employs two classification techniques. It uses the maximum entropy model to predict which attribute values are present in a sentence and the Conditional Random Fields (CRF) model to identify attributes within a sentence. The implementation by Li and Bontcheva [32] uses the Hieron large margin algorithm for hierarchical classification [15] to identify instances of an ontology.

Construction of Partial Parse Trees: A small number of OBIE systems construct a semantically annotated parse tree for the text as a part of the information extraction process. The constructed parse trees are not meant to comprehensively represent the semantic content of the text as aimed by text understanding systems such as TACITUS [27]. Hence, this type of processing can still be categorized under *shallow* NLP, typically used by IE systems, as opposed to *deep* NLP used by text understanding systems, although they conduct more analysis than looking for occurrences of regular expressions.

A representative system for OBIE systems that employ this approach is the implementation by Maedche et al. [36]. This system, which has been developed for the German language, makes use of a NLP toolkit for the German language named Saarbücker Message Extracting System (SMES). The SMES system consists of several components and operates at the lexical level (words) as well at the clause level. It produces an *under-specified dependency structure* as the output, which is basically a partial parse tree. This structure is used for information extraction. The Text-To-Onto system developed by Maedche and Staab [37], which uses the SMES system to construct an ontology, can be considered another OBIE system that adopts this approach.

Analyzing HTML/XML Tags: IE and OBIE systems that use HTML or XML pages as input can extract certain types of information using the tags of these documents. For example, a system that is aware of the HTML tags for tables can extract information from tables present in html pages. The first row of the table denotes attributes and the remaining rows indicate the attribute values

[2] `http://www.usa.gov/Agencies/Federal/All_Agencies/index.shtml`

for individual records or objects. XML documents would provide more opportunities to extract information in this manner because they allow users to define their own tags.

The SOBA OBIE system extracts information from HTML tables into a knowledge base that uses F-Logic [10]. This system uses a corpus of web pages about soccer games as its source.

Web-Based Search: Using queries on web-based search engines for information extraction appears to be a new technique. (It has not been recognized as an IE technique even in the review of IE techniques compiled by Moens in 2006 [42].) The general idea behind this approach is using the Web as a big corpus.

Cimiano et al. have implemented an OBIE system named "Pattern-based Annotation through Knowledge on the Web (PANKOW)" that semantically annotates a given web page using web-based searches only [13]. It conducts web searches for every combination of identified proper nouns in the document with all the concepts of the ontology for a set of linguistic patterns. Such patterns include Hearst patterns like "`<CONCEPT>s such as <INSTANCE>`" [26]. The concept labels for the proper nouns are determined based on the aggregate number of hits recorded for each concept. The C-PANKOW system [14] operates on the same principles but improves performance by taking the context into consideration. The OntoSyphon system uses a similar approach but aims to learn all possible information about some ontological concepts instead of extracting information about a given document [41].

In addition, Wu et al. have used search engine results to improve their Kylin system by adding more training examples for their classifiers [52]. Here, the vast amount of information available from the Web is used to overcome the data sparsity problem.

Finally, it is worth pointing out that some OBIE systems use more than one information extraction technique. For example, it can be seen that the improved Kylin system [52] uses classification as well as web-based search.

3.2.2 Ontology Construction and Update

OBIE systems can be classified based on the manner in which they acquire the ontology to be used for information extraction. One approach is to consider the ontology as an input to the system. Under this approach, the ontology can be constructed manually or an "off-the-shelf" ontology constructed by others can be used. Most OBIE systems appear to adopt this approach. Such systems include SOBA [10], the implementation by Li and Bontcheva [32], the implementation by Saggion et al. [49] and PANKOW [13].

The other approach is to construct an ontology as a part of the OBIE process. This can be achieved by building an ontology from scratch or by using an existing ontology as the base. The OBIE systems that aim to construct an ontology for a given domain obviously use this approach. Such systems include Text-To-Onto [37] and the implementation by Hwang [28]. Kylin (through Kylin Ontology Generator [54]) and the implementation by Maedche et al. [36] construct an ontology as a part of the process although their main aim is to identify new instances for the concepts of the ontology.

In addition, it is possible to update the ontology by adding new classes and properties through the information extraction process. (Identifying objects and their property values are not considered updates to the ontology here.) Such updates can be conducted for both cases mentioned above. However, only few systems update the ontology in this manner. Such systems include the implementations by Maedche et al. [36] and Dung and Kameyama [19].

3.2.3 Components of the Ontology Extracted

An ontology consists of several components such as classes, data type properties, object properties (including taxonomical relationships), objects (instances), property values of the objets and constraints. The OWL specification [2] defines the types of components supported by OWL, which is generally regarded as the standard language for specifying ontologies.

OBIE systems can be classified based on the components of the ontology extracted by them. Ontology construction systems generally extract information related to classes only. Among such systems, the implementation by Hwang [28], extracts class names and the taxonomy (class hierarchy) only. In contrast, Text-To-Onto [37] discovers class names, taxonomical relationships as well as non-taxonomical relationships.

The systems that construct an ontology and find information regarding instances extract many components of an ontology. The Kylin system [52] extracts class names, the taxonomy and data type properties during the ontology construction process. In subsequent phases it extracts instances and their data type property values. The implementation by Maedche et al. [36] also extracts all these components.

Many OBIE systems that concentrate on instances only extract identifiers (names) for instances. Such systems include the implementation by Li and Bontcheva [32], PANKOW [13] and OntoSyphon [41]. Some systems extract property values of the instances as well. Such systems include SOBA [10], the implementation by Embley [20] and the implementation by Saggion et al. [49]. It is difficult to determine whether they extract the values for both data type properties and object properties or for data type properties only.

3.2.4 Types of Sources

Although all OBIE systems extract information from natural language text, the sources used by them can be quite different. Some systems are capable of handling any type of natural language text while others have specific requirements for the document structure or target specific web sites.

Many OBIE systems can handle any type of documents including web pages and word-processed documents but require that they be related to a particular domain. Such systems include the implementation by Maedche et al. [36], the implementation by Embley [20] and the implementation by Saggion et al. [49].

In contrast, SOBA retrieves the web pages that it processes using its own web crawler. It can only handle html pages as it makes use of html tags in the information extraction process. The Kylin system has been designed specifically for Wikipedia. It makes use of structures specific to Wikipedia pages like infoboxes.

Table 1 presents a summary of the classification described in this section.

3.3 Implementation Details and Performance Evaluation

3.3.1 Tools used

One main category of tools used by OBIE systems is *shallow* NLP (Natural Language Processing) tools. The word "shallow" distinguishes these tools from text understanding systems that perform a deeper analysis of natural language. These tools perform functions such as Part-Of-Speech (POS) tagging, sentence splitting and identifying occurrences of regular expressions. They are used by almost all information extraction techniques. For example, linguistic rules represented by regular expressions can be directly implemented using these tools whereas the features that are used for classification can be extracted using them. Widely used such tools include GATE [1], SProUT [4]

Table 1: Summary of the Classification of OBIE systems

System	Information Extraction Method(s)	Ontology Construction and Update[a]	Components of the Ontology Extracted	Types of Sources
Kylin [52]	Classification, Web-based search	Constructed by process; not updated.	Classes, taxonomy, data type properties, objects, property values	Wikipedia pages
PANKOW [13]	Web-based Search	Off-the-shelf; not updated	Objects	No restriction
OntoSyphon [41]	Web-based Search	Off-the-shelf; not updated	Objects	No restriction
Maedche et al. [36]	Partial parse trees	Constructed by process; updated	Classes, taxonomy, data type properties, objects, property values	Documents from a domain
Text-To-Onto [37]	Partial parse trees	Constructed by process; N/A	Classes, taxonomy, other relationships	Documents from a domain
SOBA [10]	Linguistic rules, Gazetteer lists, Analyzing tags	Off-the-shelf; not updated	Objects, property values	html files from a domain
Embley [20]	Linguistic rules	Manually defined; not updated	Objects, property values	Documents from a domain
Saggion et al. [49]	Linguistic rules and Gazetteer lists	Manually defined; not updated	Objects, property values	Documents from a domain
Li and Bontcheva. [32]	Classification	Off-the-shelf; not updated	Objects	Documents from a domain
Hwang [28]	Linguistic rules	Constructed by process; N/A	Classes, taxonomy, properties	Documents from a domain
ontoX [55]	Linguistic rules	Manually defined; not updated	Objects, data type property values	Documents from a domain

[a]Update is not applicable to ontology construction systems since their objective is constructing the ontology.

and the tools developed by Stanford NLP Group[3]. In addition, the Saarbücker Message Extracting System (SMES) used by Maedche and his group [36, 37] can be categorized as a shallow NLP system although it appears to conduct more analysis than other shallow NLP systems, as mentioned earlier.

