Ontology-based information extraction: An introduction and a survey of current approaches

Daya C. Wimalasuriya and Dejing Dou

Journal of Information Science 2010; 36; 306 originally published online Mar 19, 2010;
DOI: 10.1177/0165551509360123

JIS

Ontology-based information extraction: An introduction and a survey of current approaches

Daya C. Wimalasuriya and Dejing Dou

Department of Computer and Information Science, University of Oregon, Eugene, OR, USA

Abstract.

Information extraction (IE) aims to retrieve certain types of information from natural language text by processing them automatically. For example, an IE system might retrieve information about geopolitical indicators of countries from a set of web pages while ignoring other types of information. 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 IE 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. In this paper, we provide an introduction to ontology-based information extraction and review the details of different OBIE systems developed so far. We attempt to identify a common architecture among these systems and classify them based on different factors, which leads to a better understanding on their operation. We also discuss the implementation details of these systems including the tools used by them and the metrics used to measure their performance. In addition, we attempt to identify the possible future directions for this field.

Keywords: information extraction; ontologies; Semantic Web

1. Introduction

The general idea behind information extraction (IE) is automatically retrieving certain types of information from natural language text. According to Russell and Norvig [1], 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 IE is a form of natural language processing in which certain types of information must be recognized and extracted from text [2].

As an example of an IE system, we can describe a system that processes a set of web pages and extracts information regarding countries and their political, economic and social indicators. Some kind of model that specifies what to look for (e.g. country name, population, capital, main cities,

Correspondence to: Daya C. Wimalasuriya, Department of Computer and Information Science, University of Oregon, Eugene, OR 97403, USA. Email: dayacw@cs.uoregon.edu

Journal of Information Science, 36 (3) 2010, pp. 306–323 © The Author(s), 2010. Reprints and Permissions: http://www.sagepub.co.uk/journalsPermissions.nav, DOI: 10.1177/0165551509360123

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 IE 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 [1]. Studies on IR 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 IE systems lies in between these two categories, their success has also been somewhere in between the levels achieved by IR and text understanding systems.

Ontology-based information extraction (OBIE) has recently emerged as a subfield of IE. 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 [3, 4]. Generally, ontologies are specified for particular domains. Since IE is essentially concerned with the task of retrieving information for a particular domain, formally and explicitly specifying the concepts of that 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 IE system described earlier. This is the general idea behind OBIE.

It appears that the term 'ontology-based information extraction' was only conceived a few years ago. But some work related to this field has been carried out earlier (e.g. work by Hwang [5] 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 [6]. Several of these systems are related to ongoing projects. This, together with the fact that the interest in IE in general is increasing, indicates that this field could experience a significant growth in the near future.

Although the field of information extraction appears to be at least few years old it appears that there have not been any serious attempts review the literature of the field and to provide an introduction to the field based on the ideas present in the literature. One aspect of this task is to provide a definition for an OBIE system so that it is possible to clearly identify whether a given system is an OBIE system or not. In addition, it is necessary to analyze the architectures of different OBIE systems and figure out the commonalities between them. It is also useful to classify the existing OBIE systems (with respect to suitable dimensions) so that the specialities of different systems can be recognized and new systems can be easily compared against existing ones. Reviewing the implementation details of different OBIE systems and presenting the metrics that researchers have used to evaluate the performance of them will also be helpful in developing new systems.

In this article, we have attempted to provide a review for the field of OBIE meeting the above-mentioned challenges. Whenever possible we have used common ideas found through our literature review. We have used such common ideas in arriving at a definition for an OBIE system and in presenting the performance metrics used with OBIE systems. In some other areas, we have attempted to provide new insights such as in identifying a common architecture among different OBIE systems and classifying them along a set of dimensions we have identified. Altogether, we have attempted to provide a clear and concise presentation of the current state of affairs and possible future directions in the field of OBIE.

This section was aimed at providing a general description of the field. In the rest of the paper, we review the literature of the field and provide our insights on them. In Section 2, we provide a definition for an OBIE system based on existing ideas. In Section 3, we discuss the common architectures of OBIE systems and their general functionality. We classify the current OBIE systems along different dimensions in Section 4. Section 5 is dedicated to the implementation details of these systems and their performance evaluation. We discuss possible future directions for the field in Section 6 and provide concluding remarks in Section 7.

2. Defining ontology-based information extraction

We have attempted to arrive at a definition for an OBIE system by identifying their key characteristics discussed in the literature, 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 IE, which is generally seen as a subfield of 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). Systems that use images, diagrams or videos as input cannot thus be categorized as OBIE systems.

  Categorizing systems that extract information from PDF documents is more problematic. For instance, Oro and Ruffolo have developed a system that processes PDF documents and presents the output in the form of an ontology [7]. On the first glance, this looks like a typical OBIE system. However, this system makes extensive use of the spatial relationships and images of the PDF documents. Oro and Ruffolo also recognize that their system is significantly different from 'NLP-oriented' systems. They state that systems that extract information from unstructured text can be categorized into two groups as NLP-oriented and PDF-oriented [7]. In our view, only the systems of the former category can be classified as OBIE systems (if they satisfy the other requirements of OBIE systems). While systems belonging to the latter category are important and interesting because of the widespread use of the PDF format, they should be seen as a separate type of systems.

- *Present the output using ontologies:* Li and Bontcheva [8] identify the use of a formal ontology as one of the system inputs and 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 IE process itself instead of treating it as an input (e.g. the Kylin system [9]). Since constructing an ontology in this manner should not disqualify a system from being an OBIE system, we believe that 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 IE process guided by an ontology:* we believe that 'guide' is a suitable word to describe the interaction between the ontology and the IE process in an OBIE system. In all OBIE systems, the IE process is *guided* by the ontology to extract things such as classes, properties and instances. This means that no new IE method is invented but an existing one is oriented to identify the components of an ontology.

