CRCTOL: A Semantic-Based Domain Ontology Learning System

Xing Jiang and Ah-Hwee Tan
School of Computer Engineering, Nanyang Technological University, Singapore 639798.
E-mail: {jian0008, asahtan}@ntu.edu.sg

Domain ontologies play an important role in supporting knowledge-based applications in the Semantic Web. To facilitate the building of ontologies, text mining techniques have been used to perform ontology learning from texts. However, traditional systems employ shallow natural language processing techniques and focus only on concept and taxonomic relation extraction. In this paper we present a system, known as Concept-Relation-Concept Tuple-based Ontology Learning (CRCTOL), for mining ontologies automatically from domain-specific documents. Specifically, CRCTOL adopts a full text parsing technique and employs a combination of statistical and lexico-syntactic methods, including a statistical algorithm that extracts key concepts from a document collection, a word sense disambiguation algorithm that disambiguates words in the key concepts, a rule-based algorithm that extracts relations between the key concepts, and a modified generalized association rule mining algorithm that prunes unimportant relations for ontology learning. As a result, the ontologies learned by CRCTOL are more concise and contain a richer semantics in terms of the range and number of semantic relations compared with alternative systems. We present two case studies where CRCTOL is used to build a terrorism domain ontology and a sport event domain ontology. At the component level, quantitative evaluation by comparing with Text-To-Onto and its successor Text2Onto has shown that CRCTOL is able to extract concepts and semantic relations with a significantly higher level of accuracy. At the ontology level, the quality of the learned ontologies is evaluated by either employing a set of quantitative and qualitative methods including analyzing the graph structural property, comparison to WordNet, and expert rating, or directly comparing with a human-edited benchmark ontology, demonstrating the high quality of the ontologies learned.

Introduction

A domain ontology is an explicit specification of a conceptualization (Gruber, 1993), comprising a formal description of concepts, relations between concepts, and axioms on the relations in the domain of interest. As the backbone of the Semantic Web (Berners-Lee, Hendler, & Lassila, 2001), domain ontologies enable software agents to carry out sophisticated tasks for users. For instance, OntoSeek (Guarino, Masolo, & Vetere, 1991) uses ontologies to help formulate queries so as to improve the precision of the information retrieved. OntoSearch (Jiang & Tan, 2006) utilizes domain ontologies with spreading activation theory for finding more relevant documents to the queries submitted. Brunner et al. (2007) employ ontologies for better product information management. The success of the Semantic Web greatly relies on the large population of high-quality domain ontologies.

Ontology building is a tedious process. Manually acquiring knowledge for building domain ontologies requires much time and resources. To reduce the costs of building ontologies, ontology learning systems (Gomez-Perez & Manzano-Macho, 2003) have been developed to extract concepts, relations between concepts, and axioms on relations from domain-specific documents. However, in the current state of the art, the technologies for learning domain ontologies are far less developed compared with other techniques for the Semantic Web (Gomez-Perez & Manzano-Macho, 2003). Most of the domain ontology learning systems (Biébow & Szulman, 1999; Bisson, Nédellec, & Cañamero, 2000; Engels, 2003; Faure & Nédellec, 1998; Missikoff, Navigli, & Velardi, 2002) only use shallow (or light) Natural Language Processing (NLP) tools to process documents and focus on extracting concepts and taxonomic (IS-A) relations. For example, OntoLearn (Missikoff et al., 2002), an ontology learning system developed at IASI-CNR, makes use of shallow NLP tools, including a morphologic analyzer, a part-of-speech (POS) tagger, and a chunk parser to process documents and employs text-mining techniques to produce ontologies based on document collections. However, the performance of the concept extraction method is greatly affected by the size of the document collections used for ontology learning.

Text-To-Onto (Maedche & Staab, 2000), also based on shallow NLP tools, is able to extract key concepts and semantic relations (including nontaxonomic ones) from texts.

Received January 16, 2009; revised July 1, 2009; accepted August 21, 2009

© 2009 ASIS&T • Published online 13 October 2009 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/asi.21231

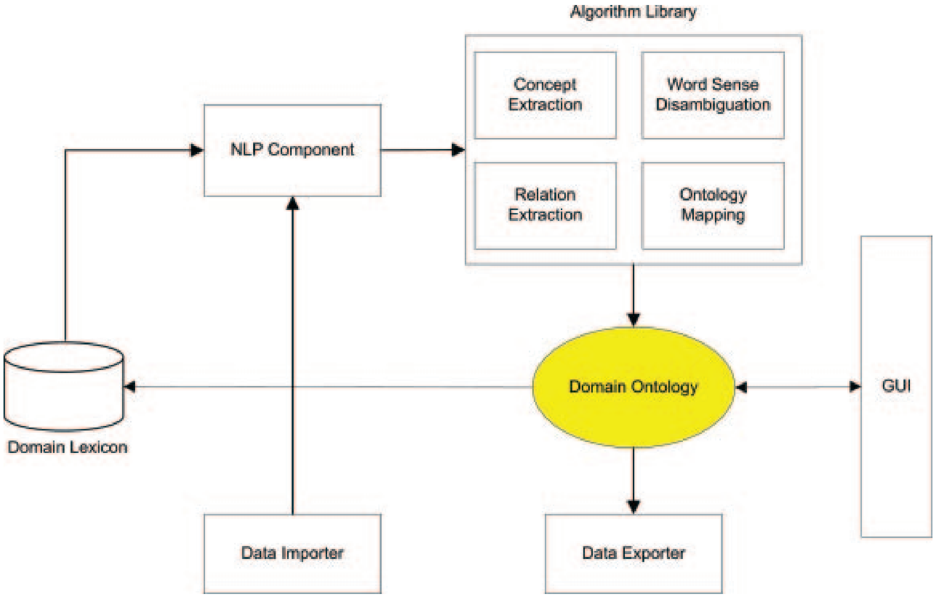


FIG. 1. The architecture of the CRCTOL system.

The selection of concepts is based on the *tf/idf* (Salton & McGill, 1986) measure used in the field of information retrieval. Semantic relations between concepts are extracted using an association rule-mining algorithm and predefined regular expressions. However, as *tf/idf* is designed primarily for document retrieval but not concept extraction, the system cannot effectively extract domain-specific concepts. Also, the identification of semantic relations is based on POS tags only, limiting the accuracy of the relations extracted. The same problems are also suffered by its successor, known as Text2onto (Cimiano & Völker, 2005).

Rajaraman and Tan (2003) extract knowledge in the form of concept frame graphs (CFGs) from text documents. Semantic relations between concepts are extracted through analyzing the POS tags of the sentences using a library of extraction rules. However, as the CFG system extracts concepts and relations from all sentences without considering their importance, it tends to extract a large number of concepts and relations, many of which have no real significance. Also, the CFG system is designed to extract nontaxonomic relations only.

In this paper we present a system, known as Concept-Relation-Concept Tuple-based Ontology Learning (CRCTOL) (Jiang & Tan, 2005), for mining ontologies from domain-specific text documents. By using a full text parsing technique and incorporating both statistical and lexico-syntactic methods, the ontologies learned by our system are more concise and contain a richer semantics in terms of the range and number of semantic relations compared with alternative systems. We conduct two case studies where CRCTOL extracts ontological knowledge, specifically key concepts and semantic relations, from a terrorism domain text collection and a sport domain text collection. At the component level, quantitative evaluation by comparing with Text-To-Onto and Text2Onto has shown that CRCTOL produces much better accuracy for both concept and relation extraction. At the ontology level, we employ a wide range of quantitative and qualitative methods, including a structural property-based method to verify the quality of the learned ontological network, comparisons to WordNet based on the taxonomic relations extracted, scoring the learned ontology's quality by the experts, and directly comparing the learned ontology with a human-edited benchmark ontology, which all demonstrate that ontologies of high quality are built.

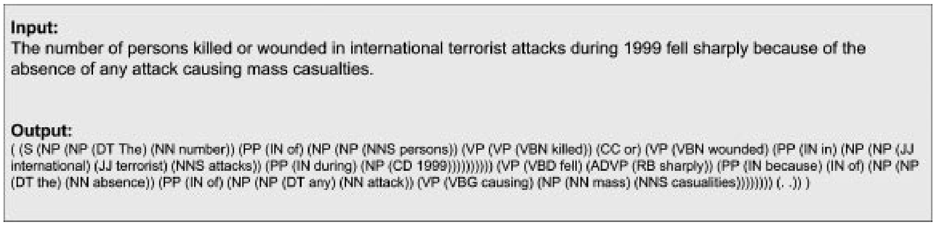
The rest of the paper is organized as follows. The next section presents the system's framework. The algorithms for concept extraction, word sense disambiguation, semantic relation extraction, and ontology mapping are described in the following four sections. In the Experiment section the two case studies are presented. Concluding remarks and future work are given in the final section.

System Architecture

The CRCTOL system (Figure 1) consists of six components: Data Importer, Natural Language Processing, Algorithm Library, Domain Lexicon, User Interface, and Data Exporter.

Data Importer: As our system only supports plain text documents, the Data Importer converts documents of other formats, such as PDF, XML, or HTML, into plain texts, where the structural information of the documents, such as DTD, is discarded.

Natural Language Processing: The NLP component incorporates a set of NLP tools, including the Stanford's Log-linear Part-Of-Speech Tagger (Toutanova, Klein, Manning, & Singer, 2003) for tagging words with POS tags and the Berkeley Parser (Petrov, Barrett, Thibaux, & Klein, 2006) for identifying the constituents in the sentences and their



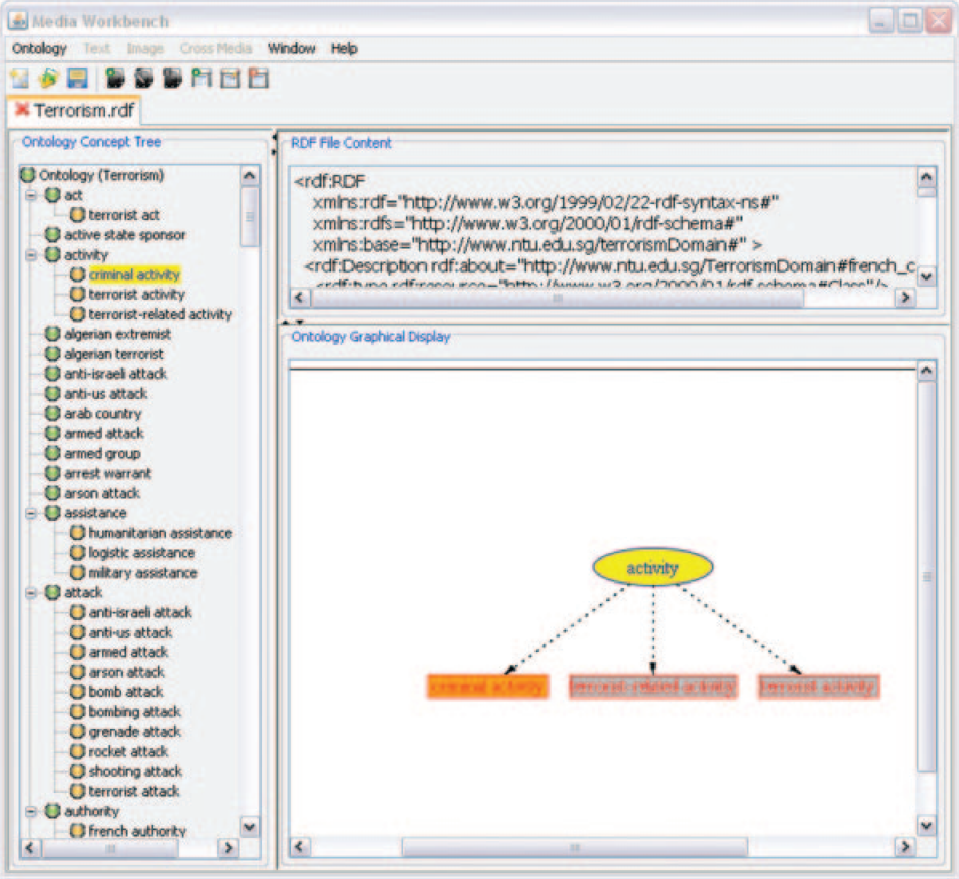
Input:

The number of persons killed or wounded in international terrorist attacks during 1999 fell sharply because of the absence of any attack causing mass casualties.

