RC22982 REVISED (W0407-054) July 23, 2004
Computer Science

IBM Research Report

GlossOnt: A Concept-focused Ontology Building Tool

Youngja Park
IBM Research Division
Thomas J. Watson Research Center
P.O. Box 704
Yorktown Heights, NY 10598

Research Division
Almaden - Austin - Beijing - Haifa - India - T. J. Watson - Tokyo - Zurich

GlossOnt: A Concept-focused Ontology Building Tool

Youngja Park
IBM T.J. Watson Research Center
P.O. Box 704, Yorktown Heights
New York 10598, USA
{*young_park*}@*us.ibm.com*

Abstract

The demand for ontologies is rapidly growing especially due to developments in knowledge management, E-commerce and the Semantic Web. Building an ontology and a background knowledge base manually is so costly and time-consuming that it hampers progress in *intelligent information access*. Therefore, semi-automatic or automatic construction of ontologies is very useful.

This paper presents a *concept-focused* ontology building method based on text mining technology. The method focuses on *a particular domain concept at a time* and actively acquires source documents for the ontological knowledge about the concepts. It begins with the analysis of glossary definitions about the target concept by extracting classes and relationships. Then, it generates advanced queries from the result of glossary definition processing and activates a Web search engine to obtain more documents relevant to the target concept. The method extends the ontological knowledge by extracting more domain-specific concepts and relationships from the search documents.

By focusing on a specific concept and highly relevant document set, this method suffers less ambiguity and can identify domain concepts and relations more accurately. In addition, by acquiring source documents from the Web on demand, the method can produce up-to-date ontologies.

1 Introduction

Although ontology has been intensively studied in the past as a discipline of philosophy, research on ontology is recently gaining new attention from the areas of knowledge engineering, intelligent information integration, knowledge management, electronic commerce and the Semantic Web to name a few (Guarino 1998; Fensel 2000).[1] These applications require domain ontologies for searching, retrieving, and integrating information from multiple sources (Berendt, Hotho, & Stumme 2002; Guha, McCool, & Miller 2003).

Ontologies have been manually created by domain experts and/or knowledge engineers, which are specifically designed for given domains and applications. However, manual ontology construction demands a lot of time and effort of human beings that hinders the progress of the above-mentioned activities. For instance, while a few human-made ontologies such as Cyc (Lenat *et al.* 1990) and UMLS (National Library of Medicine) exist, these projects consumed many person-years to come into existence.

Another challenge is keeping an existing ontology, if available, up-to-date. New concepts and new attributes of existing concepts are constantly introduced. For instance, an existing medical ontology might not contain SARS (Severe Acute Respiratory Syndrome), not to mention its relation with Hong Kong or Beijing. Wireless internet access was not available on cellular phones a few years ago, but it is an important feature nowadays. In order to overcome the "knowledge-acquisition bottleneck", automatic or semi-automatic tools for building ontologies are necessary.

With recent advances in the text mining field, attempts at building ontologies automatically or semi-automatically from domain documents have become feasible (Hahn & Schnattinger 1998; Kietz, Volz, & Maedche 2000). However, the technology for automatic ontology construction is still in its infancy and suffers from a few problems.

First, ontologies represent shared conceptualization of the universe or a particular domain. Therefore, building ontologies requires deep human-level language understanding. Most previous approaches rely on shallow language processing (Hahn & Schnattinger 1998; Maedche & Staab 2000b), and thus fail to discover deep-level relationships. Second, recognizing domain concepts and extracting inter-relationships between two concepts from a large document collection are still very unreliable due to the ambiguity in human languages and the problem of data sparseness.

In this paper, we propose a semi-automatic method for building *partial* ontologies, which focuses on a particular domain concept at a time. We call the ontologies partial ontologies because they represent only domain concepts and relationships regarding the target concept. The proposed method takes a target concept from the user. It then actively searches knowledge sources about the target concept, such as domain glossaries and web documents, and extracts from the documents ontological concepts and relationships that are relevant to the target concept. Thus, this system builds a conceptual network for a concept by mining information from relevant texts.