  An important question here is whether the 'information extractors', which are the components of an IE system that extract different ontological concepts, should be considered a part of the ontology or not. (The terms 'extractor' and 'information extractor' have been used to carry this meaning [9, 10].) To the best of our knowledge, this is an open question. When we say that the IE process of an OBIE system is guided by an ontology, we consciously accommodate both possibilities: the information extractors may be either part of an ontology or may lie outside it.

  Several authors have argued that information extractors should be considered a part of an ontology when *linguistic rules* are used as the IE technique [11–14]. This technique basically relies on regular expressions that indicate the presence of ontological concepts in the text and is described in detail in Section 4.1.1. The authors of OBIE systems that use other IE techniques, such as *classification* and *web-based search*, generally ignore this question.

  We see two problems with including linguistic rules in an ontology. Firstly, they are *known* to contain errors (because they are never 100% accurate), and objections can be raised on their inclusion in ontologies in terms of formality and accuracy. Secondly, it is hard to argue that linguistic extraction rules should be considered a part of an ontology while information extractors based on other IE techniques (such as classifiers used to identify instances of a class when classification is used as the IE technique) should be kept out of it: all IE techniques perform the same task with comparable effectiveness (generally successful but not 100% accurate). But the techniques advocated for the inclusion of linguistic rules in ontologies (such as *extraction ontologies* presented by Embley [11]) cannot accommodate such IE techniques. Because of these two concerns, our view is that it is better to think of information extractors as lying outside the ontology. Further, based on the second concern raised above, we assert that either all information extractors (that use different IE techniques) should be included in the ontologies or none should be included.

A related issue is the use of the term 'ontology-driven information extraction', which has been used in several publications [13, 15, 16]. In most cases, this can be seen as a synonym for OBIE, which has emerged due to the lack of a standard terminology. We use the term ontology-based information extraction since it appears to be the term used by a majority of publications. However, Yildiz and Miksch make a distinction between these two terms [13]. 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 too is based on the view that linguistic rules should be considered a part of an ontology. As mentioned earlier, we do not subscribe to this view and as such we do not agree with the proposed distinction between two types of systems as ontology-based and ontology-driven.

Combining these factors with the definitions of information extraction presented by Russell and Norvig [1] and Riloff [2], we provide the following 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 [5, 17] 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 IE, it can be seen as an important step in the OBIE process. Further, ontology construction itself actually extracts some information because it identifies the concepts and relationships of the domain in concern. In fact, it can be seen that this follows the paradigm of 'open information extraction', which advocates the automatic discovery of relations of interest from text instead of using relations that are provided in advance [18]. Hence, it makes sense to categorize these systems under OBIE as well.

However, it can be seen that most OBIE systems only extract instances and property values with respect to classes and properties of a given ontology. This task is often known as *ontology population* [19, 20]. A more restrictive and formal definition can be provided for such systems. We have used such a definition in one of our works on OBIE [21]. The general idea behind this definition is describing an OBIE system that only performs ontology population as a set of information extractors, each extracting individuals for a class or property values for a property. The definition we have presented here encompasses these systems as well as those that construct ontologies as mentioned earlier. Hence the definition presented above should be construed as the definition for a generic OBIE system.

The fact that the output of OBIE systems are represented using ontologies makes them useful in realizing the vision of the Semantic Web. As described by Berners-Lee et al. [22], 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 are widely used to 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 and can therefore directly process the output of OBIE systems. This gives rise to one of the major potentials of OBIE described below.

Although OBIE is a relatively new field of study, it is generally agreed that it has a lot of potential [9, 17, 23, 24]. 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 [25]. OBIE systems, as well as general IE systems, are necessary to process this information automatically. This is essential because manually processing them 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. Popov et al. [26] state that it is hard to imagine that such contents would be manually created given the size of the web and assert that the *automatic metadata generation* would be the snowball to unleash an avalanche of metadata through the web, making the Semantic Web come true. Ontology-based information extraction provides such an automatic mechanism to generate semantic contents (called 'metadata' by Popov et al.) by converting the information contained in existing web pages into ontologies. This has been pointed out by several authors including Wu and Weld [9] and Cimiano et al. [23]. This process is also known as the *semantic annotation* of web pages [23, 26, 27].

3. *Improving the quality of ontologies:* as pointed out by Kietz et al. [17] and Maynard et al. [24] among others, OBIE 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.

3. Common architectures and general functionality

Although 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 architecture. It represents the union of different components found in different OBIE systems. As such, many 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 4.2) 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. They should not, however, be recognized as parts of an OBIE system.

As represented by Figure 1, the textual input of an OBIE system first goes through a preprocessor component, which converts the text to a format that can be handled by the IE 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 the actual extraction takes places. This can be implemented using several techniques as described in Section 4.1. No matter what technique is used, it is guided by an ontology. A semantic lexicon for the language is often used to support this purpose. For example, the WordNet [28] 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 [29]. 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 the Web Ontology Language (OWL) [30]. 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 [31].)