Semantic lexicons, also known as lexical-semantic databases and lexical-semantic nets, also play an important role in many OBIE systems. These tools organize words of a natural language according to meanings and identify relationships between the words. Such relationships related to subsumption (hypernyms and hyponyms) are sometimes seen as giving rise to a *lexical ontology*. The information contained in semantic lexicons are utilized in different manners by OBIE systems. For example, the Kylin Ontology Generator uses them to assist the construction of an ontology [54]. For the English language, WordNet [21] is the most widely used semantic lexicon. Similar tools are available for some other languages such as GermaNet[4] for German and Hindi WordNet[5] for Hindi.

Ontology Editors are also used by OBIE systems. These tools can be used to manually construct an ontology which is later used by an OBIE system. They can also be used to correct the output of the information extraction process in systems that operate in a semi-automatic manner. Protégé [3] and OntoEdit[6] are two widely used ontology editors. In addition, GATE toolkit includes its own ontology editor.

Manually annotating a natural language text with ontological concepts is also useful in developing OBIE systems. Such annotations are often used as a gold standard in evaluating the accuracy of an OBIE system. GATE provides a tool for this.

3.3.2 Text Corpora

It is generally accepted that Message Understanding Conferences (MUC) and their successor - Automatic Content Extraction (ACE) Program - fueled the development in information extraction by providing standard text corpora and standard extraction tasks. This had allowed the researchers to objectively evaluate different IE systems and identify strengths and weaknesses of individual systems. As such, it can be expected that having standard text corpora and well defined tasks will have a similar positive impact on the development of ontology-based information extraction.

However, since no such conferences or standard text corpora currently exist for OBIE, most researchers have compiled their own corpora for OBIE systems. For example, Saggion et al. [49] have collected a set of around 100 company web sites and a set of company reports and newspaper articles for a test case on company intelligence; Li and Bontcheva [32] have created a corpus covering the topics of business, international politics and UK politics for their OBIE system; Cimiano et al. have selected 30 files from a popular travel web site to create a corpus for the PANKOW system [13]. In many cases, the researchers have also manually annotated the selected corpus with ontological information in order to create a gold standard to evaluate the accuracy. It should be noted that this is a time consuming and costly exercise. It is clear that having semantically annotated standard corpora, similar to the corpora provided by MUC/ACE conferences, would relieve the researchers of this difficulty.

There have been some attempts to create standard text corpora that can be used to evaluate OBIE systems. Peters et al. have created one such corpus named "OntoNews" [40]. They have collected 292 news articles from three news agencies and annotated them with the concepts of the PROTON ontology[7]. In this annotation process, they have identified occurrences of the classes of the

[3] `http://nlp.stanford.edu/software/index.shtml`
[4] `http://www.sfs.uni-tuebingen.de/GermaNet/`
[5] `http://www.cfilt.iitb.ac.in/wordnet/webhwn/`
[6] `http://www.ontoknowledge.org/tools/ontoedit.shtml`
[7] `http://proton.semanticweb.org/`

PROTON ontology. Since PROTON ontology is quite deep (up to eight levels), these annotations are quite complex and therefore seen as quite difficult for an OBIE system to recognize [40]. Hence, this corpus can be expected to evaluate the effectiveness of different OBIE systems well.

3.3.3 Performance Measures

In Information Extraction (as well as in Information Retrieval), *precision* and *recall* are the two most used metrics for performance measurement. Precision shows the number of correctly identified items as a percentage of the total number of items identified. Recall shows the number of correctly identified items as a percentage of the total number of correct items available. They are defined by the following formulae.

$$Precision = \frac{\mid correct\ answers \mid}{\mid total\ answers \mid}$$

$$Recall = \frac{\mid correct\ answers \mid}{\mid correct\ answers\ in\ the\ gold\ standard \mid}$$

Most IE systems face a tradeoff between improving precision and recall. Recall can be increased by making a lot of extractions but that is likely to reduce precision. Similarly, precision can be increased by making only few extractions that are clearly correct but that would reduce recall.

The F-measure is often used together with precision and recall. It is a weighted average of the two metrics defined by the following equation.

$$F - measure = \frac{(\beta^2 + 1)Precision * Recall}{(\beta^2 * Recall) + Precision}$$

Here, β denotes the weighting of precision vs. recall. In most situations, 0.5 is used for β, giving equal weights for precision and recall.

As pointed out by Maynard et al. [40], using precision and recall with OBIE systems can be problematic because these metrics are binary in nature. They expect each answer to be categorized as correct or incorrect. Maynard et al. state that OBIE systems should be evaluated in a scalar manner, allowing different degrees of correctness [40]. Such metrics are useful in evaluating the accuracy in identifying instances of classes from text; the score can be based on the closeness of the assigned class to the correct class in the taxonomy of the ontology. However, it should be noted that binary measures *can* be used for the task of identifying property values of instances; each property value (such as the number of employees or the CEO of a company) would be either *correct* or *incorrect*.

For the task of identifying instances of an ontology (often known as ontology population), Cimiano et al. have used a performance measure called "Learning Accuracy (LA)" [14]. They have adopted this metric from a work by Hahn et al. [24]. This measures the closeness of the assigned class label to the correct class label based on the subsumption hierarchy of the ontology. It gives a number between 0 and 1 where 1 indicates that all assignments are correct. Learning accuracy is defined as follows:

For each candidate pair (i, c) of the output, where i is an instance and c is the assigned class label, there is a pair $(i, gold(i))$ in the gold standard and $c, gold(i) \in O$, where O is the set of classes in the ontology used.
The least common superconcept (lcs) between two classes a and b is defined by:

$$lcs(a, b) = \arg\min_{c \in O}(\delta(a, c) + \delta(b, c) + \delta(top, c))$$

where $\delta(a, b)$ is the number of edges on the shortest path between a and b and *top* is the root class. Now, the taxonomy similarity T_{sim} between two classes x and y is defined as:

$$T_{sim}(x, y) = \frac{\delta(top, z) + 1}{\delta(top, z) + \delta(x, z) + \delta(y, z) + 1}$$

where $z = lcs(x, y)$. Then learning accuracy for a set of instance - class label pairs, X is defined as follows.

$$LA(X) = \frac{1}{\mid X \mid} \sum_{(i,c) \in X} T_{sim}(c, gold(i))$$

Maynard et al. have defined two metrics called Augmented Precision (AP) and Augmented Recall (AR) that can be used for OBIE systems [40]. These combine the concepts behind precision and recall with cost-based metrics. Experiments have shown that these measures are at least as effective as Learning Accuracy. In particular, Augmented Recall may be a useful metric because Learning Accuracy appears to be more of a measure of precision.

It should be noted that these accuracy measures are used only for the tasks of identifying objects and property values. Evaluating the quality of a constructed ontology (i.e., classes, taxonomy and properties) is quite subjective because it is difficult to come up with a gold standard for this task.

4 Implementations on Ontology-Based Information Extraction

This section presents the details of some implementation work carried out by the author to explore the use of Ontology-Based Information Extraction.

4.1 Extracting Information from University Websites Using Multiple Ontologies

The objective of this implementation was to explore the use of multiple ontologies in OBIE. This section first describes the rationale behind using multiple ontologies in OBIE and then presents the details of the implementation.

4.1.1 Rationale for Using Multiple Ontologies in OBIE

All the OBIE systems mentioned earlier 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. The widely used definition of an ontology, presented by Gruber [23] and later refined by Studer et al. [51], allow the existence of multiple ontologies for the same domain by defining an ontology as a formal and explicit specification of ***a*** shared conceptualization. This leaves open the possibility for the existence of multiple conceptualizations for the same domain and ontologies that are based on them. In fact, such different ontologies have been developed for several domains. In addition, issues related to the existence of multiple ontologies such as integrating them and discovering "mappings" between the concepts of different ontologies has become an active research area as evidenced by the research papers published on these topics [12, 18, 22, 35].

Generally speaking, it can be seen that multiple ontologies developed for the same domain belong to one of the following categories.

1. **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".