Output:

```
( (S (NP (NP (DT The) (NN number)) (PP (IN of) (NP (NP (NNS persons)) (VP (VP (VBN killed)) (CC or) (VP (VBN wounded) (PP (IN in) (NP (NP (JJ
international) (JJ terrorist) (NNS attacks)) (PP (IN during) (NP (CD 1999))))))))))) (VP (VBD fell) (ADVP (RB sharply)) (PP (IN because) (IN of) (NP (NP
(DT the) (NN absence)) (PP (IN of) (NP (NP (DT any) (NN attack)) (VP (VBG causing) (NP (NN mass) (NNS casualities)))))))) (. .)) )
```

FIG. 2. A sample input sentence and the corresponding output of the NLP component.



```
<rdf:RDF
   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
   xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
   xmlns:base="http://www.ntu.edu.sg/terrorismDomain#" >
 <rdf:Description rdf:about="http://www.ntu.edu.sg/TerrorismDomain#french_c
```

FIG. 3. The user interface.

relationships.[1] With the NLP component, we can utilize a full text parsing technique for text analysis. This function distinguishes our system from many alternative systems that only use POS tagging or shallow parsing techniques. An example of the full text parsing technique is given in Figure 2, where the POS and the syntactic tags have been assigned to the sentence.

Algorithm Library: The algorithm library consists of a statistical algorithm that extracts key concepts from a document collection, a word sense disambiguation (WSD) algorithm that disambiguates the key concepts, a rule-based algorithm that extracts relations between the key concepts, and a modified generalized association rule-mining algorithm that prunes unimportant semantic relations.

Domain Lexicon: The domain lexicon contains terms specific to the domain of interest and their attributes, which are used in the NLP component for analyzing documents. For instance, the word *bin* is usually recognized as "NN,"[2] indicating *bin* as a common noun. However, *bin* can also be treated as a proper noun ("NNP") as it is the name of "Osama Bin Ladin." The domain lexicon records such information in order to improve the accuracy of the NLP component. It is manually built and can be updated by the user during the learning process.

User Interface: After a domain ontology is built, it will be shown in the user interface (see Figure 3). The left panel

[1] In general, the Berkeley Parser can be used directly to process documents. But for special cases where the words have unusual attributes, we first need to train the Log-linear tagger to assign correct POS tags and then parse documents with the Berkeley Parser.

[2] The POS tag introduction is available at ftp://ftp.cis.upenn.edu/pub/treebank/doc/tagguide.ps.gz

DOI: 10.1002/asi

of this interface lists the concepts of the domain ontology. When a concept is selected, its surrounding information will be shown as a graph in the right panel of the interface. Therefore, the users can easily explore the internal structure of the learned ontology. Furthermore, this interface enables the users to edit the learned ontology by adding or removing concepts and relations between concepts. The concepts and relations extracted incorrectly can thus be removed or corrected by the user.

Data Exporter: In CRCTOL, the learned domain ontology is modeled as a graph, which is a compact and abstract representation of the ontology, and can be easily translated into other knowledge representation languages. The Data Exporter is used to translate the learned ontology into a particular representation language. At present, CRCTOL supports two ontology languages: RDFS ("RDF Schema Specification," 2004) and OWL ("Web Ontology Language (OWL)," 2004).

The overall procedure for ontology learning is summarized as follows.

Data Preprocessing: Documents of other formats are converted to plain text before learning ontologies.

NLP Analyzing: Input files are processed using the NLP component. POS and Syntactic tags are assigned to individual words and sentences in the documents.

Concept Extraction: Concepts are extracted and identified by a statistical algorithm from texts. These concepts are called the key concepts in the target domain.

Word Sense Disambiguation: The senses of the key concepts are identified using a variant of the LESK algorithm (Lesk, 1986).

Semantic Relation Extraction: The semantic relations of the key concepts are extracted from the text, which include taxonomic and nontaxonomic relations.

Ontology Mapping: An ontology is built in this step by mapping the concepts and relations extracted. The final ontology is stored in the form of a graph.

Ontology Exportation: The users explore the built ontology with the user interface, modify the ontology if necessary, and export the learned ontology.

Concept Extraction

To build a domain ontology, the initial step is to find the important concepts of the target domain. As terms correspond to linguistic representation of concepts in the texts (Sagar, Dungworth, & McDonald, 1980), concept extraction is thus used to extract those domain-specific terms from texts. In our system, concept extraction consists of two steps. First, possible candidate terms (i.e., a set of lexical units) are extracted from texts with certain linguistic filters,[3] i.e., term extraction. Then, domain-specific terms are identified from those candidate terms with a particular statistical measure, i.e., term selection. This module plays a key role in the ontology learning process, whose performance greatly affects the system's overall performance for building domain ontologies.

Concept Extraction Procedure

Ontology learning systems typically adopt one of the following two approaches to extract concepts. The first one, as used by Xu, Kurz, Piskorski, and Schemeier (2002), initially identifies a set of single-word terms, particularly nouns, from the texts as the seed concepts. Then, multiword terms are formed by combining these single-word terms using certain statistical measures such as the Mutual Information measure (Fano, 1961). As a result, the multiword terms may not be natural in the texts and are coined merely from the statistical aspect.

The second approach, adopted by Text-To-Onto (Maedche & Staab, 2000) and OntoLearn (Missikoff et al., 2002), employs a set of predefined linguistic filters (particularly the POS tag-based rules) to extract possible candidate terms, including single-word terms and multiword terms, from texts. Then, some statistical measures, e.g., *tf/idf*, are used to rank the extracted terms. Only terms whose values or ranks are greater than a threshold are selected as the concepts.

In CRCTOL, we follow the second approach for concept extraction. However, from prior experiments we find that most domain-specific concepts are multiword terms. The small number of relevant single-word terms can either be found appearing frequently in the multiword terms or easily inferred based on the multiword terms. But the existing relevance measures, such as the *tf/idf* measure, all prefer single-word terms. In this case, although the extracted concepts are correct, it is hard to enrich the learned ontology. For example, in the terrorism domain, if we have the concept *international terrorist group*, the concept *group* can be included automatically in the ontology if it is missing. On the other hand, it is inconceivable to add the concept *international terrorist group* into the ontology if we only have the concept *group*. Therefore, we consider a different strategy for term extraction, focusing on multiword term extraction. Single-word terms are added if they appear frequently in the multiword terms or they are found related to the multiword terms through certain semantic relations in the texts. Compared with the ontologies learned with the existing approach, our ontologies can be easily enriched.

The detailed procedure for concept extraction in CRCTOL is described below.

1. Extract all possible multiword terms from texts. As concepts are nouns or noun phrases in texts, only lexical units with the NP[4] tag are collected.

2. Remove articles and descriptive adjectives such as "a," "many," and "several" from the terms extracted.

[3] In particular, the linguistic filters are a set of POS and Syntactic tag-based rules.

[4] "NP" is the tag used in our NLP software to annotate nouns and noun phrases.

DOI: 10.1002/asi

3. Group all possible sets of two or more words from each extracted term to form candidate terms. For instance, *terrorist attack* is generated from *international terrorist attack*.

4. For each generated multiword term t, compute its domain relevance value $DRM(t)$.[5] The $DRM(t)$ score, described below, is a statistical measure for evaluating a term's relevance to the target domain. Terms with high *DRM* values are selected to form an initial concept list of the domain ontology.

5. Let V be the set of single-word terms appearing in the initial concept list as the syntactic head of a term t. For instance, *attack* is the syntactic head of the term *(NP (JJ terrorist) (NN attack))*. We compute for each single-word term in V its frequency in the initial concept list. Those with frequency above a threshold δ are added to the concept list.

Concept Extraction Measure

Review of existing relevance measures. In Text-To-Onto and its successor Text2onto, the *tf/idf* measure is used to determine the domain relevance of these extracted terms. Particularly, given an extracted term t in a document set d, the term frequency (*tf*) and inverse document frequency (*idf*) are computed as follows:

$$tf = \frac{\textit{count of term t in d}}{\textit{total number of terms in d}}, \tag{1}$$

$$idf = \log_2 \frac{\textit{the size of d}}{\textit{count of documents where term t appears}}, \tag{2}$$

$$tf/\,idf = tf \times idf \tag{3}$$

The original *tf/idf* is designed for identifying important keywords in individual documents for the purpose of information retrieval. It is, however, not suitable for identifying significant concepts of a text collection. For example, given a domain-specific concept t, it may appear in many documents in d as it is popularly used in the domain of interest, i.e., $df(t) \approx$ *the side of d*. However, t may not be selected as $idf(t) \approx 0$. In other words, *tf/idf*'s performance is sensitive to the size of the document set d. It cannot work effectively on datasets with limited number of documents, even if these documents may be very long.

To overcome the deficiency of *tf/idf*, the KFIDF measure (Xu et al., 2002) is proposed that utilizes multiple document collections of different domains for concept extraction. The measure is computed by:

$$\text{KFIDF}(w, D_i) = docs(w, D_i) \times \log\left(\frac{n \times |D|}{|D(w)|} + 1\right), \tag{4}$$

where $docs(w, D_i)$ is the number of documents of the particular domain D_i in which a term w occurs, n is a smoothing factor, $|D(w)|$ is the number of different domains in which w occurs, and $|D|$ is the total number of different domains. Words that have high KFIDF values in D_i will be selected as the concepts of D_i. However, KFIDF only considers the importance of the document frequency for concept extraction. To effectively identify and separate domain-specific terms, it requires that the total number of different domains $|D|$ should be large enough, as many terms would have the same KFIDF values with a small $|D|$.

In the OntoLearn system, two statistical measures *DR&DC* (Missikoff, Velardi, & Fabriani, 2003) are used together to identify domain-specific concepts.

Domain Relevance (DR): The domain relevance of a term t in domain D_i is given by:

$$DR(t, D_i) = \frac{p(t|D_i)}{\sum_{i=1}^{n} p(t|D_i)} \tag{5}$$

where $DR \in [0, 1]$, n is the number of document collections, and the conditional probability $p(t|D_i)$ is estimated as:

$$E(p(t|D_i)) = \frac{freq(t \in D_i)}{\sum_{i=1}^{n} freq(t \in D_i)}.$$

Domain Consensus (DC): The domain consensus of a term t in domain D_i is given by:

$$DC(t, D_i) = H(P(t, d_j)) = \sum_{d_j \in D_i} p(t, d_j) \times \log_2\left(\frac{1}{p(t, d_j)}\right) \tag{6}$$

where d_j are documents in D_i, and the probability $p(t, d_j)$ is estimated as:

$$E(p(t, d_j)) = \frac{freq(t \in d_j)}{\sum_{d_j \in D_i} freq(t \in d_j)}.$$

Terms with high *DR* values and *DC* values, ranked by a linear combination of *DR* and *DC* (i.e., $\alpha \times DR + (1 - \alpha) \times DC$, $\alpha \in [0, 1]$), are selected as domain-specific terms.