As mentioned earlier, the OBIE system is 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 [32] 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 may also include a reasoning component. The nature

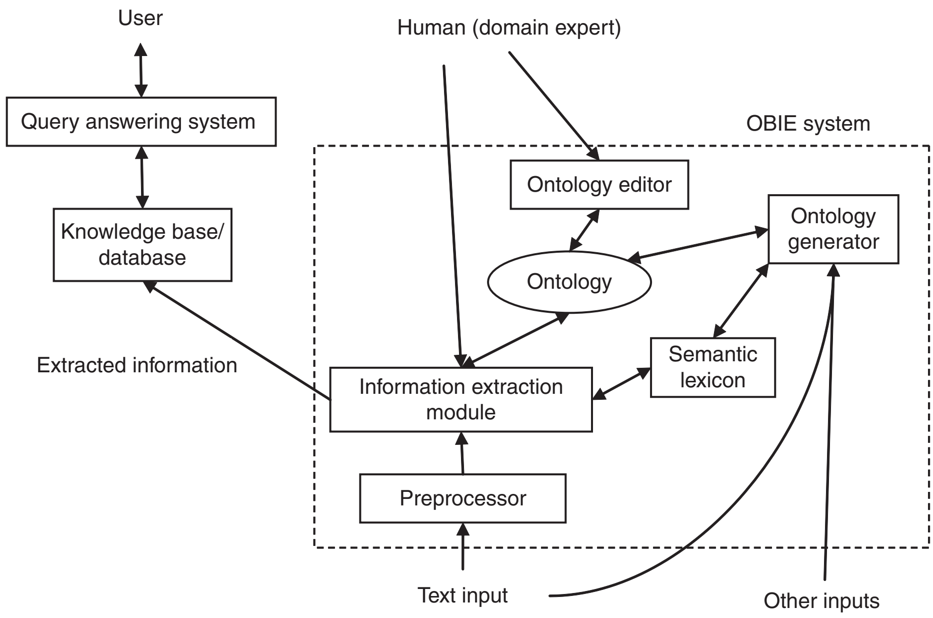

Fig. 1. General architecture of an OBIE system.

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. [33] operates as 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 IE module uses linguistic rules and gazetteer lists (described in Section 4.1). 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 system [9], the input consists of a set of Wikipedia pages. Kylin can be considered an OBIE system because it extracts information with respect to the structures of 'infoboxes' of Wikipedia, which are organized into an ontology [16]. (An infobox presents a summary of the content of a page in Wikipedia.) The selected files of Wikipedia go through a preprocessor before being used by the IE module. In this case, the IE module employs classification as described in Section 4.1.3. The ontology is constructed by a special component, a named 'Kylin Ontology Generator'. This ontology generator makes use of the structures of Wikipedia infoboxes and WordNet, which is a semantic lexicon for the English language as mentioned earlier. The information extracted by Kylin is not expected to be directly used in a knowledge base or a database but is used in developing a 'communal correction' system for Wikipedia, which allows users to verify and correct the extractions of Kylin [10]. Note that this can also be viewed as an instance where humans interact with the IE module as allowed by the architecture.

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

4. Classification of current OBIE systems

In this section, we provide a classification of current OBIE systems along a set of dimensions that we have identified in order to obtain a better understanding of their operation.

4.1. *Information extraction method*

Over the years, several types of IE techniques have been developed. Moens has presented a comprehensive categorization and analysis of these techniques in the form of a textbook [34]. Most of 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 IE methods employed by the OBIE systems we have studied.

4.1.1. **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 IE system [35], implemented in 1993, appears to be one of the earliest systems to use this method. The General Architecture for Text Engineering (GATE) [36], which is a widely used NLP framework, provides an easy-to-use platform to employ this technique.

Embley's OBIE systems [11] appear to be some 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 [13]. Such regular expressions are also used by the Textpresso system for biological literature [37], which is more of an IR 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 [5]. In addition, the OBIE systems that use the GATE architecture, such as KIM [27] and the implementation by Saggion et al. [33] 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. [12] define a concrete ontology as the combination of an abstract ontology and the lexicon for that abstract ontology. Making a similar argument, Buitelaar et al. [14] state that ontologies should be linguistically grounded. They concentrate on the words that identify the ontological concepts and the morphology of such words but the general idea is the same. As detailed in Section 2, we do not subscribe to this view. Our view is that it is better to keep information extractors outside the ontology.

The above-mentioned OBIE systems use manually identified linguistic rules. This means that a person or a group of persons have to read all the documents of the corpus and figure out suitable extraction rules. It can be seen that this a tedious and time consuming exercise which does not scale well. In order to address this issue, some systems aim to automatically mine extraction rules from text. Vargas-Vera et al. [19] have designed and implemented an OBIE system that operates on these principles back in 2000. They have used a dictionary induction tool named Crystal [38] to identify extraction rules. This tool operates on the principles of the inductive learning algorithm [39] and searches for the *most specific generalization* that covers all positive instances. The positive instances (words in the text identifying instances and property values of the ontology) are specified by a human using a mark-up tool. A sentence and phrase analysis tool named Marmot [40] is also used in this process.

A different technique for mining linguistic extraction rules has been used by Romano et al. [41] in their IE system that extracts information from medical narrative reports (physicians' notes). This approach uses an algorithm for the longest common subsequence problem [42]. It first finds the set of sentences that contain a particular type of information and treats these as sequences of words. Then it discovers the longest common subsequence of words for these sentences using the algorithm mentioned above. Finally, it discovers 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. Although this approach has been used by a generic IE system (the system developed by Romano et al. [41] is not an OBIE system), it can be adopted by OBIE systems as well.

4.1.2. Gazetteer lists

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

This technique is used by several OBIE systems. These systems often use gazetteer lists containing all the instances of some classes of the ontology. They have been used in the SOBA system [43] to get details about football games and in the implementation by Saggion et al. [33] to obtain details about countries and regions.

It is clear that one has to be careful in using gazetteer lists 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 list 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, we believe that the following conditions need to be satisfied:

1. Specify exactly what is being identified by the gazetteer.
2. Specify where the information for the gazetteer lists was 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 (www.usa.gov/Agencies/Federal/All_Agencies/index.shtml).