2. **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 paying special attention to the concepts unique to it. For instance, the class for universities in the ontology for North American universities would have a property named "hasPresident" whereas the ontology for British universities might have a property named "hasViceChancellor".

The following are the opportunities and challenges involved with using multiple ontologies in OBIE.

1. **Possible improvement in recall:**

   It can be hypothesized that using multiple onotlogies (related to either category described above) in OBIE systems would improve *recall*. As described in section 3.3.3 recall shows the number of correctly identified items as a percentage of the total number of correct items available and is one of the two widely used performance metrics for IE systems (the other being *precision*).

   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 gay 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. Using both ontologies might therefore extract more instances than what either one is capable on its own.

   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. For the ontologies of the university domain described above, an ontology for North American universities can be expected to be more successful in making extractions for US and Canadian universities. Such an ontology might also make extractions related to concepts that can generally be considered unique to North American universities such as mascots and nicknames for sports teams. Ontologies defined for other sub-domains can be expected to be similarly successful in their own domains. Hence, together the set of ontologies can be expected to make more extractions than what is possible under a common ontology.

   It has to be noted that the recall achieved by an OBIE system depends on the success of the actual information extraction process, which is *not* a part of the ontology. But the ontology has a significant impact on the information extraction process employed by an OBIE system. If classification is used as the information extraction technique, the training examples used to train the classifier would be determined by the ontology and thus the ontology would affect the classifier. Similar effects can be observed in other information extraction techniques such as linguistic extraction rules and web-based search. Hence, the fact that the information extraction process of an OBIE system is not a part of the ontology should not invalidate the hypothesis about the effect of multiple ontologies on recall.

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?".

3. **Relationship with mappings:**

   In answering questions such as the ones described in the previous case, it may be necessary to translate instances of one ontology to another. "Mappings" between concepts of different ontologies can be used for this purpose. Mappings are also useful in merging ontologies and in data integration. The discovery of mappings between ontologies and their use is an active research area as shown by the review paper compiled on this topic by Choi et al. [12]. OBIE systems that use multiple ontologies would provide an opportunity to test the effectiveness of the techniques that discover mappings between ontologies. Moreover, it may be possible to develop special techniques to discover mappings between ontologies based on extractions made by OBIE systems using the ontologies.

4.1.2 Introduction to the Project

In order to explore the above mentioned opportunities and challenges in using multiple ontologies in an OBIE system, a system was designed and implemented for the domain of universities. Two ontologies were used by the system, one for North American universities and the other for universities from other parts of the world. While using a set of ontologies related to a set of sub-domains such as British universities, research universities and community colleges would have been better, the assumption was that using two ontologies would facilitate carrying out some experimentation on the use of multiple ontologies in OBIE systems while limiting the implementation work. Linguistic rules expressed as regular expressions were used as the information extraction technique. To test how a system with a single ontology would perform in this domain, an OBIE system that uses a common ontology for universities was also developed. It was thought that using either of the specialized ontologies for this system would not be appropriate.

4.1.3 Design of the System

Corpus: Designers of most IE and OBIE systems create a corpus by compiling a set of documents related to their domain of interest. For example, in developing an OBIE system for the domain of business organizations, a set of new articles related to that topic can be used as the corpus.

However, this approach could not be directly applied to this system since it was expected to use the URL of the university websites as the input. Such URLs display the "welcome page" of the website and most of the useful information lie elsewhere in the website. Hence, the approach adopted was to retrieve the relevant webpages from the websites to create the corpus for a given university. This is essentially an information retrieval task since it deals with the problem of retrieving relevant documents from the set of all documents in a university web site. An Application Programming Interface (API) for a search engine such as Google was expected to be used for this purpose.

An alternative to this approach is to include all the documents of a website in the corpus. A web crawler program could have been used for this purpose. But this approach would download

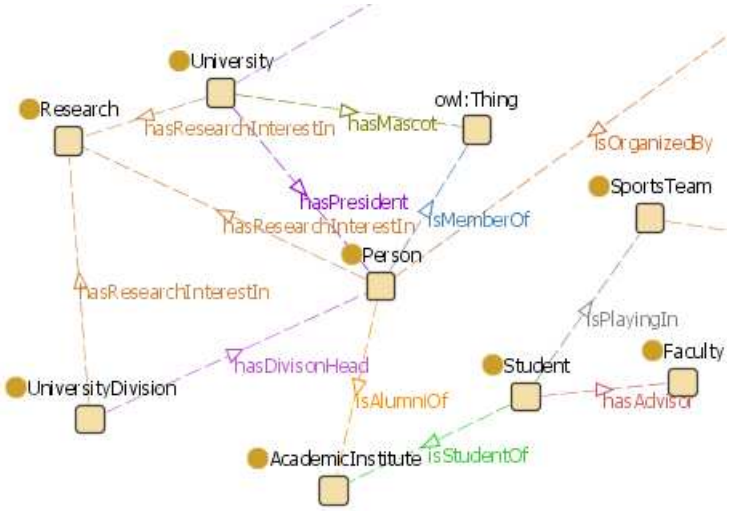

Figure 2: A section of the ontology for North American universities

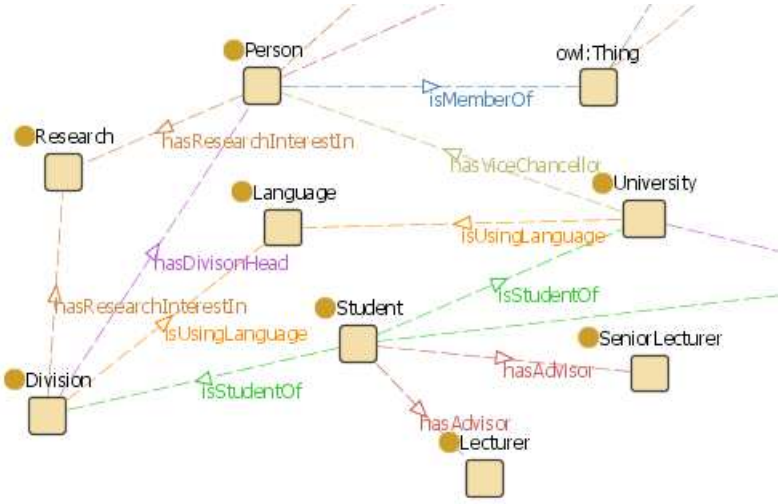

Figure 3: A section of the ontology for universities from other parts of the world

many irrelevant pages as university web sites contain many documents not related to the university ontologies such as course material and personal websites. It was seen that first selecting the relevant pages as a pre-processing step would increase precision and reduce the load on the information extraction process in terms of the number of documents to be processed.

The URLs of 100 universities were used, 50 from North American universities and 50 from universities of other parts of the world. From each group 30 were used as the training set while the remaining 20 were used as the test set. The universities to be included in the training set and test set were randomly selected.

Ontologies: As mentioned earlier, three ontologies were expected to be used in the implementation: two ontologies - one for North American universities and the other for universities from other parts of the world - for the system with multiple ontologies and one common university ontology for the single-ontology system. These ontologies were manually developed by studying the webpages from the training set and by studying other university ontologies available on the Web. An ontology developed as a part of the Simple HTML Ontology Extensions (SHOE) project[8] was used as the guide in developing the ontology for North American universities. An ontology developed by a research group in University of Manchester, UK [9] was helpful in developing the ontology for universities from other parts of the world. The common ontology was designed capture the semantics common to these two ontologies.

In designing ontologies for North American and non-North American universities, an attempt was made to capture the differences between two types of universities whenever possible. For example, an object property named "hasPresident" was defined for the university class in the North American ontology while a property names "hasViceChancellor" was defined in the non-North American ontology. The corresponding property was named "hasFunctionalHead" in the common ontology since it was expected to represent the common characteristics of all universities. The extraction rules used to identify values for these properties in individual instances (representing different universities) were also different depending on the ontology.

In some situations, different names were used for components of American and Non-American ontologies just to highlight the difference between the two ontologies. For example, a class named "Location" was defined in the North American ontology while the corresponding class in the non-North American ontology was named "Place". Such changes do not represent actual differences in the underlying conceptualizations.

The ontologies were developed using the Protégé [3] ontology editor. Figures 2 and 3 show sections of the ontologies for North American universities and universities of other parts of the world respectively.

System Architecture: Figure 4 schematically represents the architecture of the system. This architecture captures the functionality of the single-ontology system as well as the system with multiple ontologies.