The above two statistical measures, however, suffer from the following problems. First, the *DR* measure does not consider the rare event property of concepts (Dunning, 1993). If we substitute the estimation $E(p(t|D_i))$ back, the *DR* measure can be written as:

$$DR(t, D_i) = \frac{\frac{freq(t \in D_i)}{{}_{i=1}^{n} freq(t \in D_i)}}{\sum_{j=1}^{n} \frac{freq(t \in D_j)}{{}_{i=1}^{n} freq(t \in D_i)}}. \tag{7}$$

After simplifying this formula, we see the *DR* measure is actually computed by:

$$DR(t, D_i) = \frac{freq(t \in D_i)}{\sum_{i=1}^{n} freq(t \in D_i)}. \tag{8}$$

So, in OntoLearn, the *Domain Relevance* value merely depends on the term's frequency in the target domain corpus and the contrasting corpora. If we adjust the size of the target domain corpus or the size of the contrasting corpus, the result will be greatly different.

Also, the *DC* measure is not suitable for concept extraction. Suppose a term t appears in two documents with a

[5] The DRM measure is a simple version of the TIM-DRM measure used in our previous work (Jiang & Tan, 2005).

TABLE 1. Contingency table.

	$\mathcal{A}$	$\overline{\mathcal{A}}$	Total
Frequency of term t	a	b	$a+b$
Frequency of other terms	c	d	$c+d$
Total	$a+c$	$b+d$	$a+b+c+d$

frequency of one in each document and a term s appears in the same two documents with a frequency of two in each document. Using the DC measure, $DC(t) = DC(s)$. This conclusion is not appropriate, as terms with a higher occurrence frequency should naturally be more important.

Domain relevance measure (DRM). In CRCTOL, we develop a new relevance measure known as Domain Relevance Measure (DRM) for concept extraction. The ideas behind this measure are presented as follows.

First, we incorporate syntactic information into multiword term extraction. Although this approach is more effective than that of using POS tag-based linguistic filters, it still suffers from the same problem that these extracted lexical units may not be cohesive enough to be treated as a term. In other words, they may be coined together by chance. Traditional approaches (Basili, Rossi, & Pazienza, 1997; Daille, 1996; Maedche & Staab, 2000) use statistical measures such as the Mutual Information measure and the likelihood ratio test to score these extracted unites for tackling this problem. In our system, we consider using the term frequency *tf* for this purpose, since the *tf* measure is simple but has been shown to produce better performance than other measures for multiword term extraction (Daille, 1996).

Second, we aim to find domain-specific terms. The simple approaches, such as the *DR* measure used in OntoLearn, only consider the frequency of the terms in different document sets for tackling this problem. As a result, their performances are greatly affected by the datasets used and many irrelevant concepts may be selected. To achieve our goal, we consider using the likelihood ratio test (Casella, 1990), which has been shown to be statistically reliable for this task.

Here, we consider only a two-class problem, selecting terms from a target domain $\mathcal{A}$ with a contrasting domain $\overline{\mathcal{A}}$ The contingency table of a term t's frequency in $\mathcal{A}$ and $\overline{\mathcal{A}}$ is given in Table 1. Suppose the probabilities of t's occurrence in $\mathcal{A}$ and $\overline{\mathcal{A}}$ are p_1 and p_2, respectively. The likelihood ratio test verifies the hypothesis that the probabilities of the term t's occurrence in $\mathcal{A}$ and $\overline{\mathcal{A}}$ have the same value p and is thus written as:

$$\lambda(t) = \frac{\max\limits_{p} p^{k_1}(1-p)^{n_1-k_1} p^{k2}(1-p)^{n_2-k_2}}{\max\limits_{p_1,p_2} p_1^{k_1}(1-p_1)^{n_1-k_1} p_2^{k_2}(1-p_2)^{n_2-k_2}}, \lambda \in [0, 1] \tag{9}$$

where k_1 and k_2 are the frequencies of t in $\mathcal{A}$ and $\overline{\mathcal{A}}$, and n_1 and n_2 are the total number of terms in $\mathcal{A}$ and $\overline{\mathcal{A}}$, respectively. Referring to the contingency table, the variables in Equation 9 are computed by $k_1 = a$, $k_2 = b$, $n_1 = a + c$, $n_2 = b + d$, max $p = \frac{a+b}{a+b+c+d}$, max $p_1 = \frac{a}{a+c}$, and max $p_2 = \frac{b}{b+d}$. Terms with low λ values will tend to have distinct occurrence probabilities in the target domain $\mathcal{A}$ and the contrasting domain $\overline{\mathcal{A}}$. They can be used to separate $\mathcal{A}$ and $\overline{\mathcal{A}}$.

Finally, besides the likelihood ratio test, we also think the document frequency of a term in the document set, df, is a good measure for judging a term's relevance to the target domain. If a term a appears in multiple documents, it would be more relevant compared with those appearing in a single document, even though a has the same term frequency as those terms.

With the above considerations we develop the Domain Relevance Measure for identifying domain-relevant concepts. Specifically, given a multiword term t extracted in the collection $\mathcal{A}$ with the linguistic filters, its Domain Relevance value $DRM(t)$ is computed by:

$$DRM(t) = \frac{tf(t)}{max(tf)} \times \frac{|\log\lambda(t)| - min|\log\lambda|}{max|\log\lambda| - min|\log\lambda|} \times \frac{df(t)}{max(df)}, \tag{10}$$

where $min|\log\lambda|$ is the minimum $|\log\lambda(t)|$ value found, $max|\log\lambda|$ is the maximum $|\log\lambda(t)|$ value found, and $DRM(t) \in [0, 1]$. Terms with high DRM values are selected as the concepts of the domain ontology.

Word Sense Disambiguation

After terms are extracted, a WSD algorithm is used to identify the intended meaning of each term in the target domain. The results will mainly be used later for taxonomic relation extraction.

In this research we develop a variant of the LESK algorithm (Lesk, 1986), known as VLESK, for word sense disambiguation. The VLESK algorithm is an automatic and unsupervised method, which can be used in different domains without retraining.

LESK Algorithm

As a well-known unsupervised WSD algorithm, the LESK algorithm disambiguates word senses using a context window based on two assumptions. First, if words are close to each other in the sentence, they would be related to the same topic. Second, if they talk about the same topic, their glosses in the dictionary should contain the same words. Lesk (1986) demonstrates the LESK algorithm for separating the sense of *Cone* in the term *Pine Cone* from that of *cone* in the term *Ice Cream Cone*.

In the Oxford Advanced Learner's Dictionary of Current English, there are three senses of *Cone* and two senses of *Pine*, as shown in Table 2. As the first sense of *Pine* and the third sense of *Cone* both contain the same words, evergreen tree, the sense of *Cone* in *Pine Cone* is thus the third sense in the dictionary, which is different from its sense in *Ice Cream Cone.*

TABLE 2. The senses of cone and pine in the dictionary.

Cone:
1. Solid body which narrows to a point
2. Something of this shape whether solid or hollow
3. Fruit of certain evergreen trees

Pine:
1. Kinds of evergreen tree with needle-shaped evergreen tree
2. Waste away through sorrow or illness.

WordNet

WordNet (Fellbaum, 1998), like traditional dictionaries, contains terms and glosses. But it differs from traditional dictionaries in many aspects. For instance, terms in WordNet are organized semantically instead of alphabetically. In addition, synonym terms are grouped together in synonym sets (called Synset). Each Synset represents a particular sense of the term, which may be linked to other Synsets by certain semantic relations in WordNet.

WordNet stores terms according to four POS tag categories: Noun, Verb, Adjective, and Adverb. In WordNet 2.0, there are 114,648 nouns stored in 79,689 Synsets, 11,306 verbs stored in 13,508 Synsets, 21,346 adjectives stored in 18,563 Synsets, and 4,669 adverbs stored in 3,664 Synsets.

The major semantic relations between nouns are the Hypernym and Hyponym relations. If the Synset A is linked to Synset B using the Hypernym relation, A is a kind of B. For instance, the Synset {attack, onslaught, onset, onrush} represents the sense: *(military) an offensive against an enemy (using weapons).* Since it is linked to the Synset {operation, military operation} through the Hypernym relation, we infer that {attack, onslaught, onset, onrush} is *a kind of* {operation, military operation}. The Hypernym and Hyponym relations are reversible and transitive. In addition, the Meronym (has-a part of) and Homonyms (is-a part of) relations are also used to connect nouns.

For adjectives and adverbs, the key semantic relations are the *Similar* and *Also-see* relations. Adjective or adverb Synsets are linked by the *Similar* relation if the two Synsets are semantically similar.

There exists one other kind of semantic relation, known as *Attribution* relation, linking adjective Synsets to noun Synsets in WordNet. If the adjective Synset A is a value of the noun Synset B, A is linked to B by the Attribution relation. For instance, the Synset {domestic} is linked to the Synset {domesticity} by the Attribution relation.

VLESK Algorithm

The VLESK algorithm implements the original LESK algorithm with the use of WordNet, similar to previous work (Banerjee & Pedersen, 2002; Voorhees, 1993). Particularly, for a target word, the glosses of its related words in WordNet are concatenated with its own gloss as the input (for noun word we utilize the *Hypernym, Hyponym, Meronym*, and *Homonyms* relations, and we use the *Similar, Also-see*, and *Attribute* relations for adjectives and adverbs). The sense whose gloss shares most common words with those of the neighbor words is selected. If no sense wins, the target word will be assigned its first sense stored in WordNet, as the first sense is the most frequent one in normal use.

TABLE 3. The 18 sense combinations of "international terrorist attack."

1: international(adj)_1 terrorist(adj) _1 attack(noun) _1
2: international(adj)_2 terrorist(adj)_1 attack(noun) _1
. . .
17: international(adj)_1 terrorist(adj)_1 attack(noun)_9
18: international(adj)_2 terrorist(adj)_1 attack(noun)_9

Note that the original LESK algorithm disambiguates each word individually. In VLESK, we adopt a parallel disambiguation approach, which is based on the assumption that the chosen sense for a word depends on the senses of its surrounding words. Particularly, given an extracted multiword term, all possible combinations of the senses of the words in the term are considered simultaneously. A score is computed for each sense combination based on the number of the same words in the expanded glosses of the words. The highest scoring combination is picked as the most appropriate one and each word in the term is assigned its corresponding sense in the winning combination. For instance, given a term "international (ADJ), terrorist (ADJ), attack (NOUN)," there are two senses of international (ADJ), one sense of terrorist (ADJ), and nine senses of attack (NOUN) in WordNet. The 18 sense combinations of the three words are shown in Table 3, among which the highest scoring combination is the first combination.

The disadvantage of the parallel disambiguation approach is that the algorithm is very computationally intensive. Assuming that there are N words on average in a term and S senses on average per word, there are S^N combinations to be compared. Similar approaches that simultaneously disambiguated all words in a context window are adopted in (Agirre & Rigau, 1996; Cowie, Guthrie, & Guthrie, 1992).

Semantic Relation Extraction

We extract semantic relations between multiword terms as well as relations between multiword terms and single-word terms from a text collection. This module is also critical for the system's overall performance.

Recall that we assume single-word terms are important concepts if they appear in the extracted multiword terms or they are linked to the multiword terms with semantic relations. During semantic relation extraction, we may add certain single-word terms into the ontology if they are linked to the multiword terms detected through nontaxonomic relations.