4.1.3. Classification techniques

Different classification techniques such as support vector machines (SVM), maximum entropy models and decision trees have been used in IE. Moens provides a comprehensive review of these techniques and categorizes them as ‘supervised classification’ techniques [34]. Following Moens, we also consider sequence tagging techniques such as Hidden Markov Models (HMM) and Conditional Random Fields (CRF) as falling under this category.

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 IE task into a set of binary classification tasks. For example, the IE system implemented by Li et al. [44], which uses uneven margins SVM and perceptron techniques, uses one binary classifier to decide whether a word token is the 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 instances and property values. The Kylin [9] OBIE system employs two classification techniques. It uses the maximum entropy model to predict which attribute values are present in a sentence and the CRF model to identify attributes within a sentence. The implementation by Li and Bontcheva [8] uses the Hieron large margin algorithm for hierarchical classification [45] to identify instances of an ontology.

4.1.4. 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 IE 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 [46]. 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 one implemented by Maedche et al. [12]. This system 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 as 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 [47], which uses the SMES system to construct an ontology, is another OBIE system that adopts this approach.

The Vulcain OBIE system developed by Todirascu et al. [48] also makes use of partial parse trees. It uses the partial syntactic structures provided by the Lexical Tree Adjoining Grammars (LTAG) parser developed by Lopez [49]. In addition, this system uses 'lexical entries' for concepts of the ontology which can be categorized as linguistic extraction rules. Therefore, the IE technique used by the Vulcain system can be seen as a combination of linguistic extraction rules and partial parse trees.

4.1.5. 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 for individual records or instances. 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 [43]. This system uses a corpus of web pages about soccer games as its source.

4.1.6. Web-based search

Using queries on web-based search engines for IE appears to be a new technique. (It has not been recognized as an IE technique even in the review compiled by Moens in 2003 [34].) The general idea behind this approach is using the web as a big corpus.

Cimiano et al. [23] have implemented an OBIE system, named Pattern-based Annotation through Knowledge on the Web (PANKOW), which semantically annotates a given web page using web-based searches only. It conducts 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>" [50]. The concept labels for the proper nouns are determined based on the aggregate number of hits recorded for each concept. The C-PANKOW system [51] 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 from a given document [15].

In addition, Wu et al. [16] have used search engine results to improve their Kylin system by adding more training examples for their classifiers. 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 IE technique. It was mentioned earlier that the implementation by Vargas-Vera et al. [19] uses linguistic rules as well as partial parse trees. The same behaviour can be observed in several other systems such as Kylin [16], which uses classification and web-based search.

4.2. Ontology construction and update

Ontology-based information extraction systems can be classified based on the manner in which they acquire the ontology to be used for IE. 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 [43], KIM [27] the implementation by Li and Bontcheva [8], the implementation by Saggion et al. [33] and PANKOW [23].

The other approach is to construct an ontology as a part of the OBIE process. As mentioned in Section 2, this approach can be seen as following the paradigm of open information extraction [18]. Ontology construction can be carried out by building an ontology from scratch or by using an existing ontology as the base. Some OBIE systems only construct an ontology and do not extract instances. Such systems include Text-To-Onto [47] and the implementation by Hwang [5]. Kylin (through Kylin ontology generator [52]) and the implementation by Maedche et al. [12] 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 IE process. (Identifying instances and their property values are not considered updates to the ontology here.) Such updates can be conducted for both cases mentioned above. However, only a few systems update the ontology in this manner. Such systems include the implementations by Maedche et al. [12] and Dung and Kameyama [53].

4.3. Components of the ontology extracted

An ontology consists of several components such as classes, data type properties, object properties (including taxonomical relationships), instances (objects), property values of the instances and constraints. The OWL specification defines the types of components supported by OWL, which is generally regarded as the standard language for specifying ontologies [30].

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 [5], extracts class names and the taxonomy (class hierarchy) only. In contrast, Text-To-Onto [47] discovers class names and taxonomical, 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 [16] 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. [12] also extracts all these components.

Many OBIE systems that concentrate on instances extract instance identifiers (names) only. Such systems include the implementation by Li and Bontcheva [8], PANKOW [23] and OntoSyphon [15]. Some systems extract property values of the instances as well. Such systems include SOBA [43], the implementation by Embley [11] and the implementation by Saggion et al. [33]. It is difficult to determine whether they extract the values for both data type properties and object properties or for data type properties only.

When extracting instances and property values for classes and properties of an ontology, it has to be decided which classes and properties to target. Since extracting information related to all the classes and properties of an ontology is often too big of a task, most systems only make extractions with respect to some classes and properties of the ontology. Structured templates can be used to specify exactly what classes and properties are targeted in this manner. This approach is adopted by the iDocument OBIE system [54], which uses queries in the SPARQL RDF query language [55], to specify such ‘extraction templates’. The use of templates in this manner to specify what to extract is somewhat similar to the approach adopted by Message Understanding conferences (MUCs). These conferences used templates consisting of empty ‘slots’ to be filled by the IE systems.

4.4. 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. [12], the implementation by Embley [11] and the implementation by Saggion et al. [33].