As represented by figure 4, the Corpus Builder component takes the university URLs as input and produces the webpages relevant to the information extraction task as output. As mentioned earlier, it achieves this by using a programming interface to a search engine. It was decided to use Google as the search engine for this purpose. Using Google, it is possible to restrict searches to a particular domain and this feature allows the searches to be conducted only in the web domain of a university.

[8] http://www.cs.umd.edu/projects/plus/SHOE//onts/univ1.0.html

[9] http://owl.cs.manchester.ac.uk/repository/download?ontology=http://www.mindswap.org/ontologies/debugging/university.owl&version=0&format=RDF/XML

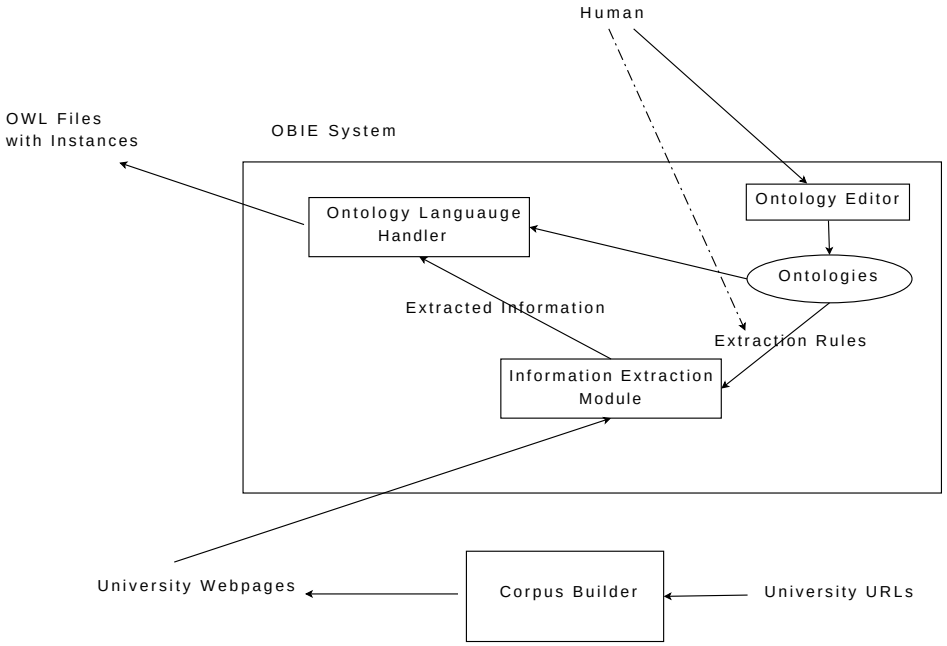

Figure 4: Architecture of the Systems

The webpages produced by the corpus builder constitute the input of the OBIE system. They are processed by the information extraction module, which is implemented using the General Architecture for Text Engineering (GATE) [1]. GATE is widely used for *shallow* natural language processing tasks such as Part-Of-Speech (POS) tagging and sentence splitting. Here GATE was used to recognize linguistic regular expressions. These regular expressions, often known as rules, identify places where instances of classes or values of properties are specified in text. (An example rule is presented in section 4.1.4). These rules are specified to GATE in files that contain grammars written in a format known as JAPE (Java Annotation Patterns Engine).

The information extraction module was designed to be an executable program, which uses GATE as a library. Java was selected as the implementation language partly because a Java library is available for GATE. The IE Module makes extractions from the webpages of universities and writes them into files which are later processed by the Ontology Language Handler. It analyzes these extractions and uses them to create instances and property values in the respective ontology. This component was implemented using a popular Java OWL API[10].

It should be noted that GATE has its own ontology plug-in, which allows annotating extractions with class names and adding instances to an ontology. However, it was seen that these functionalities are normally used to identify instances of classes rather than identifying property values. Further, it was decided that clearly separating the information extraction task and the task of adding instances and property values to ontology would make the design cleaner. Therefore, GATE libraries were not used for the task of adding instances and property values to the ontology, which was delegated to the Ontology Language Handler.

As mentioned earlier, the ontology was manually defined. The extraction rules were specified in the form of JAPE grammars for each component in the ontology for which extractions are made. These rules were written manually by studying the documents of the training set. The effect of the

[10] `http://owlapi.sourceforge.net/index.html`

ontology in writing these rules represent the impact of the ontology on the information extraction process.

In the system that uses multiple ontologies, a decision has to be made as to which ontology is to be used when processing webpages of a university. The approach adopted in the system is to use the ontology defined for the region of the university. Therefore, the ontology for North American ontology is used for universities from Canada and USA while the other ontology is used for universities from other countries. The region of the university is identified by analyzing the hostname of the URL of the university website - .edu and .ca (and .us, which is rarely used by university websites) represent North American universities while other domains represent universities from other countries. It should be noted that this approach is suitable when different ontologies represent different sub-domains. When multiple ontologies represent different perspectives, it makes sense to perform extractions using all the ontologies.

The OWL files produced by the Ontology Language Handler component, which contain instances and their property values, constitute the final output of the system. Precision and recall of the system are established by comparing these against a gold standard for the same documents specified by a human.

4.1.4 Implementation

As mentioned earlier, the regular expressions that make the actual extractions from text (known as extraction rules) were identified by analyzing the webpages of the training set. Since the success of the entire system hinges on these rules, a considerable effort was spent on correctly identifying them. Once these rules were identified, they were written as JAPE grammars that can be used by GATE.

The following JAPE statement represents the regular expressions that were used to identify values for the "isFoundedOn" property whose domain includes the university class.

```
(
({Token.string == "founded" | Token.string == "Founded"})
({Token.string == "in" | Token.string == "In"} |
{Token.string == "on" | Token.string == "On"})
({Token.category == "CD"}): year
)
```

Here "Token" denotes string tokens recognized by the tokenizer for the English language used by GATE. The "category" of a token is the POS tag assigned to it by GATE POS tagger. The category "CD" represents cardinal numbers. Hence the above JAPE rule accepts phrases such as "founded in 1837". The token that represents the number is named "year" and it is annotated as a value of the "isFoundedOn" property using the other parts of the JAPE rule (not shown here).

Similar JAPE rules were written for all the classes and properties of the ontology, whose instances and property values were expected to be extracted by the system. Separate JAPE rules were written for the three ontologies in separate files. Some JAPE rules such as the one represented above were used all three ontologies since they did not change depending on the ontology.

The Corpus Builder component was implemented to use Google queries as designed. In order to retrieve the webpages relevant to different components of the ontology, it conducts a series of queries on the web domain of the university in concern. For instance, to find the documents directly related to the different attributes of a university such as the year it was founded and its location, the Corpus Builder queries for the terms "about" and "quick facts". It was seen that these queries retrieve the relevant webpages by experimenting with universities in the training set. For this query,

Table 2: Summary of the results obtained

System	Domain	Precision(%)	Recall(%)
Multiple Ontologies	North American universities	58.75	47.00
	Other universities	53.85	51.85
	All universities	**56.82**	**48.70**
Single Ontology	North American universities	56.94	41.00
	Other universities	48.00	44.44
	All universities	**53.28**	**42.21**

the top two pages returned by Google were added to the corpus.

Although the developed ontologies covered many different aspects of a university, it was decided to restrict the implementation to a subset of these features in this stage. As such information extraction was carried out only on the pages returned by queries on "about" and "quick facts" and the classes and properties found in these pages were selected for extraction. These includes classes such as *Department* and *UniversitySystem* (the latter defined only in the ontology for North American universities) and properties such as *hasPresident* and *isLocatedIn*.

The programs were executed on an unremarkable personal computer. For a given university, they were capable of producing results within few minutes.

4.1.5 Results and Discussion

Both the single-ontology system and the system that uses multiple ontologies were successfully implemented. The results produced by them, in the form of OWL files containing instances and property values, were compared against human specified gold standards. The precision and recall for each system were computed through this process.

It was observed that in some situations the systems have made partially correct extractions. For example, in identifying the name of a department or a school, the systems might not include the last word of the name or add an additional word incorrectly. In these situations, the extractions were recognized as partially correct and a figure of 0.5 was used for computing precision and recall (as opposed to 1.0 allocated to correct extractions).

Table 2 presents the calculated precision and recall for each system. It shows the precision and recall for each sub-domain (North American universities and universities from other parts of the world) as well as for all universities. It should be noted that a figure for all universities is not the average of the corresponding figures for North American and non-North American universities since the number of extractions made for the two sub-domains are different.