Taxonomic Relation Extraction

Taxonomic relations are the most important semantic relations in a domain ontology, the extraction of which has been

TABLE 4. The lexico-syntactic patterns used for taxonomic relation extraction.

1. NP_0 such as $NP_1\{, NP_2, \ldots (and\|or) NP_n\}$,	Hyponym (NP_i, NP_0);
2. NP_1 is a kind of NP_0,	Hyponym (NP_1, NP_0);
3. $NP\{, NP\}^*\{, \}$ or other NP_0,	Hyponym (NP, NP_0);
4. $NP\{, NP\}^*\{, \}$ and other NP_0,	Hyponym (NP, NP_0);
5. NP_0, including $\{NP\}^*$ or/and NP,	Hyponym (NP, NP_0);

well studied in the field of lexicon building. A simple method for taxonomic relation extraction is string matching. For instance, *international terrorist organization* is recognized as a Hyponym of *terrorist organization* as they contain the same syntactic head *terrorist organization*. Another method is using lexico-syntactic patterns to extract taxonomic relations. The CRCTOL system uses a combination of both methods.

Extracting through lexico-syntactic patterns. The first method utilizes the well-known lexico-syntactic patterns (Etzioni et al., 2004; Hearst, 1992) for taxonomic relation extraction. For instance, the lexico-syntactic pattern "such as" (Hearst, 1998) is a popularly used pattern for taxonomic relation extraction. Given a sentence containing the "such as" pattern:

$$NP_0 \text{ such as } NP_1\{, NP_2, \ldots, (\text{and}|\text{or})NP_n\},$$

Hyponym relations (NP_i, NP_0) (for $i = 1$ *to* n) are extracted from the sentence, where the term NP_i is seen as a kind of term NP_0.

A total of five lexico-syntactic patterns are used in our system for taxonomic relation extraction. They are listed in Table 4.

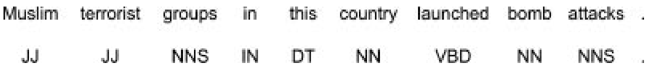
Extracting through term structure. Taxonomic relations can also be extracted based on the term structure through string matching. Several heuristics are developed and described below.

1. For terms of the form [*word, head*], if there is a term [*head*] in the ontology, establish a taxonomic relation between [*word, head*] and [*head*]. This method is similar to string match. However, the sense of the matched words, which is identified by our VLESK algorithm, must be the same in [*word, head*] and [*head*]. For instance, the sense of word *attack* will be the same in term *terrorist attack* and term *international terrorist attack* if *international terrorist attack* is identified as a kind of *terrorist attack*.

2. The semantic relations in WordNet can also be used for taxonomic relation extraction. In CRCTOL, we only use the taxonomic and synonymic relations. There are several cases for utilizing them.

a. If both *term*1 and *term*2 are in WordNet, and there exists a Hyponym or Hypernym relation between *term*1 and *term*2, a taxonomic relation (*term*1, *term*2) is extracted.

b. For terms of the form: *term*1 (*word*11,. . .,*word*1_n, *head*1), and *term*2 (*word*2_1,. . .,*word*2_n, *head*2), if *term*1 has been found holding the taxonomic relation with *term*0 in stage (1), and *head*1 is in the same Synset as *head*2 in WordNet, a taxonomic relation is extracted for *term*2 and *term*0. For instance, if we have the taxonomic relation (*terrorist group, group*), since the sense of the word *organization* in term *terrorist organization* is in the same Synset as that of *group* in term *terrorist group*, we can have the relation (*terrorist organization, group*).



Muslim	terrorist	groups	in	this	country	launched	bomb	attacks	.
JJ	JJ	NNS	IN	DT	NN	VBD	NN	NNS	.

FIG. 4. The POS tags assigned for the sample sentence.

Nontaxonomic Relation Extraction

Same as the conventional approach of learning nontaxonomic relations (Buitelaar, Olejnik, & Sintek, 2004; Ciaramita, Gangemi, Ratsch, Saric, & Rojas, 2005; Gamallo, Gonzales, Agustini, Lopes, & Lima, 2002), we hypothesize that verbs indicate nontaxonomic relations between concepts. A semantic relation of the (*Concept*, *Relation*, *Concept*) is thus extracted if its lexical realization can be found from the texts, which is in the form of ($Noun_1$, *Verb*, $Noun_2$), where $Noun_1$ is the subject of *Verb* and $Noun_2$ is the object of *Verb*.

We adopt a rule-based method for extracting (*Noun*, *Verb*, *Noun*) tuples from texts, similar to these conventional approaches. These tuples are used to represent the nontaxonomic relations between the concepts extracted. The noun and verb terms are identified by the regular expressions below:

$$Noun : (DT)?(JJ)^*(NN|NNS|NNP|NNPS)^+$$
$$Verb : (VB|VBD|VBN|VBZ)^+$$

where JJ represents an adjective, NN, NNS, NNP, and NNPS represent nouns, DT represents an article, and VB, VBD, VBN and VBZ represent verbs.

However, different from the conventional approaches that only utilize POS tagging or shallow (or light) parsing techniques for nontaxonomic relation extraction, our rules are based on the parse trees obtained with the full-text parsing technique. These parse trees would provide grammatical relations between phrases or words in the sentences, allowing us to find the nontaxonomic relations effectively.

An example to illustrate the difference between CRCTOL and the conventional approaches is given below, where we extract relations from the sentence: *Muslim terrorist groups in this country launched bomb attacks*.

If we use the POS tag-based rules to extract nontaxonomic relations, for example, use the following rule defined in Text2Onto:

$$(NN|NNS) + (VBD) \quad (NN|NNS)+,$$

the relation *(Country, Launch, Bomb Attack)* will be extracted from the sentence (refer to Figure 4). Although this extracted tuple satisfies the rule, it is wrong in the semantic sense.

In CRCTOL, the parse tree is utilized for nontaxonomic relation extraction. By analyzing the parse tree of this sample sentence (see Figure 5), we find the true subject of the

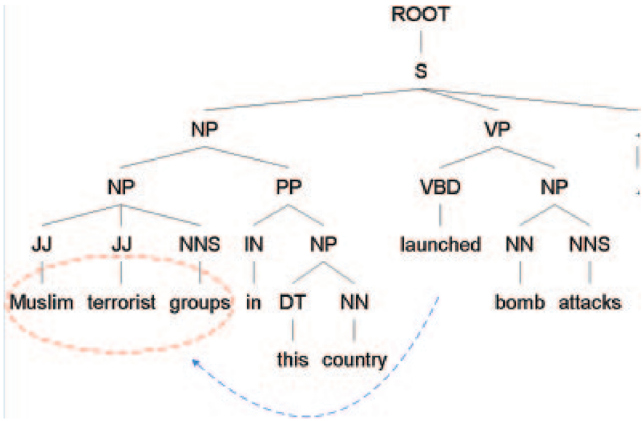


FIG. 5. The parse tree of the sample sentence.

verb *Launch* is *Muslim Terrorist Group*. The correct relation (*Muslim Terrorist Group, Launch, Bomb Attack*) is thus extracted.

In fact, the primary motivation behind CRCTOL for using a full text parsing technique is because of its effectiveness for nontaxonomic relation extraction. By adopting such an approach, the built ontology contains much more semantics, supporting more advanced applications.

Ontology Mapping

When concepts and relations are extracted from texts, they are classified and integrated to form an ontology. First, taxonomic relation mapping is performed, which builds the main structure of the domain ontology. Then, nontaxonomic relation mapping is performed, linking concepts with other types of semantic relations.

Taxonomic Relation Mapping

Taxonomic relation mapping is straightforward. Because the taxonomy is transitive, relations that can be derived are removed. For instance, given three relations *(terrorist group, group), (international terrorist group, terrorist group)*, and *(international terrorist group, group),* the last one will be removed.

Nontaxonomic Relation Mapping

We adopt a variant of the generalized association rule-mining algorithm for mapping nontaxonomic relations. First, all nontaxonomic relation tuples of concept C_i in which C_i is the subject of the relation (C_i, R, C_k) are collected. Then tuples that have similar verbs (i.e., the verbs can be found in the same Synset in WordNet) and the same object concepts are merged. Finally, only tuples with certain confidence and support values are kept. Note that the setting of the support and confidence values depends on the size of the datasets. The procedure for merging nontaxonomic relations is presented in Algorithm 1. The resultant ontology is a semantic network representing the discourse universe of the target domain.

Experiments

We conducted two case studies in which the CRCTOL system is used to build a terrorism domain ontology and a sport event ontology. For the first case study, documents of the US State Department report “Patterns of Global Terrorism (1991–2002)” are downloaded from the Website of the Federation of American Scientists (http://www.fas.org/irp/threat/terror.htm) as the test corpus. The PGT corpus contains a total of 104 html files, each of which is about 1,500 words. For the second case study, the SmartWeb Football dataset (http://www.dfki.de/sw-lt/olp2_dataset/) is used, which consists of 3,542 English documents. Both case studies use the same contrasting corpora, which are collected from the TREC collection, covering the commercial, computer, energy, and general domains.

As new methods are proposed for the three major tasks of concept extraction, word sense disambiguation, and nontaxonomic relation extraction, we first conduct experiments to evaluate the three individual components’ performance. A subset of the PGT dataset is manually annotated as the benchmark dataset. The corresponding components of Text-To-Onto[6] and its successor Text2Onto[7] are used as the baselines. In addition, a version of the CRCTOL system implemented with the Stanford Parser (Klein & Manning, 2002) is compared to evaluate the robustness of the proposed methods in handling different full-text parsing tools.

Then we estimate the system’s overall performance by evaluating the quality of the ontologies built from the two text collections. In particular, as no benchmark ontology is available for the PGT dataset, we use a set of quantitative and qualitative methods to evaluate the quality of the learned ontology, which include a structural property-based method to verify the quality of the learned ontological network, comparing with WordNet based on the taxonomic relations extracted, and scoring the learned ontology by the experts. As for the SmartWeb Football dataset, we directly compare our results with the accompanied human-edited benchmark ontology. The details are described in the following sections.

Component Level Evaluation

Concept extraction.

Multiword term extraction. Ontology learning systems with shallow NLP techniques utilize only lexical information for multiword term extraction. In CRCTOL, we make use of both syntactic information and lexical information. Experiments are conducted to evaluate CRCTOL’s performance for multiword term extraction against Text-To-Onto and Text2Onto, which use a POS tag-based rule defined by the following regular expression:

$$(DT)?(VBG|JJ|JJR|JJS)*(NN|NNS)+$$

[6] Version1.0, released 09/11/2004, http://sourceforge.net/projects/texttoonto

[7] Version Beta 4, released 07/08/2008, http://ontoware.org/projects/text2onto/

Algorithm 1. Nontaxonomic relation merging algorithm.

```
For each leaf concept $C_i$ in the concept hierarchy tree:
Repeat
    merge tuples ($C_i$, $R$, $C_k$), where $C_i$ is the subject of relation $R$ and $C_k$ are concepts which are
associated to $C_i$ through relation $R$.
    identify tuples of $C_i$'s sibling $C_j$ such that ($C_j$, $R$, $C_k$).
    If $C_i$ and $C_j$ have the same semantic relation $R$ with $C_k$
        If tuple ($C_h$, $R$, $C_k$) exists, where $C_h$ is a hypernym concept of $C_i$, the frequency of ($C_h$, $R$, $C_k$)
          is its frequency found directly in texts plus those of $C_i$ and $C_j$
        Else create new tuple ($C_h$, $R$, $C_k$). Its frequency is the sum of $C_i$ and $C_j$.
Until no tuples can be merged.
prune tuples whose support and confidence values are below certain threshold values.
```

where DT is the POS tag to represent an article, JJ , JJR, and JJS are the POS tags to represent adjectives, VBG is the POS tag to represent gerunds, and NN and NNS are the POS tags for nouns.