Table 1
Summary of the classification of OBIE systems

System	Information extraction method(s)	Ontology construction and update[1]	Components of the ontology extracted	Types of sources
Kylin [16]	Classification, web-based search	Constructed by process; not updated.	Classes, taxonomy, datatype properties, instances, property values	Wikipedia pages
PANKOW [23]	Web-based search	Off-the-shelf; not updated	Instances	No restriction
OntoSyphon [15]	Web-based search	Off-the-shelf; not updated	Instances	No restriction
Maedche et al. [12]	Partial parse trees	Constructed by process; updated.	Classes, taxonomy, datatype properties, instances, property values	Documents from a domain
Text-To-Onto [47]	Partial parse trees	Constructed by process; N/A	Classes, taxonomy, other relationships	Documents from a domain
SOBA [43]	Linguistic rules, Gazetteer lists, Analyzing tags	Off-the-shelf; not updated	Instances, property values	html files from a domain
Embley [11]	Linguistic rules	Manually defined; not updated	Instances, property values	Documents from a domain
Saggion et al. [33]	Linguistic rules, gazetteer lists	Manually defined; not updated	Instances, property values	Documents from a domain
Li and Bontcheva [8]	Classification	Off-the-shelf; not updated	Instances	Documents from a domain
Hwang [5]	Linguistic rules	Constructed by process; N/A	Classes, taxonomy, properties	Documents from a domain
ontoX [13]	Linguistic rules	Manually defined; not updated	Instances, datatype property values	Documents from a domain
Vulcain [48]	Partial parse trees, linguistic rules	Manually defined; not updated	Instances, property values	A set of emails related to a particular domain
Vargas-Vera et al. [19]	Linguistic rules	Manually defined; not updated	Instances, property values	Web pages from a particular site
KIM [26]	Linguistic rules, gazetteer lists	Manually defined; not updated	Instances, property values	Documents from a domain
iDocument [54]	Linguistic rules, gazetteer lists	Off-the-shelf; not updated	Instances, property values	Documents from a domain

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

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 IE 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. It shows how different OBIE systems can be categorized under each of the four dimensions discussed above.

5. Implementation details and performance evaluation

5.1. Tools used

One main category of tools used by OBIE systems is shallow NLP 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 POS tagging, sentence splitting and identifying occurrences of regular expressions. They are used by almost all IE 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 tools include GATE [36], sProUT [56] and those developed by the Stanford NLP Group [57] and the Center for Intelligent Information Retrieval (CIIR) at the University of Massachusetts [58]. In addition, the Saarbücker Message Extracting System (SMES) used by Maedche and his group [12, 47] 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 is utilized in different manners by OBIE systems. For example, the Kylin ontology generator uses them to assist the construction of an ontology [52]. For the English language, WordNet [28] is the most widely used semantic lexicon. Similar tools are available for some other languages such as GermaNet [59] for German and Hindi WordNet [60] 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 IE process in systems that operate in a semi-automatic manner. Protégé [29] and OntoEdit [61] 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 architecture provides a tool named OCAT (Ontology Corpus Annotation Tool) for this purpose. The iDocument OBIE system developed by Adrian et al. [62] uses such a tool to allow users to accept or reject the extractions (presented as annotation in the text) made by the system and to make new annotations. This system uses scanned images of paper documents as inputs and the annotations are made with respect to the PIMO ontology [63]. (This ontology is used by a personal information management system named Semantic Desktop [64] which Adrian et al. have integrated into their OBIE system [62]). A similar tool is also used in the implementation by Vargas-Vera et al. [19].

5.2. Text corpora

It is generally accepted that Message Understanding Conferences (MUCs) and their successor – the Automatic Content Extraction (ACE) Programme – fuelled the development in IE 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 OBIE.

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. [33] 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 [8] have created a corpus covering the topics of business, international politics and UK politics for their OBIE system; Cimiano et al. [23] have selected 30 files from a popular travel web site to create a corpus for the PANKOW system. 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 the 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 [24]. They have collected 292 news articles from three news agencies and annotated them with the concepts of the PROTON ontology [65]. In this annotation process, they have identified occurrences of the classes of the PROTON ontology. Since PROTON ontology is quite deep (up to eight levels), these annotations are complex and therefore seen as difficult for an OBIE system to recognize [24]. Hence, this corpus can be expected to evaluate the effectiveness of different OBIE systems well.

5.3. Performance measures

In IE (as well as in IR), precision and recall are the two most used metrics for performance measurement. Precision shows the number of correctly identified items as a proportion of the total number of items identified while recall shows the number of correctly identified items as a proportion of the total number of correct items available. Using {Relevant} and {Retrieved} to denote the sets of relevant and retrieved documents respectively, precision and recall are often represented using the following formulae [66, 67]. They are related to the usage of these measures in IR.

$$\text{Precision} = \frac{|\{\text{Relevant}\} \cap \{\text{Retrieved}\}|}{|\{\text{Retrieved}\}|}$$

$$\text{Recall} = \frac{|\{\text{Relevant}\} \cap \{\text{Retrieved}\}|}{|\{\text{Relevant}\}|}$$

We can also represent these measures in a generic form through the following formulae. These formulae can be used with IR as well as with IE.

$$\text{Precision} = \frac{|\text{correct answers}|}{|\text{all answers}|}$$

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

Most IE (and IR) systems face a trade-off 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 a 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.

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

Here, β denotes the weighting of precision vs recall. In most situations, 1 is used for β, giving equal weights for precision and recall. This is known as F1 score.

As pointed out by Maynard et al. [24], 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 [24]. 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. A similar approach can be adopted regarding property values.

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) [51]. They have adopted this metric from a work by Hahn et al. [68]. 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 can be 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) ∈ 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:

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

where δ(*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:

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

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

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

Maynard et al. have defined two metrics called augmented precision (AP) and augmented recall (AR) that can be used for OBIE systems [24]. These combine the concepts behind precision and recall with cost-based metrics. Experiments have shown that these measures are at least as effective as LA. In particular, AR may be a useful metric because LA is more of a measure of precision.

It should be noted that these accuracy measures are used only for the tasks of identifying instances 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.

6. Future directions

Since OBIE is a new field with a lot of potential, it can be expected to grow in different directions. Here, we try to identify several major directions. These are described at a higher level in order to encompass the various technologies that can be used in the development of the field.