It can be seen that these figures, especially the figures for precision, are somewhat lower than results obtained by other IE and OBIE systems. For example, the Kylin OBIE system [52] has consistently recorded figures around 90% for precision but its recall is in a comparable range with this system. The OBIE system implemented by Saggion et al. [49] have shown a precision higher than 90% and a recall higher than 60% in a majority of their test cases. In addition, the information extraction systems presented in Message Understanding Conferences have been able to identify instances for concepts such as persons, organizations and locations with precisions exceeding 90% [6].

It appears that the following are the main reasons that have contributed towards the drop of performance in this system.

1. *Difficulties associated with processing webpages with complex structures:* Most IE and OBIE systems process pure text files or webpages with simple structures such as news articles or

webpages with a specific structure (e.g., Wikipedia pages). But these systems use webpages from different websites, which often contain complex structure such as frames and pop-up menus. It was seen that the information extraction system has encountered difficulties in processing them.

2. *Errors in Information Retrieval:* As mentioned earlier, the systems employ information retrieval as a pre-processing step to find the relevant webpages from the set of all webpages in the web domain of a university. It was seen that this process has sometimes returned incorrect results. For instance, personal webpages or webpages of different research groups have sometimes been retrieved when searching for "about" and "quick facts". These webpages have resulted in many incorrect extractions.

3. *Possible improvements for extraction rules that were not detected in the training set:* It was seen that some possible extraction rules have not been recognized in analyzing the training set. The inclusion of these rules would have improved precision and recall.

It can be seen that the system that uses multiple-ontologies has recorded a slightly higher recall (as well as a slightly higher precision) than the single-ontology system. This appears to provide some support to the hypothesis that the use of multiple-ontologies would result in an improvement in recall. However, it is prudent not to consider this result as a definitive piece of evidence that proves the superiority of OBIE systems that use multiple ontologies; this is just a single experiment and the observed improvement in recall are not very large although they appear to be significant. Further experiments and analyses would be required before a decision can be made on the validity of this hypothesis.

However, this work does prove that ontology-based information extraction systems can be designed and implemented to make use of multiple ontologies. To the best of my knowledge, this is the first OBIE system that uses multiple ontologies in this manner.

It was also observed that the definition of recall becomes a bit fuzzy when performing information extraction with multiple ontologies. For information extraction, the following is used as the definition for recall as mentioned earlier.

$$Recall = \frac{|\ correct\ answers\ |}{|\ correct\ answers\ in\ the\ gold\ standard\ |}$$

However when using multiple ontologies, it has to be decided whether to establish the gold standard with respect to the concepts found in the ontology used in the particular task or with respect to the concepts found in all the ontologies. The value obtained for recall might change depending on which definition is used for the gold standard.

For example, the common ontology used by the single-ontology system described here does not have a class for University Systems, since it was thought that it was a concept found only in North American universities. However, a University System class was defined in the ontology for North American universities. If a university system is specified in a particular webpage, a decision has to be made whether to include that in the gold standard when processing that webpage through the single-ontology system. It can be seen that including such instances in the gold standard would reduce the recall for the single-ontology system. Further, this approach would better represent the advantages of using multiple ontologies.

In evaluating the recall of the system, only the instances relevant to the current ontology were specified in the gold standard. The figures for precision and recall do not significantly change even when all possible extractions, with respect to all the ontologies, are included in the gold standard. However, in other systems this might result in significant differences in the reported recall.

4.1.6 Possible Improvements

This implementation can be improved and extended in several ways. The following are some of them.

1. *Improving the Information Retrieval process:* It may be possible to improve the IR process employed by the system by using techniques such as following promising hyperlinks (e.g., links for "about university") together with Google search. This will reduce the number of irrelevant documents added to the corpus which would in turn improve both precision and recall.

2. *Improving the Information Extraction Module:* As mentioned earlier, some possible improvements for extraction rules were detected when evaluating results. In addition, making use of an HTML parser in the information extraction module may also improve performance.

3. *Extending the system to cover all the parts of the ontologies:* As mentioned earlier, the current systems make extractions related to some parts of the ontology only. Extending the system to cover the entire ontology, by writing extraction rules for all the classes and properties of the ontologies, would result in a better experiment on the use of multiple ontologies.

4.2 Extracting Information from Teen Emails

4.2.1 Objective of the Project

This project was conducted in collaboration with Dr. Stephen Fickas and one of his graduate students, Mahshid Mohammadi. The source data of the project consisted of a set of emails written by mentally challenged teenagers. These emails are monitored by a group of reviewers in order to judge the mental capabilities of the teenagers. Their main task is evaluating whether the children are capable of writing meaningful replies for the emails they receive. In performing this task, the reviewers compare the source emails with reply emails and categorize the logical dependence between the source and reply as "absent","adequate" and "achieved". (The term *contingency* is used to denote the logical dependence between source and reply emails.) The objective of the project was to explore whether this task can be performed using NLP techniques.

4.2.2 Design and Implementation

It was decided to approach this problem as a classification task; each pair of emails have to be categorized as "absent", "adequate" or "achieved" in terms of the level of contingency. It was expected to use several linguistic features as the input to the classification process. These include the fraction of common words among all the words in the two emails, the fraction of words with common meanings in the two emails (the synsets of WordNet can be used for this purpose) and some special features. One such special feature is based on the detection of situations where the source email ends with a question and the reply email consists of a single word (a quite frequent occurrence).

In addition to the features mentioned above, it was expected to derive one feature from the output of an OBIE system. Here, the objective was to determine whether the source and reply emails contain related concepts of an ontology. It was expected to identify the instances of classes of an ontology in the source email and determine whether instances of the same classes or possible property values of the instances are found in the reply emails. It was intended to use the range of properties (in terms of the POS tags) to find possible occurrences of property values in the reply emails. The fraction of source email instances with a corresponding instance or a property value in the reply email was expected to be used as a feature for the classifier.

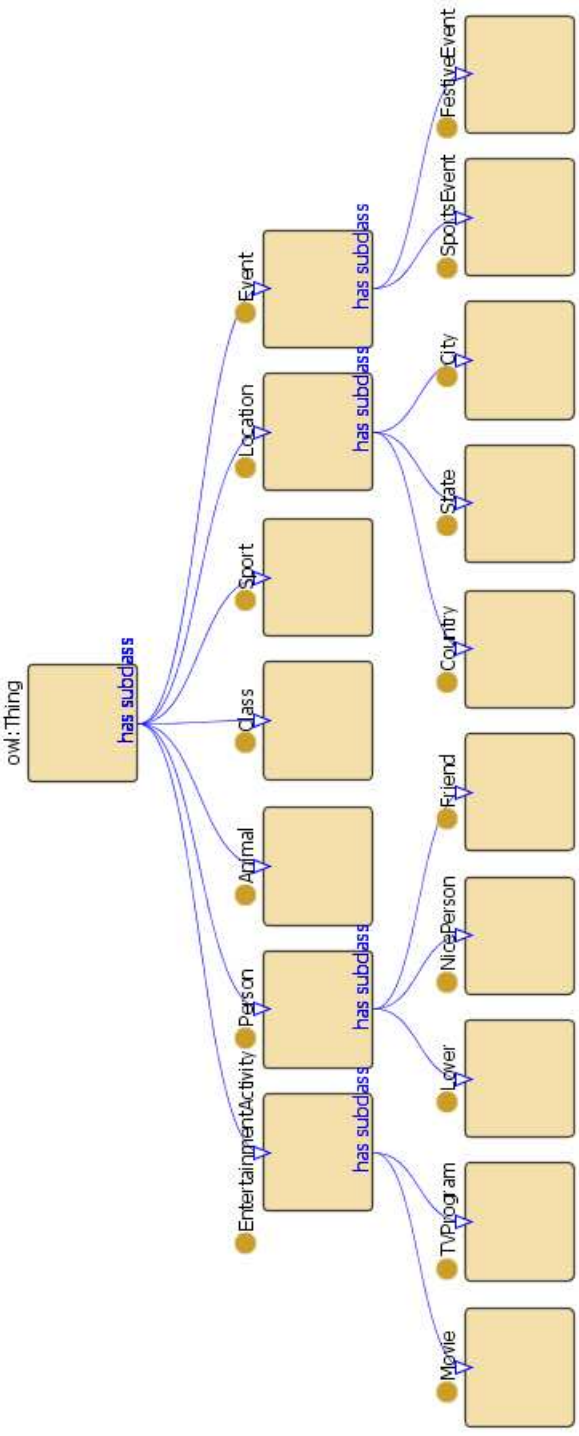

Figure 5: The classes of the ontology used for teen emails

Table 3: Summary of the results obtained

	Precision(%)	**Recall(%)**
Source Emails	81.25	35.14
Reply Emails	50.00	23.68
All Emails	70.00	31.25

The pairs of emails available were divided into a training set and test set in order to develop a classifier. Around 60% of the emails were allocated to the training set while the rest were used as the test set. For the OBIE system, an ontology was constructed by studying the emails of the training set. This ontology represents the main concepts found in the emails. Figure 5 shows the classes of this ontology.