Documents of the PGT corpus (1991–1994) are used as the test corpus. Manual annotation of the document set identifies 3,269 multiword terms used as the target list for evaluation. The four components are then used to extract multiword terms from the texts separately. Their performance, in terms of precision, recall, and F-measure, is summarized in Table 5. We can see both versions of CRCTOL outperform Text-To-Onto and Text2Onto in the experiment, showing that our approach is effective in multiword term extraction.

However, further investigations find the poor performance of Text-To-Onto and Text2Onto may also be caused by other factors. For example, although using the above POS tag-based rule to extract multiword terms, Text2Onto just returns those multiword terms whose words are all nouns. As a result, given the sample sentence: *Some African countries have been the venue for terrorist activity in the past*, the multiword term *terrorist activity* is not returned, as *terrorist* is tagged as "JJ." In addition, the different NLP tools used for processing texts affect the performance. Therefore, we reimplement this defined rule with our NLP tools to extract multiword terms from the texts for comparison.

The performance of our reimplemented POS tag-based rule for multiword term extraction on the texts processed by the Berkeley Parser and the Stanford Parser, in terms of precision, recall, and F-measure, is summarized in Table 6. We can see the performance of the POS tag-based rule for multiword term extraction is indeed not that poor, but is slightly lower than that of CRCTOL under the same conditions. The lower precision scores of the POS tag-based rule can be attributed to its deficiency in separating gerunds from verbs' present participle. For example, following the definition, it identifies *harboring representatives* from the sentence *The Government of Sudan persisted in harboring representatives of Mideast terrorist groups* as a term. In contrast, CRCTOL works more effectively, since this sentence can be parsed as *(VP (VBG harboring)) (NP (NP (NNS representatives)) (PP (IN of) (NP (NNP Mideast) (JJ terrorist) (NNS groups))))*, which indicates *harboring* should not be extracted.

Domain concept extraction. After evaluating the performance for multiword term extraction, we use the same corpus to assess the ability of the proposed procedure with the DRM measure in identifying domain-relevant concepts. Manual annotation of the 33 documents identifies 496 single-word terms and 2,311 multiword terms as domain-specific concepts. The performance is evaluated in terms of precision and recall of the top K concepts selected.

As we can only control the number of multiword terms to be selected by CRCTOL, the K value is thus determined by the results of CRCTOL. In our experiments, we set the number of multiword terms to be selected as 100, and finally there are 125 terms selected by CRCTOL from the documents.[8] Therefore, the number of concepts to be selected by Text-To-Onto and Text2Onto is also set to 125.[9]

As shown in Table 7, both versions of CRCTOL produce much better performance in identifying domain-specific concepts. However, due to the many missing multiword terms shown in the previous experiments, such results are not sufficient to demonstrate the effectiveness of CRCTOL.[10] Therefore, we pair the procedures used in Text-To-Onto and Text2Onto with the Berkeley Parser and the Stanford Parser to extract domain-relevant concepts for comparison.

Note that Text-To-Onto and Text2Onto do not use the same procedure to generate candidate terms. In particular, given a multiword term extracted from the texts, for example, *international terrorist group*, Text-To-Onto further generates a possible candidate, say *terrorist group*, whereas Text2Onto does not. In our reimplementations, we follow their procedures strictly.

The performance of the reimplemented Text-To-Onto and Text2Onto for concept extraction on the texts processed by the Berkeley Parser and the Stanford Parser is shown in Table 8. Although both yield good results, their performance is still lower than that of CRCTOL. As we have shown that the difference in multiword term extraction is small, the better performance of CRCTOL in identifying domain-relevant concepts should be attributed to the effectiveness of

[8] The Δ is 2 and both versions of CRCTOL selected 25 single-word terms from the multiword terms.

[9] All the systems use a same stopword list for experiments.

[10] Note that there are also some other measures such as the entropy measure and the C/NC measure implemented in Text-To-Onto and Text2Onto for concept extraction. However, the former suffers from the same problem as the DR&DC measure and the latter is only for multiword term extraction, which make them not suitable for concept extraction. Therefore, we only use the tf/idf measure with Text-To-Onto and Text2Onto for concept extraction.

Table 5. The performance of Text-To-Onto, Text2Onto, and CRCTOL in multiword term extraction.

	Multi-word terms extracted				
System	Correct	Wrong	Precision	Recall	*F*-Measure
Text-To-Onto	2,707	352	88.5%	82.8%	85.5%
Text2Onto	988	32	96.7%	30.2%	46.1%
CRCTOL(+Berkeley Parser)	3,090	32	99.7%	97.4%	98.6%
CRCTOL(+Stanford Parser)	3,113	62	93.5%	95.9%	94.7%

Table 6. The performance of the reimplemented POS tag-based rule for extracting multiword terms from the texts processed by the Berkeley Parser and the Stanford Parser.

	Multiword terms extracted				
Parser for processing texts	Correct	Wrong	Precision	Recall	*F*-Measure
Berkeley Parser	3,098	188	95.7%	97.1%	96.4%
Stanford Parser	3,089	219	90.3%	95.8%	93.1%

Table 7. The performance of Text-To-Onto, Text2Onto, and CRCTOL in concept extraction.

			Correct concepts extracted	
System	Precision	Recall	Single-word	Multiword
Text-To-Onto	47.2%	2.1%	51	8
Text2Onto	74.4%	3.3%	88	5
CRCTOL (+Berkeley Parser)	92.8%	4.1%	24	92
CRCTOL(+ Stanford Parser)	92.0%	4.1%	23	92

Table 8. The performance of the reimplemented Text-To-Onto and Text2Onto for extracting concepts from the texts processed by the Berkeley Parser and the Stanford Parser.

				Correct concepts extracted	
Parser for processing texts	Concept extraction component	Precision	Recall	Single-word	Multiword
Berkeley Parser	Text-To-Onto	80.0%	3.5%	86	14
	Text2Onto	84.8%	3.7%	89	17
Stanford Parser	Text-To-Onto	76.0%	3.4%	82	13
	Text2Onto	80.0%	3.5%	87	13

the proposed concept extraction procedure with the DRM measure.

Finally, we evaluate the robustness of these concept extraction components in handling datasets with different term and document distributions. In particular, we change the test corpus used by concatenating documents of the same year into a single file. Therefore, we have four documents as the inputs, each of which is very long. The robustness is then evaluated by comparing the performance of the different components on the two corpuses.

As the original Text-To-Onto and Text2Onto systems have shown their deficiency in the previous experiment, we use our enhanced reimplementations with the Berkeley and Stanford Parsers as the baselines in the last experiment. The performance is also compared based on the top 125 concepts extracted by these components.

The performance of CRCTOL compared with those of the reimplemented Text-To-Onto and Text2Onto on the new corpus is given in Table 9. We can see that both versions of CRCTOL produce a similarly high level of performance on the two corpuses, while the performance of Text-To-Onto and Text2Onto degrades greatly in the new corpus. Such results clearly show that the concept extraction component in CRCTOL is more robust in identifying domain-relevant concepts from document sets with different term and document distributions.

Table 9. The performance of CRCTOL compared with those of the reimplemented Text-To-Onto and Text2Onto for concept extraction on the new test corpus.

Parser for processing texts	Concept extraction component	Precision	Recall	Correct concepts extracted	
				Single-word	Multiword
Berkeley Parser	Text-To-Onto	55.2%	2.4%	33	36
	Text2Onto	60.0%	2.7%	34	41
	CRCTOL	92.8%	4.1%	23	93
Stanford Parser	Text-To-Onto	48.8%	2.2%	30	31
	Text2Onto	56.8%	2.5%	33	38
	CRCTOL	95.2%	4.2%	24	95

Discussion. We have presented the performance of CRCTOL for concept extraction. We can see the two versions of CRCTOL produce roughly equivalent performance in the experiments, showing that the influence of the different full-text parsing tools used on concept extraction is small. Meanwhile, they both greatly outperform Text-To-Onto and Text2Onto, showing that this component is effective for extracting domain-relevant concepts. It is notable that the better performance of CRCTOL is mainly attributed to the effectiveness of the proposed concept extraction procedure with the DRM measure in identifying domain-relevant terms. Incorporating syntactic information for multiword term extraction does not improve the performance of concept extraction greatly.

Word sense disambiguation. We evaluated the performance of the VLESK algorithm by assigning senses to each word of the 100 multiword terms extracted by CRCTOL in the previous concept extraction stage. Note that we use the term itself as the context window for disambiguation in the experiment. These single-word terms are thus not disambiguated and have several senses in the ontology.

As the VLESK algorithm performs better if the gloss contains more words and each Synset in WordNet also contains some example sentences, we further add these example sentences for word sense disambiguation. Particularly, only nouns, verbs, adjectives, and adverbs in the example sentences are added. Meanwhile, a stop word list, which contains 40 words, is used to remove common words such as *a, which, that,* and *me* from the glosses.

Note that Text-To-Onto and Text2Onto do not provide word sense disambiguation function. We thus apply two other baseline algorithms in this experiment. The first one is the WordNet 1^{st} sense method (Moldovan & Novischi, 2004) which assigns a word its first sense in WordNet. The other one is the random sense algorithm that assigns the word sense randomly. Then, given the senses predicted by the three algorithms, the performance is computed as the number of the words correctly disambiguated divided by the total number of words disambiguated.

The performance of the three algorithms is presented in Table 10. We can see that our VLESK algorithm is much better than the random sense algorithm, since the random algorithm is an exceedingly cheap solution. However, our algorithm does not outperform the WordNet 1^{st} sense baseline greatly, especially on terms with three words. This is because each word's first sense in WordNet is obtained from a large amount of human annotated text and is the most frequent one. Nevertheless, our algorithm is still the best one.

Table 10. The performance of the VLESK algorithm compared with the two baseline algorithms.

Algorithm	Two-word terms	Three-word terms	All terms
VLESK	78.6%	87.5%	79.1%
Random	47.3%	75.0%	50.5%
WordNet 1st	71.4%	87.5%	73.7%

It is noticeable that the performance of all three algorithms in our experiments, even the random sense algorithm, are better than those reported results of the LESK-like algorithms, whose average precisions are about 50% on the benchmark datasets (Banerjee & Pedersen, 2002). Such an improvement would be attributed to the target domain we work on and the dictionary we use. First of all, the terms extracted from the terrorism domain documents typically have specific meanings. In addition, WordNet has collected many terms relevant to the terrorism domain and their associated relations can be used for sense disambiguation. For example, the stored hyponym relation between the second sense of *act* and *terrorist attack* in WordNet helps to identify the correct sense of *act* in the term *terrorist act*. Therefore, our WSD algorithms obtain generally better performance.

Semantic relation extraction. As all the systems use a similar approach to taxonomic relation extraction, we only compare the performance of CRCTOL with those of Text-To-Onto and Text2Onto in nontaxonomic relation extraction. In particular, we first conduct experiments on simple structure sentences.[11] Then we evaluate the performance on general

[11] A simple structure sentence is in the form Subject + Verb + Object, where the Verb comes after its Subject and is followed by its Object, e.g., *five terrorists died in the battle*. No complex components such as adverbial clause or auxiliary verbs are used. The defined POS tag-based rules can thus easily extract the nontaxonomic relations from the texts.

Table 11. The performance of Text-To-Onto, Text2Onto, and CRCTOL for nontaxonomic relation extraction on simple structure sentences.