1. *Improving the effectiveness of the IE process*: research work on this dimension is related to the wider field of IE, rather than being limited to OBIE. Generally speaking, this can be seen as targeting to improve precision and recall. Discovering new IE techniques and integrating existing ones would play an important role in this process. The mechanisms in which such techniques can be guided by ontologies have to be investigated in order to use them for OBIE.

In addition, some well-known problems have to be tackled in improving the effectiveness of IE and OBIE. One such problem is *reference reconciliation (object reconciliation)*, which refers to the task of determining whether two instances are one and the same [69]. Saggion et al. [33] illustrate the importance of this problem to OBIE using an example of a company director and a person accused of criminal conduct having the same name. This issue has been extensively studied by several works including the work by Dong et al. [69].

The KIM semantic annotation platform [27], which can be seen as an OBIE system, has attempted to address this problem by assigning URIs to all the entities of the knowledge base which stores the extracted information. The knowledge base is pre-populated with about 80,000 entities and each new extraction is either matched to an existing entity URI or given a new one. The basis of matching entities is not described in detail but it appears to be derived from string matching. While this approach has worked well for KIM, the extensive pre-population of the knowledge base and the nature of the ontology (the KIMO ontology used by KIM concentrates on geographical locations and companies which tend to have standard names) appear to have eased its task. More complex techniques may be needed by other OBIE systems.

2. *Integrating OBIE systems with the Semantic Web*: 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. Several options exist for this task including the definition of web services that answer ontology-based queries. Moreover, a decision has to be made on where to place the OBIE systems and Semantic Web interfaces. They can be provided for each web site or they can be implemented independent of individual sites, probably generating semantic contents for a particular domain. While these tasks are not actually a functionality of an OBIE system, they will nevertheless have a strong impact on their design and implementation.
3. *Improving the use of ontologies*: since the ontologies are used to guide the IE process and present the results in OBIE, having 'good' ontologies is essential to the success of an OBIE system. Systems that construct ontologies in an automatic or semi-automatic manner will play an important role in this process. Further, OBIE can be used to evaluate the quality of ontologies. It can also be expected that OBIE systems that automatically refine ontologies through the IE process will be more widely used in the future.

It is also interesting to note that most OBIE systems use a *single* ontology. However, there is no rule that forbids a system from using multiple ontologies. There have been several works on the use of related but distinct ontologies [70, 71]. We have explored the theoretical basis for using multiple ontologies in IE and have presented the details of two case studies on the use of multiple ontologies in OBIE in a recent publication [21]. More analyses and experiments are needed to explore this area thoroughly.

7. Conclusion

In this paper, we have reviewed the new field of OBIE and a number of systems that are categorized under it. Among other things, we have provided a definition for the field, identified a common architecture for the systems and classified the existing systems along different dimensions. We believe that these will be useful for future research work in this area.

References

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

[2] E. Riloff, Information extraction as a stepping stone toward story understanding. In: A. Ram and K. Moorman (eds), *Understanding Language Understanding: Computational Models of Reading* (MIT Press, Cambridge, MA, 1999).

[3] T.R. Gruber, A translation approach to portable ontology specifications, *Knowledge Acquisition* 5(2) (1993) 199–220.

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

[5] C. Hwang, Incompletely and imprecisely speaking: using dynamic ontologies for representing and retrieving information. In: E. Franconi and M. Kifer (eds), *Proceedings of the 6th International Workshop on Knowledge Representation Meets Databases* (ACM, New York, 1999).

[6] B. Adrian, G. Neumann, A. Troussov and B. Popov, *Proceedings of the First International and KI-08 Workshop on Ontology-Based Information Extraction Systems* (DFKI, Kaiserslautern, Germany, 2008).

[7] E. Oro and M. Ruffolo, Towards a system for ontology-based information extraction from PDF documents. In: R. Meersman and Z. Tari (eds), *Proceedings of the OTM 2008 Confederated International Conferences, CoopIS, DOA, GADA, IS, and ODBASE 2008. Part II on On the Move to Meaningful Internet Systems* (Springer-Verlag, Berlin, 2008).

[8] Y. Li and K. Bontcheva, Hierarchical, perceptron-like learning for ontology-based information extraction. In: *Proceedings of the 16th International Conference on World Wide Web* (ACM, New York, 2007).

[9] F. Wu and D.S. Weld, Autonomously semantifying wikipedia. In: *Proceedings of the Sixteenth ACM Conference on Information and Knowledge Management* (ACM, New York, 2007).

[10] D.S. Weld, R. Hoffmann and F. Wu, Using Wikipedia to bootstrap open information extraction, *SIGMOD Record* 37(4) (2008) 62–68.

[11] D.W. Embley, Toward semantic understanding: an approach based on information extraction ontologies. In: *Proceedings of the 15th Australasian Database Conference* (Australian Computer Society, Darlinghurst, Australia, 2004).

[12] A. Maedche, G. Neumann and S. Staab, Bootstrapping an ontology-based information extraction system. In: P.S. Szczepaniak, J. Segovia, J. Kacprzyk and L.A. Zadeh (eds), *Intelligent Exploration of the Web*, (Physica-Verlag GmbH, Heidelberg, Germany, 2003).

[13] B. Yildiz and S. Miksch, OntoX – a method for ontology-driven information extraction. In: *Proceedings of the 2007 International Conference on Computational Science and its Applications* (Springer, Berlin, 2007).

[14] P. Buitelaar, P. Cimiano, P. Haase and M. Sintek, Towards linguistically grounded ontologies. In: *Proceedings of the 6th European Semantic Web Conference on the Semantic Web: Research and Applications* (Springer-Verlag, Berlin, 2009).