The OBIE system was constructed to recognize instances of classes of the ontology. As in the implementation on university ontologies, linguistic extraction rules were used as the information extraction technique. These were specified using JAPE rules of GATE, as in the case of the implementation on university ontologies. In addition, gazetteers were used to identify instances of some classes. For example, a list of animals were used to identify instances of the "Animal" class. This list was obtained from a website[11].

4.2.3 Results and Discussion

The results obtained for the OBIE system were moderately successfully. Table 3 shows the precision and recall calculated for the test set. It shows separate figures for source and reply emails as well as figures for the entire set of emails. Since the OBIE systems only extracts instances of classes, the gold standard consisted of instances only and did not include any property values.

It was seen that many emails are very short and that this reduces the effectiveness of the information extraction process. Reply emails were generally shorter than source emails and this might explain the drop of performance for reply emails that can be observed in table 3. In addition, the frequent occurrence of spelling mistakes was also a hindrance to the information extraction process. The use of an automatic spelling corrector[12] did not significantly improve the situation mainly because it was unable to determine correct spellings for many unorthodox words.

While the implementation of the OBIE system was moderately successful, the main project was not completed. This decision was partly caused by the fact that the initial results were not very encouraging. It is possible that the underlying problem is too hard for NLP techniques.

5 Directions for Future Research Work

As mentioned earlier, ontology-based information extraction is a relatively new field with a lot of potential. The following are some directions in which future research work on this area can be conducted.

5.1 Exploring Use of Multiple Ontologies in OBIE

As mentioned in section 4.1.1, multiple ontologies developed for the same domain belong to one of the following categories.

[11] `http://www.rickwalton.com/curricul/lanimals.htm`

[12] `http://jazzy.sourceforge.net/`

1. Providing different perspectives
2. Specializing in sub-domains

That section also discussed the opportunities and challenges involved with the use of multiple ontologies in OBIE. A possible improvement in recall was recognized as one of the main opportunities in the use of multiple ontologies. The ability of this approach to support multiple perspectives (possibly in a query answering system developed based on the OBIE system) and the ability to make use of mappings were seen as additional opportunities. However, there are complex issues involved with all these factors, such as the ontology *not* having any direct influence on recall (although it has some indirect influence because of its effects on the IE process), which have to be resolved to realize these opportunities.

The work presented in section 4.1 can be seen as an initial work on the use of multiple ontologies in OBIE. Further research work that use multiple ontologies from different domains and different IE techniques are necessary to obtain a better understanding of the opportunities and challenges involved with this approach. Such research work should investigate different issues involved with this approach. The following are some of such issues.

- **Use of multiple ontologies related to different perspectives:**

  The implementation in section 4.1 uses multiple ontologies specializing in different sub-domains. The use of multiple ontologies that provide different perspectives on the same domain, without specializing on sub-domains, would be more complex and would probably present more advantages. In this case, each document of the corpus should be processed with respect to all the ontologies since it does not make sense to select a single ontology from the set of ontologies for a given document (as done in section 4.1). It can be seen that this can be performed in one of the following manners.

  1. *Merging the ontologies in to a single ontology and using the merged ontology for IE:* Research work has shown that it is possible to merge different, independent ontologies into a single, merged ontology [17]. Mappings between the concepts of the ontologies are necessary for this exercise. Following this approach, it is possible to merge the ontologies in concern into a single ontology and then use that ontology for IE. It should be noted that this merged ontology contains concepts from all the original ontologies and no knowledge is lost during merging. In order to apply this approach, the mappings between the concepts on the ontologies should be known in advance.
  2. *Performing IE for each ontology and merging the extractions as necessary:* In this approach, IE is performed separately for each ontology. Mappings between the ontologies are not required beforehand. In addition, the extractions made may actually help the discovery of mappings. However it would be necessary to discover the mappings correctly (after the IE process) in order to make use of all the extractions in query answering.

- **Query answering using extractions made from multiple ontologies:**

  Mappings between the concepts of the ontologies are needed in answering queries using extractions made with respect to different ontologies. For example, if one ontology has a class named "Company" and another has a class named "Organization", it should be known that all companies are organizations (which can be formally expressed as a mapping) in order to include all extractions for "Company" in answering the question "What are the organizations found in the documents?". In addition, query answering involves handling the problem of reference reconciliation (object reconciliation), which refers to the task of determining whether

two objects are one and the same [16]. For the example stated above, it would be necessary to identify whether an extraction made for the "Organization" class and an extraction made for the "Company" class refer to the same real world entity. Further, conflicts between the extractions made with respect to different ontologies will have to be resolved in answering queries.

It should be noted that these complications arise only when the different ontologies are not merged into a single ontology, as described above. It can be expected that such merging will not be possible at least in some situations because of the unavailability of mappings.

- **Relationship with mappings:**

  As mentioned earlier, mappings between the concepts of different ontologies are essential for using multiple ontologies in OBIE. In addition, extractions made with respect to different ontologies may guide the discovery of mappings. For example, if all extractions made for instances of class A of ontology O_1 are included as instances of class B of ontology O_2, it may be hypothesized that class A of ontology O_1 is a subclass of class B of ontology O_2. It should be investigated whether intuitions like this can be formalized into algorithms that discover mappings between ontologies.

- **Formalizing different perspectives on recall:**

  As mentioned in section 4.1.5, problems arise when comparing the recall of a multiple-ontology OBIE system with that of a single-ontology OBIE system. The question here is whether to establish the gold standard for the single-ontology system with respect to the ontology it uses or with respect to all the ontologies. It can be seen that the advantages of using multiple ontologies are better represented when the gold standard is established with respect to all ontologies. It should be possible to formally define two types of recall to address this issue: a *local recall* with respect to the ontology used by the system and a *global recall* with respect to all the ontologies.

5.2 Developing a Generic Framework for OBIE

Although several OBIE systems have been developed so far they normally concentrate on a particular domain and/or a particular source. They normally do not attempt to come up with a generic framework for ontology-based information extraction which can be applied in different domains and different sources. It can be argued that such frameworks are necessary to realize the full potential of OBIE, such as its ability to create semantic contents for the Semantic Web.

Hence, designing a generic framework for ontology-based information extraction should be an important research exercise. The following factors should be important in designing such a framework.

- **Standardizing the use of human knowledge in IE:**

  All IE techniques rely on human knowledge. In techniques such as classification, human knowledge is provided using a formal set of training examples with manually specified annotations while in some techniques such as linguistic extraction rules a human just specified some rules after studying the training examples. If a framework is to be developed for OBIE, a standard mechanism has to be used to specify the human knowledge that will be used by the system. It is possible that the annotations made by a human in the documents of the training set with respect to a given ontology would account for most of the human knowledge used by the system. Standard mechanisms would be necessary to specify additional knowledge.

For the task of manually annotating text with ontological concepts (in order to provide the training examples for the OBIE framework), easy-to-use GUI tools will be needed. Currently, it appears that this task is performed without the use of any software tools in some research work. GATE [1] provides a tool named OCAT (Ontology-Based Corpus Annotation Tool) for this purpose. It is possible to annotate words or phrases as instances of classes with this tool. However, it does not support annotating text as property values. It should be possible to adopt an existing tool such as OCAT of GATE for the task of creating training examples for the framework. Since most of such tools, including GATE, are developed as open source software, it is possible to make some changes in the tools if necessary.

- **Reusing the knowledge captured by IE techniques:**

  When an IE systems is successfully developed, it contains some valuable *knowledge* that can be used to extract information from natural language text. In some information extraction techniques such as decision trees (a classification technique) and linguistic extraction rules, this knowledge can be explicitly stated while in some other techniques such as neural networks and Support Vector Machines (classification techniques) the knowledge is implicit. When an OBIE system is developed, it contains knowledge regarding how to extract instances of individual classes and values of individual properties. Hence, it would be quite useful to develop a mechanism to reuse this knowledge in developing new OBIE systems. This may be easier in IE techniques that can explicitly state their knowledge. For example, if one OBIE system has developed a decision tree to identify instances of "Location" class, it would be advantageous to reuse these rules in designing a new ontology that contains a "Location" class with the same meaning.