	Relations extracted				
System	Correct	Wrong	Precision	Recall	F-Measure
Text-To-Onto	29	1	96.8%	22.5%	59.6%
Text2Onto	27	6	81.8%	20.9%	51.4%
CRCTOL(+Berkeley Parser)	114	8	93.4%	88.4%	90.9%
CRCTOL(+Stanford Parser)	117	4	96.7%	90.7%	93.7%

Table 12. The performance of the reimplemented POS tag-based rules for nontaxonomic relation extraction on simple structure sentences.

Parser for	Relations extracted				
processing texts	Correct	Wrong	Precision	Recall	F-Measure
Berkeley Parser	93	17	84.5%	72.1%	78.3%
Stanford Parser	94	15	86.2%	72.9%	79.6%

sentences that include both simple structure sentences and complex structure sentences. Such a setting can clearly show the advantage of our proposed method for nontaxonomic relation extraction.

Experiments on simple structure sentences. Documents of the PGT corpus (1991–1997) are used as the test corpus in this experiment. Manual annotation of the documents identified 111 qualified sentences, containing 129 semantic relations. The four systems are then used to extract nontaxonomic relations from these qualified sentences.

The performance of the four systems, in terms of precision, recall, and F-measure, is summarized in Table 11. We can see that both versions of CRCTOL outperform Text-To-Onto and Text2Onto in the experiment, especially on the recall value. However, similar to the previous sets of experiments on concept extraction, the poor performance of the Text-To-Onto and Text2Onto systems for extracting relations from these simple structure sentences may also be due to other factors, such as the NLP tool used. Therefore, we further implement the POS tag-based rules defined in Text-To-Onto and Text2Onto to extract nontaxonomic relations for evaluating this approach's performance accurately.

The performance of the POS tag-based rules for nontaxonomic relation extraction on the texts processed by the Berkeley Parser and the Standard Parser, in terms of precision, recall, and F-Measure, is summarized in Table 12. We can see that the performance of the POS tag-based rule is in fact not that bad under the same conditions. But it still extracts fewer correct relations from the texts.

Experiments on general sentences. Documents of the PGT corpus (1991) are used as the test corpus. There are 289 sentences in the documents, containing 380 semantic relations. The four systems are used to extract nontaxonomic relations from these sentences. Their performance, in terms of precision, recall, and F-measure, is given in Table 13. We can see that both versions of CRCTOL outperform the baselines greatly when extracting nontaxonomic relation from the general sentences.

The same as the previous experiment on simple structure sentences, we also pair the POS tag-based rules together with the Berkeley and Stanford Parsers for nontaxonomic relation extraction. The performance of the POS tag-based rules with the Berkeley Parser and the Stanford Parser on the 289 sentences, in terms of precision, recall, and F-measure, is given in Table 14. Although all the systems' performance degrades in this experiment, we can see that the degradation of the POS tag-based rules is extremely great. It extracts much fewer relations from the texts, many of which are wrong.

Discussion. We have reported the CRCTOL's performance in nontaxonomic relation extraction. We can see that the influence of the full-text parsing tools used on nontaxonomic relation extraction is small, as both versions of the CRCTOL system extract many more relations from the texts, especially from the general sentences. Such differences are exactly due to the ineffectiveness of the POS tag-based rules in identifying the subjects and objects of the verbs (an example is given in Nontaxonomic Relation Extraction, above). When processing simple structure sentences, the deficiency is not obvious. However, for dealing with complex structure sentences its ineffectiveness becomes immediately apparent. For example, for the sentence *Sikh extremists probably also were responsible for a bombing in New Delhi in late April that killed three people*, it requires a deep understanding of the content to extract the correct relation *(bombing, kill, people)*, which cannot be handled with the POS tag-based rules. As a result, systems with the POS tag-based rules can extract only a few nontaxonomic relations from the texts, since ordinary documents mainly consist of complex structure sentences. As for the CRCTOL system, the only drawback of utilizing the parse tree for nontaxonomic relation extraction is that it requires more time to analyze the sentence structure and build the parse tree. But such costs are reasonable considering that many more relations can be extracted.

Ontology Level Evaluation

In the previous sections we reported the performance of the three components of the CRCTOL system separately. The experimental results show that these components

Table 13. The performance of Text-To-Onto, Text2Onto, and CRCTOL for nontaxonomic relation extraction on general sentences.

	Relations extracted				
System	Correct	Wrong	Precision	Recall	*F*-Measure
Text-To-Onto	21	2	91.3%	5.5%	48.4%
Text2Onto	3	1	75.0%	0.8%	37.9%
CRCTOL(+Berkeley Parser)	211	48	81.5%	55.5%	68.5%
CRCTOL(+Stanford Parser)	213	45	82.4%	55.3%	68.8%

Table 14. The performance of the reimplemented POS tag-based rules for nontaxonomic relation extraction on general sentences.

Parser for	Relations extracted				
processing texts	Correct	Wrong	Precision	Recall	*F*-Measure
Berkeley Parser	82	35	70.1%	21.6%	45.8%
Stanford Parser	81	35	69.8%	21.3%	45.6%

Table 15. The top 15 multiword terms extracted.

Terms	DRM
Terrorist group	0.6153
Terrorist attack	0.5729
International terrorism	0.3772
Terrorist act	0.2399
Terrorist activity	0.1758
Terrorist organization	0.1744
State sponsor	0.1647
Security force	0.1278
Car bomb	0.1006
Terrorist incident	0.0981
Terrorist operation	0.0604
Domestic terrorism	0.0509
Islamic extremist	0.0506
International terrorist attack	0.0416
Military personnel	0.0370

outperform the corresponding components of Text-To-Onto and Text2onto. In this section we estimate the CRCTOL system's overall performance by evaluating the quality of the ontologies learned from the two text collections.

The terrorism domain ontology. As no benchmark ontology is available for the PGT dataset, we use a set of quantitative and qualitative methods to make a more objective evaluation about the quality of the learned ontology.

Ontology building. In all, 200 multiword terms are selected from the initial 11,745 multiword terms as the domain ontology concepts. A high filtering rate is used due to the requirement of creating a concise target domain ontology. Table 15 lists the 15 most highly ranked multiword terms with the DRM values.

Recall that single-word terms can be added to the ontology during the semantic relation extraction stage. Besides the 47 single-word terms found in the concept extraction stage, 144 single-word terms are also added as the concepts of the final ontology, each of which has at least 12 relations linked to the multiword concepts. After relation mapping, there are 271 semantic relations kept in the ontology.

An example of the extracted concepts and the associated relations is shown in Figure 6. We can see that the concept "militant group" is a subclass of concept "group" in the ontology. It is linked to concepts "authority," "weapon," "facility," and "Pakistan" through nontaxonomic relations "surrender to," "acquire," "threaten," and "base in," respectively. The direction of the links indicates that "militant group" is the subject in these semantic relations.

To maintain a generic solution, our work so far does not discriminate concepts and instances. In fact, the distinction between concepts and instances depends on the task requirement and human judgment. For example, the term "Pakistan" is a concept in the ontology generated and it can also be considered as an instance of the concept "Country." Most ontology learning systems also do not make such a distinction. The few exceptions include Text-To-Onto (Maedche & Staab, 2000), which employs a predefined thesaurus, and Text2Onto (Cimiano & Völker, 2005), which implements specific methods for instance discovery.

Structural property-based method. It is known that for established knowledge networks, such as WordNet and Hindi WordNet, their graph representations, like the one shown in Figure 6, hold the *small world* property (Ramanand, Ukey, Singh, & Bhattacharyya, 2007). Since our built domain ontology is similar to these knowledge networks, its graph representation should also hold the same property. We therefore can indirectly gauge the quality of the built ontology by measuring whether its graph representation is consistent with that of a *small world* graph. It is more objective than human experts' judgment and can be easily implemented.

Degree Distribution: An essential characteristic of the *small world* graph is that its degree distribution $p(k)$ of the nodes in the graph follows a power-law distribution (Watts, 2003).

We present the degree distribution of the built ontology's graph representation and its log-log plot in Figure 7. We see that its degree distribution follows a power-law distribution that is characterized by an exponent $\gamma = -1.1754$, showing that this graph is a *small world* graph.

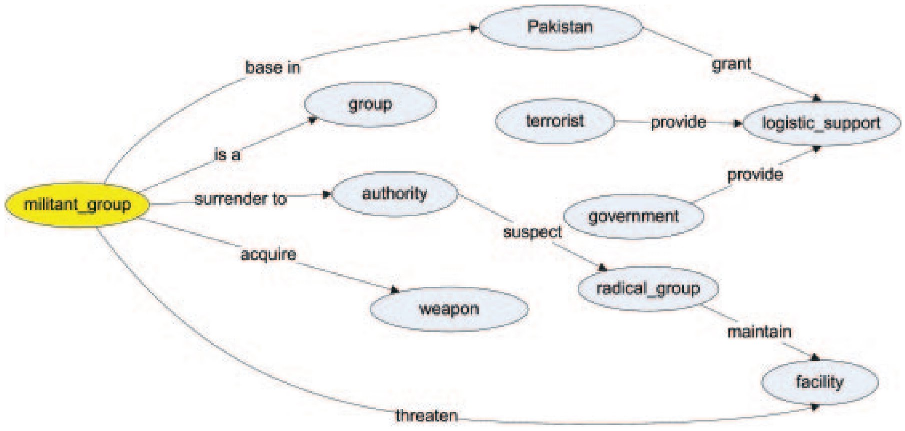


FIG. 6. The concept *militant group* and the relations around it.

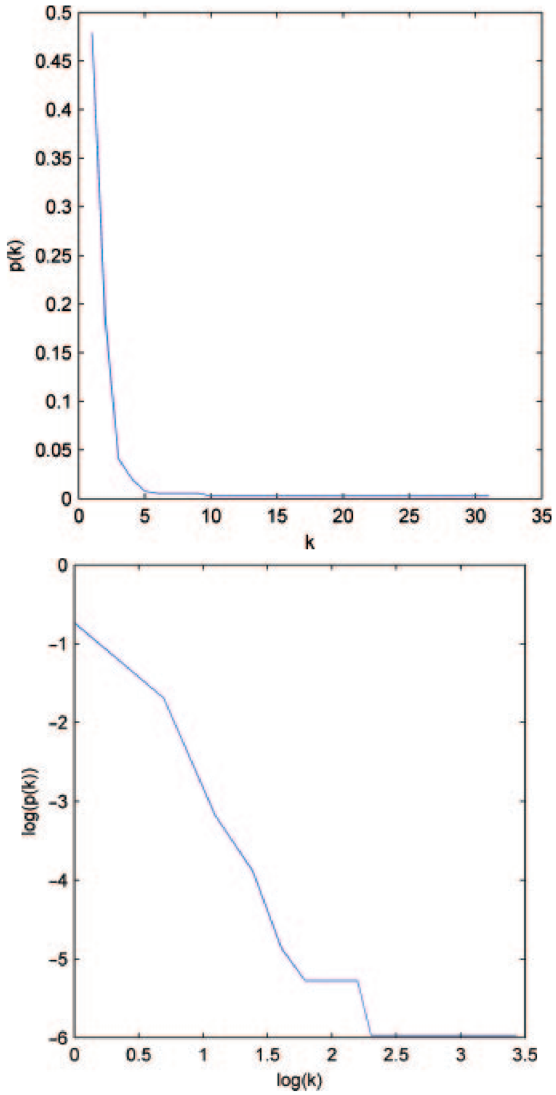


FIG. 7. The degree distribution $p(k)$ and its log-log plot for the domain ontology built on the PGT dataset.