[15] L. McDowell and M.J. Cafarella, Ontology-driven information extraction with OntoSyphon. In: *Proceedings of the 5th International Semantic Web Conference* (Springer, Berlin, 2006).

[16] F. Wu, R. Hoffmann and D.S. Weld, Information extraction from Wikipedia: moving down the long tail. In: *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (ACM, New York, 2008).

[17] J. Kietz, A. Maedche and R. Volz, A method for semi-automatic ontology acquisition from a corporate intranet. In: *Proceedings of the EKAW'00 Workshop on Ontologies and Text* (Springer, Berlin, 2000).

[18] M. Banko, M.J. Cafarella, S. Soderland, M. Broadhead and O. Etzioni, Open information extraction from the web. In: *Proceedings of the 20th International Joint Conference on Artificial Intelligence* (AAAI Press, Menlo Park, CA, 2007).

[19] M. Vargas-Vera, E. Motta, J. Domingue, S.B. Shum and M. Lanzoni, Knowledge extraction by using an ontology-based annotation tool. In: *Proceedings of the Workshop on Knowledge Markup and Semantic Annotation* (ACM, New York, 2001).

[20] T. Declerck, C. Federmann, B. Kiefer and H-U. Krieger, Ontology-based information extraction and reasoning for business intelligence applications. In: *Proceedings of the 31st Annual German Conference on Advances in Artificial Intelligence (Demos)* (Springer-Verlag, Berlin, 2008).

[21] D.C. Wimalasuriya and D. Dou, Using multiple ontologies in information extraction. In: *Proceedings of the 18th ACM Conference on Information and Knowledge Management* (ACM, New York, 2009).

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

[23] P. Cimiano, S. Handschuh and S. Staab, Towards the self-annotating web. In: *Proceedings of the 13th International Conference on World Wide Web* (ACM, New York, 2004).

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

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

[26] B. Popov, A. Kiryakov, D. Ognyanoff, D. Manov and A. Kirilov, KIM – a semantic platform for information extraction and retrieval, *Journal of Natural Language Engineering* 10(3–4) (2004) 375–392.

[27] B. Popov, A. Kiryakov, A. Kirilov, D. Manov, D. Ognyanoff and M. Goranov, KIM – semantic annotation platform. In: *Proceedings of the 2nd International Semantic Web Conference* (Springer-Verlag, Berlin, 2003).

[28] G.A. Miller and C. Fellbaum, *WordNet: A Lexical Database for the English Language (2006)*. Available at: http://wordnet.princeton.edu (accessed 25 June 2009).

[29] M. Musen, N. Noy, M. O'Connor, T. Redmond, D. Rubin, S. Tu, T. Tudorache and J. Vendetti, *Protégé Ontology Editor and Knowledge Acquisition System (2005)*. Available at: http://protege.stanford.edu (accessed 25 June 2009).

[30] M. Dean, G. Schreiber, S. Bechhofer, F.V. Harmelen, J. Hendler, I. Horrocks, D.L. McGuinness, P.F. Patel-Schneider and L.A. Stein, *OWL Web Ontology Language Reference (2004)*. Available at: www.w3.org/TR/owl-ref (accessed 25 June 2009).

[31] T. Berners-Lee, *Cleaning up the User Interface (1997)*. Available at: www.w3.org/DesignIssues/UI.html (accessed 25 June 2009).

[32] 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: *Proceedings of the 33rd International Conference on Very Large Databases* (ACM, New York, 2007).

[33] H. Saggion, A. Funk, D. Maynard and K. Bontcheva, Ontology-based information extraction for business intelligence. In: *Proceedings of the 6th International and 2nd Asian Semantic Web Conference* (Springer, Berlin, 2007).

[34] M. Moens, *Information Extraction: Algorithms and Prospects in a Retrieval Context (The Information Retrieval Series)* (Springer-Verlag, Secaucus, NJ, 2003).

[35] 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: Ruzena Bajcsy (ed.), *Proceedings of the 13th International Joint Conference on Artificial Intelligence* (Morgan Kaufmann, Chambéry, France, 1993).

[36] H. Cunningham, K. Bontcheva, V. Tablan and D. Maynard, *General Architecture for Text Engineering (GATE) (2003)*. Available at: www.gate.ac.uk (accessed 25 June 2009).

[37] 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) (2004) 1984–1998.

[38] S. Soderland, D. Fisher, J. Aseltine and W.G. Lehnert, CRYSTAL: inducing a conceptual dictionary. In: *Proceedings of the 14th International Joint Conference on Artificial Intelligence* (Morgan Kaufmann, Montreal, Canada, 1995).

[39] T.M. Mitchell, Generalization as search, *Artificial Intelligence*, 18(2) (1982) 203–226.

[40] W.B. Croft, Lab report special section: the University of Massachusetts Center for Intelligent Information Retrieval, *SIGIR Forum* 29(1) (1995) 1–7.

[41] R. Romano, L. Rokach and O. Maimon, Automatic discovery of regular expression patterns representing negated findings in medical narrative reports. In: *Proceedings of the 6th International Workshop on Next Generation Information Technologies and Systems* (Springer, Berlin, 2006).

[42] E.W. Myers, An O(ND) difference algorithm and its variations, *Algorithmica* 1(2) (1986) 251–266.

[43] P. Buitelaar and M. Siegel, Ontology-based information extraction with SOBA. In: *Proceedings of the Fifth International Conference on Language Resources and Evaluation* (European Language Resources Association, Genoa, Italy, 2006).

[44] Y. Li, K. Bontcheva and H. Cunningham, Using uneven margins SVM and perceptron for information extraction. In: *Proceedings of the 9th Conference on Computational Natural Language Learning* (Association for Computational Linguistics, Morristown, NJ, 2005).