  In other words, what is necessary is some kind of *coupling* between the IE technique and the ontology. The attempts made by Embley [20] to define *extraction ontologies* and by Yildiz and Miksch [55] to include linguistic extraction rules as annotation properties of OWL can be seen as work aimed towards this objective. However, as mentioned earlier, the ontology itself may not be the correct place to keep this kind of knowledge. One option might be to use some kind of a special repository to store the knowledge captured by an OBIE system in extracting information related to an ontology. A semi-structured format such as XML may be suitable for this purpose.

- **Mining text for ontologies:**

  Currently no suitable ontologies exist for many domains. Further, there are many limitations in several existing domain ontologies since many of them were initially developed as "toy ontologies". Therefore a framework for OBIE should be capable of mining an ontology (or ontologies) for the domain in concern from the text itself. This may be a semi-automatic process since ontology construction is a difficult process that can be considered AI-complete. However, it is better to design an ontology construction process that does not rely on specific structures such as infoboxes of Wikipedia (which the ontology generation component of the Kylin OBIE system relies on [54]) in order to make it more generic.

- **Improving the ontologies through IE:**

  Since it is not easy to come up with a perfect ontology for a domain, the OBIE framework should allow updating the ontology through the information extraction process. New sub-classes and properties may be identified and some concepts may be removed from the ontology. As mentioned in section 3.2.2, some existing OBIE systems attempt to perform this function.

The algorithms employed by these systems may have to be improved and generalized in order to be included in a framework for OBIE.

As a final point regarding the development of a framework for OBIE, such a framework should not require the existence of multiple ontologies for a given domain. The framework should be able to work with a single domain ontology. However, it is better if it also supports multiple ontologies.

5.3 Integration of Different IE Techniques in OBIE Systems

Different OBIE systems make use of different IE techniques as mentioned earlier. It is possible that a combination of these techniques would produce better results than a single technique. Hence, integrating different IE techniques in OBIE systems (as well as in general IE systems) should be an important research work. Such integrated IE techniques can be used by OBIE frameworks described above.

The use of classification is one way to integrate different IE techniques. In this approach, the IE system resulting from each technique is considered a feature for the classifier. The output of the larger system is based on the output of this classifier. A technique such as decision trees, which produce explicit rules may be more suitable for the classifier than techniques with implicit rules. Romano et al. [47] have developed an IE system that employs this principle for extracting information from medical narrative reports (notes written by physicians). Applying similar techniques in OBIE would be an interesting exercise.

It can be seen that linguistic extraction rules and classification techniques are widely used by current OBIE systems. In addition, these techniques have been extensively studied in the literature of information extraction. Hence, it can be expected that these techniques would play a major rule in an integrated IE technique. However, recent works have shown the potential of techniques such as web-based search, gazetteers and HTML parsing. Therefore, an integrated IE technique should attempt to make use of these techniques as well whenever possible. In short, an integrated IE technique should attempt to make use of all the techniques available for the difficult task of information extraction.

5.4 Text Mining for Linguistic Extraction Rules

Linguistic extraction rules that use regular expressions are employed by many IE and OBIE systems. These rules are generally specified by humans who discover them by reading text. However, it should be possible to develop algorithms to discover such regular expressions automatically. This can be considered a text mining task since it involves the discovery of hidden knowledge from text (in the form of rules that can be used to extract certain types of information).

Romano et al. [47] have achieved some success in mining for such linguistic extraction rules. They have employed an efficient algorithm [44] for the Longest Common Subsequence problem (the problem of finding the longest subsequence common to all sequences in a set of sequences) for this purpose. They first find the set of sentences that contain a particular type of information and treat these sentences as sequences of words. Then they discover the longest common subsequence of words for these sentences using the algorithm for the Longest Common Subsequence problem mentioned above. Finally, they discover linguistic rules by analyzing the longest common subsequence for the sentences and the number of characters that can occur between individual words of the subsequence. Results have shown that this technique is capable of discovering good linguistic extraction rules.

It is possible that this technique can be generalized and improved in several ways. One possible improvement is to use a lexical semantic database to discover more general rules. Synsets of WordNet [21] that contain words with the same meaning can be used for this purpose. For example,

if two sentences contain the terms "located" and "situated", they can be considered as the same token (corresponding to a synset of WordNet) for the purpose of discovering subsequences. Another possible improvement is the use of Part-Of-Speech (POS) tags in discovering common subsequences and in defining the words that can occur between two words of the common subsequence. It may be reasonable to develop an algorithm that works on different levels - words, synsets and POS tags.

It should be noted that manually discovering linguistic extraction rules is a quite difficult task. The difficulty of this task was observed in the implementations presented in this paper where it was later seen that many useful patterns have not been discovered. As such, algorithms that automatically discover such extraction rules would improve the effectiveness of this IE technique. These algorithms would still require human specified annotations in the text. While annotating text in this manner (for example, marking all the names of terrorist organizations in a set of news articles) is a repetitive task, it is not as difficult as discovering extraction rules.

5.5 Developing Semantic Web Interfaces for OBIE systems

As mentioned earlier, the ability to generate semantic contents for the Semantic Web is one of the major factors that make OBIE an interesting research field. However, the manner in which such contents are to be integrated with the Semantic Web has not been clearly identified.

One approach is to implement semantic web interfaces for each website. This approach is sometimes known as implementing *wrappers* for the websites [31]. (However, the word wrapper is used as a synonym for an information extraction system in some other work [30]). An alternative approach would be to establish semantic content providers for different domains independent of the individual websites. Such content providers would perform OBIE on several websites and present the output to the software agents of the Semantic Web. Exploring how to practically implement either type of interface would constitute an important research work.

6 Conclusion

This paper presents the results of my studies on Ontology-Based Information Extraction. These studies were of two types - reviewing current research work on the field and doing some implementations to explore the field. Reviewing the current research work enabled me to formulate a definition for an OBIE system, to identify a common architecture for OBIE systems and to classify existing OBIE systems along their key dimensions. Further, these studies allowed me to obtain a good understanding of the design and implementation of OBIE systems. The implementation work allowed me to make practical use of the knowledge obtained on OBIE systems by reviewing current research work. It also allowed me to get familiar with the widely used software tools and techniques of the field. In addition, these implementations produced some interesting results. In particular, the implementation which used multiple ontologies for information extraction (from university websites) proves that it is possible to use multiple ontologies for OBIE and explores the opportunities and challenges of this approach.

These studies enabled me to achieve a thorough understanding of Ontology-Based Information Extraction and to get some practical experience on conducting research work in the development of OBIE systems. Directions provided by my advisor, Dr. Dejing Dou, and other committee members were invaluable in these studies.

I am quite confident that the studies presented in the paper have prepared me well for future research work in the field of Ontology-Based Information Extraction. The directions for future research work presented in section 5 would provide a guide for such research work.

7 Acknowledgements

I would like to thank my advisor, Dr. Dejing Dou, as well as the other committee members - Dr. Arthur Farley and Dr. Michal Young - for their help in the studies that produced this paper. In addition, I would like to thank Dr. Stephen Fickas and his student, Mahshid Mohammadi for their involvement in the project on teen email data and Dr. Sarah Douglas for her help in finding good textbooks for the field of Natural Language Processing.

References

[1] General Architecture for Text Engineering (GATE).
`http://www.gate.ac.uk/`.

[2] OWL Web Ontology Language.
`http://www.w3.org/TR/owl-ref/`.

[3] The protégé ontology editor and knowledge acquisition system.
`http://protege.stanford.edu/`.

[4] SProUT (Shallow Processing with Unification and Typed Feature Structures).
`http://sprout.dfki.de/`.

[5] *Proceedings 1st International and KI-08 Workshop on Ontology-based Information Extraction Systems*, volume 400. DFKI, 2008.

[6] C. Aone, L. Halverson, T. Hampton, and M. Ramos-Santacruz. SRA: Description of the IE2 system used for MUC-7. In *Proceedings of the 7th Message Understanding Conference*, 1998.