Clustering Coefficient: A graph is considered *small world* if its average clustering coefficient $\bar{c}$ is much higher than that of a random graph constructed on the same node set, whose value is closer to $\frac{1}{N}$, where N is the number of nodes in the network (Watts, 2003).

For our graph, we compute an average clustering coefficient of:

$$\bar{c} = 0.5113$$

which is much greater than that of a random graph on the same node set ($\bar{c} = 0.0026$). This again supports the claim that the learned ontology has the *small world* property.

Paradigmatic relation evaluation. Paradigmatic relations, such as synonym relation and taxonomic relation, are patterns of association between lexical units that have high semantic similarities (Rapp, 2002). In this section we evaluate the paradigmatic relations learned by the CRCTOL system. Specifically, we refer to WordNet to judge the taxonomic relation extracted, which would require fewer subjective scores.

There are a total of 176 taxonomic relations stored in the built domain ontology. However, some of these relations are not suitable for evaluation, as they contain concepts that are not recorded in WordNet. For example, the concept *guerilla group* is not in WordNet, so we cannot check whether the taxonomic relation between *group* and *guerilla group* is correct with WordNet. After removing those relations, 28 taxonomic relations are used for assessment.

Among the 28 taxonomic relations, 19 are found in WordNet. Five relations not found in WordNet are judged by the experts as the correct ones for the terrorism domain. An example of such taxonomic relations is one between *terrorist attack* and *bombing*. Only four relations are found to be wrong. The overall accuracy of 85.7% illustrates the accuracy of the system in taxonomic relation extraction.

Human judgment. Finally, we use a qualitative method designed in ConceptNet (Liu & Singh, 2004) to assess the quality of the built ontology. Five students are employed for this evaluation. Each student is asked to rate 20 randomly selected concepts of the ontology, where the assessment is

Table 16. Two dimensions of evaluating the quality of the built domain ontology, rated by five students.

Goodness of concept (average score)	Noise of relation (average score)
3.48	1.60

performed along the two following dimensions, on a Likert 1 (strongly disagree) to 5 (strongly agree) scale:

- *Goodness of concept*. The students are asked to score whether the selected concept is good enough to be kept in the ontology.
- *Noise of relation.* The students are asked to rate whether the associated semantic relations of the selected concept contain wrong information or nonsensical data.

The results of this experiment are given in Table 16 and interpreted as follows. For the goodness of concept, most of the selected concepts are rated good enough as concepts of the terrorism domain. As for the noise of relation, only a few selected relations are rated as incorrect, showing that a relatively clean ontology has been built. On the whole, the scores indicate that the learned terrorism domain ontology is of good quality.

The sport event domain ontology. Different from the PGT dataset, a human-edited benchmark ontology is accompanied with the SmartWeb Football dataset, which has defined a set of key concepts and relations in the football event domain. Therefore, we could directly compare our result with this benchmark ontology for the concepts taxonomic relations and nontaxonomic relations extracted.[12] However, as the benchmark ontology is manually built independent of this text collection, not all the concepts and relations defined in the ontology can be found in the dataset. Also, certain important concepts and relations of the sport event domain are missed by the benchmark ontology but can be found in the dataset.

Concept extraction. This benchmark ontology consists of 608 concepts, which are represented by 1,007 terms, including both single-word terms and multiword terms in the texts. For example, the concept "LeagueFootballMatch" is defined to be represented by four terms in the texts, namely, *football league match, football league game, soccer league match*, and *soccer league game.* After removing concepts whose terms do not appear in the documents, there are 429 concepts, represented by 629 terms. Our experiments are then to evaluate how many out of the 629 terms can be extracted from the texts.[13]

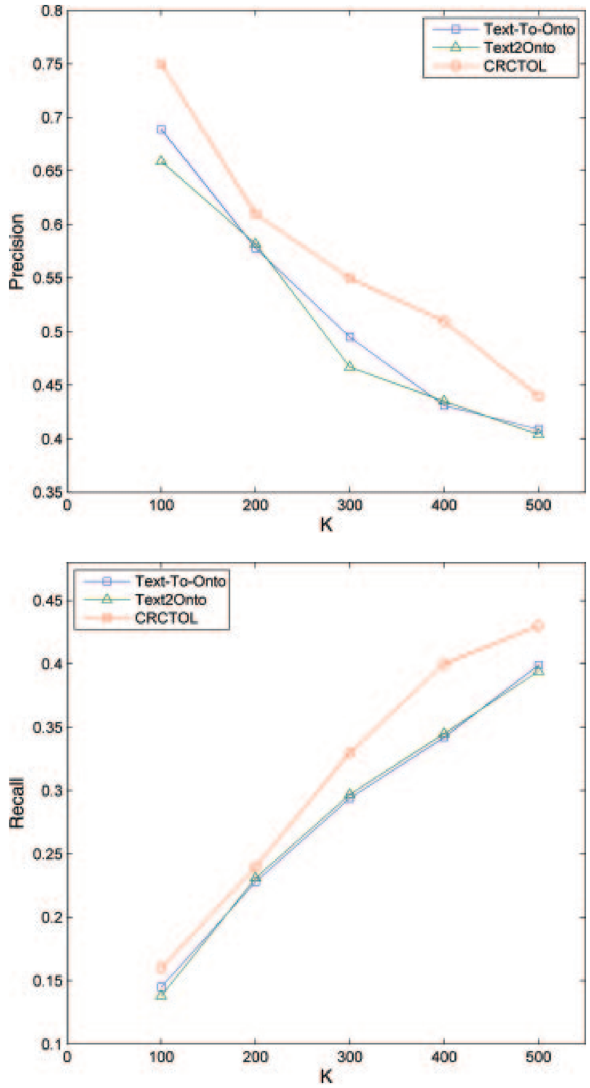


FIG. 8. The performance of Text-To-Onto, Text2Onto, and CRCTOL for concept extraction with different K values.

The experimental results of Text-To-Onto, Text2Onto, and CRCTOL, by setting the top 100, 200, 300, 400, and 500 multiword terms to be extracted by CRCTOL from the texts, are given in Figure 8. Compared with the results of the PGT dataset, the relatively poorer performance of CRCTOL on this SmartWeb Football dataset can be attributed to several factors. First, most of the terms occur with low frequency and document frequency in the dataset. For example, only about 150 out of the 629 terms have a frequency greater than 50 in the 3,542 documents. It is thus more difficult to separate the domain-relevant concepts with those irrelevant concepts. Second, some concepts are represented by different terms in the documents, for example, the concept *FIFA/Coca-Cola world ranking* is represented by *Coca-Cola world ranking* instead of *FIFA/Coca-Cola world ranking,* and the concept *LeagueFootballMatch* is represented by *league game* instead of *football league game* or *soccer league game*. In addition, some relevant terms of the sport event domain are not recorded in the benchmark ontology but found in the dataset,

[12] As we have shown that the deficiency of the original Text-To-Onto and Text2Onto systems in the previous experiments, we only compare CRCTOL with the reimplementations in this section.

[13] In this paper, we do not consider the problem of finding synonymic terms, as it is only used to refine the learned ontology but not necessary for learning an ontology. Also, people have studied this problem well and many effective methods (e.g., Baroni & Bisi, 2004), can be used to solve this problem.

Table 17. Summary of key features compared with other ontology learning systems.

	OntoBuilder	OntoLearn	Text-To-Onto /Text2Onto	CFG	CRCTOL
NLP tools	Not known	POS tagger Chunk Parser	POS tagger	POS tagger	POS tagger Full parser
Concept extraction measure	Not known	DR-DC	*tf*/*idf*	No	DRM
Taxonomic relation extraction	Yes	Yes	Yes	No	Yes
Nontaxonomic relation extraction	No	Not known	Yes	Yes	Yes

for example, the concept *team manager.* Nevertheless, we can see CRCTOL still outperforms Text-To-Onto and Text2Onto in the experiments and extract enough domain-relevant terms from the texts.

Taxonomic relation extraction. For simplicity, we only evaluate the taxonomic relation extracted based on the top 400 multiword terms extracted from the texts, as such a setting produces the highest F-Measure value in the concept extraction stage. Also, we do not compare CRCTOL with Text-To-Onto and Text2Onto in this experiment as the three systems use a similar approach for taxonomic relation extraction.

There are 633 taxonomic relations extracted for the 499 terms (99 single-word terms are found appearing frequently in the 400 multiword terms). After removing relations whose associated terms are not in included the benchmark ontology, there are 200 taxonomic relations used for evaluation.

First, we compute the number of taxonomic relations that can be directly matched with those taxonomic relations defined in the ontology. For example, as *central midfielder* has been specified as a subclass of *midfielder* in the ontology, the found relation is thus judged as correct. In total, there are 87 out of the 200 relations found in the benchmark ontology and the precision is calculated as 43.5%.

Then we compute the number of taxonomic relations that can be derived from the benchmark ontology. For example, since the relation "match day is a subclass of period" can be inferred by relations "match day is a subclass of TournamentRoundStage" and "TournamentRoundStage is a subclass of period," it is judged as correct as well. Under this condition, another 61 relations are qualified. The longest inference rule used involves six relations defined in the benchmark ontology.

With the above results, we can see a total of 148 out of the 200 extracted taxonomic relations are correct and the precision is 74.0%. Such a result demonstrates the effectiveness of our taxonomic relation extraction approach. But compared with the human-edited benchmark ontology, we find the structure of the learned ontology is relatively flat. The maximum depth of the learned ontology is 3 (e.g., *football match→match→competition*), as the system cannot effectively identify the subtle difference between the concepts from the texts. This limitation is also shared by other ontology learning systems. As such, human efforts are still required for refining the learned ontology.

Nontaxonomic relation extraction. Finally, we evaluated the performance of Text-To-Onto, Text2Onto, CRCTOL for nontaxonomic relation extraction when setting the top 400 multiword concepts extracted by CRCTOL.

There are 97 nontaxonomic relations defined in the benchmark ontology. By removing relations whose domain/range is literal and relations whose associated concepts cannot be found in the texts, 28 nontaxonomic relations are left as the benchmark.

Different from the previous experiments on taxonomic relation extraction, we cannot directly compare our results against the 28 nontaxonomic relations, as their representations are not comparable. For example, the relation *inMatch* in the benchmark ontology is defined as:

```
<rdf:Property rdf:about = "&sportevent;inMatch″
      rdfs:label = "inMatch″>
   <rdfs:range rdf:resource = "&sportevent;
      Football″/>
   <rdfs:domain rdf:resource = "&sportevent;
      MatchTeam″/>
</rdf:Property>
```

As no simple method can be used to map the verbs of these extracted relations (for example, *play in*, to the label of this relation, *inMatch*), we employed one student to manually compare the extracted relations with the 28 benchmark relations.

A total of 2,316 nontaxonomic relations are extracted by CRCTOL from texts. After ontology mapping, 250 relations are kept in the learned ontology. However, it is not appropriate to simply compare all the 250 extracted relations against the benchmark relations, as some relations' subject or object is not defined in the benchmark ontology or the relations, but they may be correct. Therefore, we only evaluate 126 out of the 250 extracted relations that share the same subject and object as the 28 benchmark relations.

As Text-To-Onto and Text2Onto do not provide solutions for removing unimportant relations, we simply remove relations whose frequency is 1 from their results (Text-To-Onto extracts 2,025 relations and Text2Onto extracts 2,059 relations from the texts). Finally, 46 relations are left for both systems for comparison.