[45] O. Dekel, J. Keshet and Y. Singer, Large margin hierarchical classification. In: *Proceedings of the Twenty First International Conference on Machine Learning* (ACM, New York, 2004).

[46] 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* (Association for Computational Linguistics, Morristown, NJ, 1998).

[47] A. Maedche and S. Staab, The Text-To-Onto Ontology Learning Environment. In: *Software Demonstration at the Eighth International Conference on Conceptual Structures* (Springer-Verlag, Berlin, 2000).

[48] A. Todirascu, L. Romary and D. Bekhouche, Vulcain – an ontology-based information extraction system. In: *Proceedings of the 6th International Conference on Applications of Natural Language to Information Systems-Revised Papers* (Springer-Verlag, London, 2002).

[49] P. Lopez, *Robust Parsing with Lexicalized Tree Adjoining Grammars* (PhD Thesis, INRIA, Nancy, France, 1999).

[50] M.A. Hearst, Automatic acquisition of hyponyms from large text corpora. In: *Proceedings of the 14th Conference on Computational Linguistics* (ACM, New York, 1992).

[51] P. Cimiano, G. Ladwig and S. Staab, Gimme' the context: context-driven automatic semantic annotation with C-PANKOW. In: *Proceedings of the 14th International Conference on World Wide Web* (ACM, New York, 2005).

[52] F. Wu and D.S. Weld, Automatically refining the Wikipedia infobox ontology. In: *Proceedings of the 17th International Conference on World Wide Web* (ACM, New York, 2008).

[53] T.Q. Dung and W. Kameyama, Ontology-based information extraction and information retrieval in health care domain. In: *Proceedings of the 9th International Conference on Data Warehousing and Knowledge Discovery* (Springer, Berlin, 2007).

[54] B. Adrian, J. Hees, L. van Elst and A. Dengel, iDocument: using ontologies for extracting and annotating information from unstructured text. In: *Proceedings of the 32nd Annual German Conference on AI* (Springer-Verlag, Heidelberg, 2009).

[55] E. Prud'hommeaux and A. Seaborne, *SPARQL Query Language for RDF (2008)*. Available at: www.w3.org/TR/rdf-sparql-query/ (accessed 25 June 2009).

[56] W. Drozdzynski, M. Becker, H.-U. Krieger, J. Piskorski, U. Schäfer and F. Xu, *SProUT (Shallow Processing with Unification and Typed Feature Structures) (2002)*. Available at: http://sprout.dfki.de (accessed 25 June 2009).

[57] C. Manning and D. Jurafsky, *The Stanford Natural Language Processing Group (1999)*. Available at: http://nlp.stanford.edu/index.shtml (accessed 25 June 2009).

[58] W.B. Croft, J, Allen, A. McCallum, R. Manmatha and D.A. Smith, *The Center for Intelligent Information Retrieval (CIIR) (1992)*. Available at: http://ciir.cs.umass.edu/ (accessed 25 June 2009).

[59] E. Hinrichs, P. Gupta, L. Lemnitzer, R. Barkey, C. Frey, M. Hinrichs and C. Kunze, *GermaNet – The German Wordnet (1997)*. Available at: www.sfs.uni-tuebingen.de/GermaNet/ (accessed 25 June 2009).

[60] P. Bhattacharyya and P. Pande, *Hindi WordNet: A Lexical Databse for Hindi (2001)*. Available at: www.cfilt.iitb.ac.in/wordnet/webhwn/ (accessed 25 June 2009).

[61] D. Fensel and F. van Harmelen, *OntoEdit (2002)*. Available at: www.ontoknowledge.org/about.shtml (accessed 25 June 2009).

[62] B. Adrian, H. Maus, M. Kiesel and A. Dengel, Towards ontology-based information extraction and annotation of paper documents for personalized knowledge acquisition. In: *Proceedings of the First International Workshop on Personal Knowledge Management* (Bonner Köllen Verlag, Solothurn, Switzerland, 2009).

[63] L. Sauermann, L. van Elst and A. Dengel, PIMO – a framework for representing personal information models. In: *Proceedings of the International Conferences on New Media Technology and Semantic Systems* (JUCS, Graz, Austria, 2007).

[64] L. Sauermann, A. Bernardi and A. Dengel, Overview and outlook on the semantic desktop. In: *Proceedings of the 1st Workshop on the Semantic Desktop at the ISWC 2005 Conference* (CEUR-WS, Galway, Ireland, 2005).

[65] *PROTON Ontology (2005)*. Available at: http://proton.semanticweb.org/ (accessed 25 June 2009).

[66] J. Han and M. Kamber, *Data Mining: Concepts and Techniques*, 2nd edition (Morgan Kaufmann, San Francisco, CA, 2006) 616–617.

[67] *Precision and Recall (2009)*. Available at: http://en.wikipedia.org/wiki/Precision_and_recall (accessed 25 June 2009).

[68] U. Hahn and K. Schnattinger, Towards text knowledge engineering. In: *Proceedings of the Fifteenth National/Tenth Conference on Artificial Intelligence/Innovative Applications of Artificial Intelligence* (American Association for Artificial Intelligence, Menlo Park, CA, 1998).

[69] X. Dong, A.Y. Halevy and J. Madhavan, Reference reconciliation in complex information spaces. In: *Proceedings of the 2005 ACM SIGMOD International Conference on Management of Data* (ACM, New York, 2005).

[70] D. Dou, D.V. McDermott and P. Qi, Ontology translation on the semantic web, *Journal of Data Semantics* 2(1) (2005) 35–57.

[71] B.C. Grau, B. Parsia and E. Sirin, Working with multiple ontologies on the semantic web. In: *Proceedings of the 3rd International Semantic Web Conference* (Springer, Berlin, 2004).