[7] D. E. Appelt, J. R. Hobbs, J. Bear, D. J. Israel, and M. Tyson. Fastus: A finite-state processor for information extraction from real-world text. In *IJCAI*, pages 1172–1178, 1993.

[8] T. Berners-Lee. Cleaning up the User Interface.
`http://www.w3.org/DesignIssues/UI.html`.

[9] T. Berners-Lee, J. Hendler, and O. Lassila. The Semantic Web. *Scientific American*, 284(5), May 2001.

[10] P. Buitelaar and M. Siegel. Ontology-based information extraction with soba. In *In: Proc. of the International Conference on Language Resources and Evaluation (LREC)*, pages 2321–2324, 2006.

[11] E. Charniak. *Statistical Language Learning (Language, Speech, and Communication).* The MIT Press, 1996.

[12] N. Choi, I.-Y. Song, and H. Han. A survey on ontology mapping. *SIGMOD Rec.*, 35(3):34–41, 2006.

[13] P. Cimiano, S. Handschuh, and S. Staab. Towards the self-annotating web. In *WWW*, pages 462–471, 2004.

[14] P. Cimiano, G. Ladwig, and S. Staab. Gimme' the context: context-driven automatic semantic annotation with c-pankow. In *WWW*, pages 332–341, 2005.

[15] O. Dekel, J. Keshet, and Y. Singer. Large margin hierarchical classification. In *ICML '04: Proceedings of the twenty-first international conference on Machine learning*, page 27, New York, NY, USA, 2004. ACM.

[16] X. Dong, A. Y. Halevy, and J. Madhavan. Reference reconciliation in complex information spaces. In *SIGMOD Conference*, pages 85–96, 2005.

[17] D. Dou, D. McDermott, and P. Qi. Ontology Translation by Ontology Merging and Automated Reasoning. In *Proceedings of EKAW Workshop on Ontologies for Multi-Agent Systems*, pages 3–18, 2002.

[18] D. Dou, D. V. McDermott, and P. Qi. Ontology Translation on the Semantic Web. *Journal of Data Semantics*, 2:35–57, 2005.

[19] T. Q. Dung and W. Kameyama. Ontology-based information extraction and information retrieval in health care domain. In *DaWaK*, pages 323–333, 2007.

[20] D. W. Embley. Towards semantic understanding – an approach based on information extraction ontologies. In *ADC*, page 3, 2004.

[21] C. Fellbaum. *WordNet: An Electronic Lexical Database (Language, Speech, and Communication)*. The MIT Press, May 1998.

[22] B. C. Grau, B. Parsia, and E. Sirin. Working with multiple ontologies on the semantic web. In *International Semantic Web Conference*, pages 620–634, 2004.

[23] T. Gruber. Ontolingua: A Translation Approach to Providing Portable Ontology Specifications. *Knowledge Acquisition*, 5(2):199–220, 1993.

[24] U. Hahn and K. Schnattinger. Towards text knowledge engineering. In *AAAI '98/IAAI '98: Proceedings of the fifteenth national/tenth conference on Artificial intelligence/Innovative applications of artificial intelligence*, pages 524–531, Menlo Park, CA, USA, 1998. American Association for Artificial Intelligence.

[25] J. Han and M. Kamber. *Data Mining: Concepts and Techniques*. Morgan Kaufmann Publishers, 2001.

[26] M. A. Hearst. Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the 14th conference on Computational linguistics*, pages 539–545, Morristown, NJ, USA, 1992. Association for Computational Linguistics.

[27] J. R. Hobbs, M. Stickel, P. Martin, and D. Edwards. Interpretation as abduction. In *Proceedings of the 26th annual meeting on Association for Computational Linguistics*, pages 95–103, Morristown, NJ, USA, 1988. Association for Computational Linguistics.

[28] C. H. Hwang. Incompletely and imprecisely speaking: Using dynamic ontologies for representing and retrieving information. In *Knowledge Representation Meets Databases*, pages 14–20, 1999.

[29] J. Kietz, A. Maedche, and R. Volz. A method for semi-automatic ontology acquisition from a corporate intranet. *EKAW-2000 Workshop "Ontologies and Text", Juan-Les-Pins, France, October 2000*, 2000.

[30] S. Kuhlins and R. Tredwell. Toolkits for generating wrappers. In *NetObjectDays*, pages 184–198, 2002.

[31] A. H. F. Laender, B. A. Ribeiro-Neto, A. S. da Silva, and J. S. Teixeira. A brief survey of web data extraction tools. *SIGMOD Record*, 31(2):84–93, 2002.

[32] Y. Li and K. Bontcheva. Hierarchical, perceptron-like learning for ontology-based information extraction. In *WWW*, pages 777–786, 2007.

[33] Y. Li, K. Bontcheva, and H. Cunningham. Using uneven margins svm and perceptron for information extraction. In *In Proceedings of Ninth Conference on Computational Natural Language Learning (CoNLL-2005)*, 2005.

[34] J. Lu, L. Ma, L. Zhang, J.-S. Brunner, C. Wang, Y. Pan, and Y. Yu. Sor: A practical system for ontology storage, reasoning and search. In *VLDB*, pages 1402–1405, 2007.

[35] A. Maedche, B. Motik, and L. Stojanovic. Managing multiple and distributed ontologies on the semantic web. *VLDB J.*, 12(4):286–302, 2003.

[36] A. Maedche, G. Neumann, and S. Staab. Bootstrapping an ontology-based information extraction system. pages 345–359, 2003.

[37] A. Maedche and S. Staab. The Text-To-Onto Ontology Learning Environment. *Software Demonstration at ICCS-2000-Eight International Conference on Conceptual Structures. August*, pages 14–18, 2000.

[38] C. Manning and H. Schütze. *Foundations of Statistical Natural Language Processing*. The MIT Press, 1999.

[39] D. Maynard. Metrics for evaluation of ontology-based information extraction. In *In WWW 2006 Workshop on Evaluation of Ontologies for the Web*, 2006.

[40] D. Maynard, W. Peters, and Y. Li. Metrics for evaluation of ontology-based information extraction. In *WWW 2006 Workshop on Evaluation of Ontologies for the Web (EON)*, 2006.

[41] L. McDowell and M. J. Cafarella. Ontology-driven information extraction with ontosyphon. In *International Semantic Web Conference*, pages 428–444, 2006.

[42] M.-F. Moens. *Information Extraction: Algorithms and Prospects in a Retrieval Context (The Information Retrieval Series)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[43] H. M. Müller, E. E. Kenny, and P. W. Sternberg. Textpresso: an ontology-based information retrieval and extraction system for biological literature. *PLoS Biology*, 2(11), November 2004.

[44] E. W. Myers. An o(nd) difference algorithm and its variations. *Algorithmica*, 1(2):251–266, 1986.

[45] E. Riloff. Information extraction as a stepping stone toward story understanding. *Understanding language understanding: computational models of reading*, pages 435–460, 1999.

[46] J. J. Ritsko and D. I. Seidman. Preface. *IBM Systems Journal*, 43(3):449–450, 2004.

[47] R. Romano, L. Rokach, and O. Maimon. Automatic discovery of regular expression patterns representing negated findings in medical narrative reports. In *NGITS*, pages 300–311, 2006.

[48] S. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*. Prentice-Hall, Englewood Cliffs, NJ, 2nd edition edition, pages 848-850, 2003.

[49] H. Saggion, A. Funk, D. Maynard, and K. Bontcheva. Ontology-based information extraction for business intelligence. In *ISWC/ASWC*, pages 843–856, 2007.

[50] A. Stavrianou, P. Andritsos, and N. Nicoloyannis. Overview and semantic issues of text mining. *SIGMOD Record*, 36(3):23–34, 2007.

[51] R. Studer, V. R. Benjamins, and D. Fensel. Knowledge engineering: Principles and methods. *Data Knowledge Engineering*, 25(1-2):161–197, 1998.

[52] F. Wu, R. Hoffmann, and D. S. Weld. Information extraction from wikipedia: moving down the long tail. In *KDD*, pages 731–739, 2008.

[53] F. Wu and D. S. Weld. Autonomously semantifying wikipedia. In *CIKM '07: Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, pages 41–50, New York, NY, USA, 2007. ACM.

[54] F. Wu and D. S. Weld. Automatically refining the wikipedia infobox ontology. In *WWW*, pages 635–644, 2008.

[55] B. Yildiz and S. Miksch. ontox - a method for ontology-driven information extraction. In *ICCSA (3)*, pages 660–673, 2007.