The performances of the three systems for nontaxonomic relation extraction are as follows. For CRCTOL, 87 out of the 126 extracted relations are judged as correct, which can be mapped to nine relations defined in the benchmark ontology. The precision is 69.4%. For Text-To-Onto and Text2Onto, only 14 out of the 46 relations are judged as correct, which can be mapped to three relations defined in the benchmark ontology. The precision is 30.4%. We can see that CRCTOL outperforms Text-To-Onto and Text2Onto in nontaxonomic relation extraction again.

Conclusion

We have presented a system for ontology learning in this paper. The proposed CRCTOL system differs from other state-of-the-art ontology learning systems in a number of ways (Table 17). First, we adopt a full text parsing technique to obtain a more detailed syntactic level of information. Second, we use a different procedure including the developed DRM measure for concept extraction, which enables us to extract a concise set of domain-specific concepts more accurately. Finally, we use a rule-based method similar to the CFG approach for nontaxonomic relation extraction. This proves to be a feasible way to extract previously unknown semantic relations. Compared with traditional methods, our system produces ontologies that are more concise and accurate, and contain a richer semantics in terms of the range and number of semantic relations compared with alternative systems.

Although we have obtained promising results, our work can be extended in several directions. First, we utilized one of the most advanced full-text parsers for ontology learning. However, it may still produce a wrong parsing output, resulting in erroneous concepts or relations to be extracted. We expect to develop effective methods to remove the wrongly parsed results so as to improve the quality of the learned ontology. Also, our method was designed for building domain ontology from scratch. The functions for enriching an existing ontology and adapting an ontology for other application domains will be very useful for practical applications.

Acknowledgments

This work was supported by the Singapore National Research Foundation Interactive Digital Media R&D Program, under research Grant NRF2008IDM-IDM004-037. We thank the anonymous reviewers for providing many invaluable comments to the previous versions of the paper. We thank Dr. Paul Buitelaar for providing the SmartWeb Football Dataset for our experiments.

References

Agirre, E., & Rigau, G. (1996). Word sense disambiguation using conceptual density. In Proceedings of the 16th Conference on Computational Linguistics (pp. 16–22). Stroudsburg, PA: ACL.

Banerjee, S., & Pedersen, T. (2002). An adapted Lesk algorithm for word sense disambiguation using WordNet. In Proceedings of the Third International Conference on Intelligent Text Processing and Computational Linguistics (pp. 136–145). Berlin, Germany: Springer.

Baroni, M., & Bisi, S. (2004). Using cooccurrence statistics and the web to discover synonyms in technical language. In Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC 2004) (pp. 1725–1728). Paris: European Language Resources Association.

Basili, R., Rossi, G.D., & Pazienza, M.T. (1997). Inducing terminology for lexical acquisition. In Proceedings of the Second Conference on Empirical Methods in Natural Language Processing (pp. 125–133). Stroudsburg, PA: ACL.

Berners-Lee, T., Hendler, J., & Lassila, O. (2001). The semantic web: A new form of web content that is meaningful to computers will unleash a revolution of new possibilities. Scientific American, 285(5), 34–43.

Biébow, B., & Szulman, S. (1999). TERMINAE: A method and a tool to build a domain ontology. In Proceedings of the 11th European Workshop on Knowledge Acquisition, Modelling and Management (pp. 49–66). Berlin, Germany: Springer.

Bisson, G., Nédellec, C., & Cañamero, D. (2000). Designing clustering methods for ontology building—The Mo'K workbench. In Proceedings of the Workshop on Ontology Learning, 14th European Conference on Artificial Intelligence (pp. 13–19). Amsterdam: IOS Press.

Brunner, J.-S., Li, M., Chen, W., Lei, Z., Daniel, C.W., Yue, P., et al. (2007). Explorations in the use of semantic Web technologies for product information management. In Proceedings of the 16th International Conference on World Wide Web (pp. 747–756). New York: ACM Press.

Buitelaar, P., Olejnik, D., & Sintek, M. (2004). A Protégé plug-in for ontology extraction from text, the semantic Web: Research and applications. In First European Semantic Web Symposium (pp. 31–44). Berlin, Germany: Springer.

Casella, G. (1990). Statistical inference. Pacific Grove, CA: Brooks/Cole.

Ciaramita, M., Gangemi, A., Ratsch, E., Saric, J., & Rojas, I. (2005). Unsupervised learning of semantic relations between concepts of a molecular biology ontology. In Proceedings of the 19th International Joint Conference on Artificial Intelligence (pp. 659–664). San Francisco: Morgan Kaufmann.

Cimiano, P., & Völker, J. (2005). Text2Onto—A framework for ontology learning and data-driven change discovery. In Proceedings of the Tenth International Conference on Applications of Natural Language to Information Systems (Vol. 3513, pp. 227–238). Berlin, Germany: Springer.

Cowie, J., Guthrie, J., & Guthrie, L. (1992). Lexical disambiguation using simulated annealing. In Proceedings of the 14th Conference on Computational Linguistics (pp. 359–365). Stroudsburg, PA: ACL.

Daille, B. (1996). Study and implementation of combined techniques for automatic extraction of terminology (pp. 49–66). In J.L. Klavans, P. Resnic (Eds.), The Balancing act: Combining symbolic and statistical approaches to language. Cambridge, MA: MIT Press.

Dunning, T. (1993). Accurate methods for the statistics of surprise and coincidence. Computational Linguistics, 19(1), 61–74.

Engels, R. (2003). Ontology Extraction Tool, Deliverable 6. Retrieved September 26, 2009, from http://www.ontoknowledge.org/del.shtml

Etzioni, O., Cafarella, M., Downey, D., Kok, S., Popescu, A.-M., Shaked, T., et al. (2004). Web-scale information extraction in Knowitall: Preliminary results. In Proceedings of the 13th International Conference on World Wide Web (pp. 100–110). New York: ACM Press.

Fano, R.M. (1961). Transmission of Information. Cambridge, MA: MIT Press.

Faure, D., & Nédellec, C. (1998). A corpus-based conceptual clustering method for verb frames and ontology acquisition. International Conference on Language Resources and Evaluation Workshop on Adapting Lexical and Corpus Resources to Sublanguages and Applications (pp. 5–12). Paris: European Language Resources Association.

Fellbaum, C. (1998). WordNet: An Electronic lexical database. Cambridge, MA: MIT Press.

Gamallo, P., Gonzales, M., Agustini, A., Lopes, G., & Lima, V.S.D. (2002). Mapping syntactic dependencies onto semantic relations. European Conference on AI Workshop on Machine Learning and Natural Language Processing for Ontology Engineering (pp. 15–22). Amsterdam: IOS Press.

Gomez-Perez, A., & Manzano-Macho, D. (2003). Deliverable 1.5: A Survey of Ontology Learning Methods and Techniques. Retrieved from http://www.ontoweb.org/Members/ruben/Deliverable1.5

Gruber, T.R. (1993). A translation approach to portable ontology specification. Knowledge Acquisition, 5, 199–220.

Guarino, N., Masolo, C., & Vetere, G. (1991). OntoSeek: Content-based access to the web. IEEE Intelligent Systems, 14(3), 70–80.

Hearst, M.A. (1992). Automatic acquisition of hyponyms from large text corpora. In Proceedings of the 14th Conference on Computational Linguistics (pp. 539–545). Stroudsburg, PA: ACL.

Hearst, M.A. (1998). Automated discovery of wordnet relations (pp. 132–152). In C. Fellbaum (Ed.), WordNet: An electronic lexical database. Cambridge, MA: MIT Press.

Jiang, X., & Tan, A.-H. (2005). Mining ontological knowledge from domain-specific text documents. In Proceedings of the Fifth IEEE International Conference on Data Mining (pp. 665–668). Washington, DC: IEEE.

Jiang, X., & Tan, A.-H. (2006). OntoSearch: A full-text search engine for the semantic web. In Proceedings of the 21st National Conference on Artificial Intelligence and the 21st Innovative Applications of Artificial Intelligence Conference (pp. 1325–1330). Menlo Park, CA: AAAI Press.

Klein, D., & Manning, C.D. (2002). Fast exact inference with a factored model for natural language parsing (pp. 3–10). In Advances in Neural Information Processing Systems. Cambridge, MA: MIT Press.

Lesk, M. (1986). Automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from an ice cream cone. In Proceedings of the Fifth Annual International Conference on Systems Documentation (pp. 24–26). New York: ACM Press.

Liu, H., & Singh, P. (2004). ConceptNet: A practical commonsense reasoning toolkit. BT Technology Journal, 22(4), 211–226.

Maedche, A., & Staab, S. (2000). Mining ontologies from text. In Knowledge Acquisition, Modeling and Management, 12th International Conference (pp. 189–202). Berlin, Germany: Springer.

Missikoff, M., Navigli, R., & Velardi, P. (2002). The usable ontology: An environment for building and assessing a domain ontology. In International Semantic Web Conference 2002 (pp. 39–53). Washington, DC: IEEE.

Missikoff, M., Velardi, P., & Fabriani, P. (2003). Text mining techniques to automatically enrich a domain ontology. Applied Intelligence, 18(3), 323–340.

Moldovan, D., & Novischi, A. (2004). Word sense disambiguation of WordNet glosses. Computer Speech & Language, 18(3), 301–317.

Petrov, S., Barrett, L., Thibaux, R., & Klein, D. (2006). Learning accurate, compact, and interpretable tree annotation. In ACL-44: Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics (pp. 433–440). Stroudsburg, PA: ACL.

Rajaraman, K., & Tan, A.-H. (2003). Mining semantic networks for knowledge discovery. In Proceedings of the Third IEEE International Conference on Data Mining (pp. 633–636). Washington, DC: IEEE.

Ramanand, J., Ukey, A., Singh, B.K., & Bhattacharyya, P. (2007). Mapping and structural analysis of multi-lingual wordnets. IEEE Data Engineering Bulleting, 30(1), 30–44.

Rapp, R. (2002). The computation of word associations: Comparing syntagmatic and paradigmatic approaches. In 19th International Conference on Computational Linguistics (pp. 1–7). Stroudsburg, PA: ACL.

RDF Schema Specification. (2004). Retrieved September 26, 2009, from http://www.w3.org/TR/ rdf-schema/

Sagar, J.C., Dungworth, D., & McDonald, P.F. (1980). English special language: Principles and practice in science and technology. Wiesbaden, Germany: Oscar Brandstetter.

Salton, G., & McGill, M.J. (1986). Introduction to modern information retrieval. New York: McGraw-Hill.

Toutanova, K., Klein, D., Manning, C.D., & Singer, Y. (2003). Feature-rich part-of-speech tagging with a cyclic dependency network. In Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology (NAACL '03) (pp. 173–180). Stroudsburg, PA: ACL.

Voorhees, E.M. (1993). Using WordNet to disambiguate word senses for text retrieval. In Proceedings of the 16th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (pp. 171–180). New York: ACM Press.

Watts, D.J. (2003). Small worlds: The dynamics of networks between order and randomness (Princeton Studies in Complexity). Princeton, NJ: Princeton University Press.

Web Ontology Language (OWL). (2004). Retrieved September 26, 2009, from http://www.w3.org/ 2004/OWL/

Xu, F., Kurz, D., Piskorski, J., & Schmeier, S. (2002). A domain adaptive approach to automatic acquisition of domain relevant terms and their relations with bootstrapping. In Proceedings of the Third International Conference on Language Resources an Evaluation (LREC'02). Paris: European Language Resources Association.

Copyright of Journal of the American Society for Information Science & Technology is the property of John Wiley & Sons, Inc. / Business and its content may not be copied or emailed to multiple sites or posted to a listserv without the copyright holder's express written permission. However, users may print, download, or email articles for individual use.