We argue that our approach is more feasible than previous

[1] In this work, an ontology refers to an engineering artifact which describes a hierarchy of concepts related by subsumption relationships (see (Guarino 1998) for more extensive discussions).

methods which try to build a *full* ontology from a collection of documents for the following reasons. First, the system intends to focus on a small number of domain concepts that are described by domain glossaries or by users. Thus, identifying target concepts and relationships in documents can be more focused and thus easier. Second, glossary definitions aim at explicitly defining the target concepts. By processing these glossary definitions, we can obtain more semantic relationships, which are not explicitly expressed in general documents. Third, we have much less semantic ambiguity in glossary definitions and search documents because they are concept-specific documents. This reduced ambiguity makes the approach more practicable. Fourth, we use a state-of-art domain entity recognition algorithm and a deep syntactic parser contrary to previous approaches which use shallow language processing. This enables us to extract domain concepts and relationships more accurately. Lastly, this approach can produce more up-to-date ontologies because the Web reflects creation of new concepts or increased focus on existing concepts more quickly.

2 Concept-focused Ontology Building

Our system, called *GlossOnt*, aims to extract domain concepts and relationships from texts which are very relevant to a target (or anchor) concept. *GlossOnt* uses a search engine to find relevant document sources about the concept of interest. A domain entity recognizer identifies domain-specific glossary items from the document collection. Then, domain concepts that are semantically related to the target concept are selected from the glossary items. A set of relation extractors find relationships in which at least one of the recognized domain concepts participate. The relationships include IS-A, PART-OF, HAS-ALIAS, USE and other relationships expressed by verbs and their arguments.

GlossOnt consists of two phases—domain glossary processing phase and search document processing phase– according to document types.

In the first phase as shown in Figure 1, *GlossOnt* analyzes domain glossaries or dictionaries. A glossary is an alphabetical list of technical terms in a specialized field of knowledge, which usually provides readers with definitions, typical usages and characteristics about domain concepts. Glossaries are very useful domain knowledge resources. For instance, we can find hypernyms and synonyms of terms by understanding their glosses.

In this work, we consider two possible scenarios. First, a user already has domain glossaries in the local computer (1) and *GlossOnt* processes the glosses one after another (i.e., a batch mode). Second, a user provides the system with a domain term of interest, then the system searches glossary definitions for the given term on the Internet (2) (i.e., an online mode). Domain entities are recognized (3) and the relationships are extracted (4) from the glossary definitions.

Domain glossaries generally provide useful knowledge (for example, IS-A and PART-OF relations) of the target concepts. However, they are short in length and thus don't provide complete information regarding the target concept. Thus, we extend our knowledge resource to Web documents or local domain documents.

After processing glossary definitions for a target concept, *GlossOnt* formulates a rich query for the domain concept (5). The advanced query usually consists of the target term, its hypernyms and other semantically relevant terms (in general, they are listed as "see also" in glossary).

We use a search engine with the query to acquire a focused document collection which consists only of documents regarding the target concept. If no glossary definition for a concept is provided or the system fails to find glossary definitions, the system performs Web search (the second step) only with the target term or with a user-provided extended query.

Figure 2 shows the architecture of the second phase of *GlossOnt*. In this phase, *GlossOnt* reads the query and activates a Web search engine to obtain more document sources. Traditional keyword-based search engines may return many irrelevant documents due to semantic ambiguity of query terms and skewed popularity of certain web pages, for example. However, advanced queries augmented with synonyms, hypernyms and other related terms were proven to enhance the search results (Cooper & Byrd 1997; Leroy, Tolle, & Chen 1999; Agirre *et al.* 2000; Kumar *et al.* 2001; AltaVista 2002). Kumar *et al.* (2001) reports that a query consisting of the topic title with query augmentations yields high quality search results with a precision of over 80%. Agirre *et al.* (2000) exploits Web documents to enrich a large lexical database, WordNet (Miller 1990). The system tries to construct lists of closely related words (topic signature) for each sense in WordNet. They build a complicated boolean query from cuewords used in the sense definitions of a word to retrieve documents that contain the target word and one of the cuewords of the target concept.

A query formulated from the result of glossary processing enables the search engine to find a high-quality set of web documents about the target concept. The system selects the top n documents returned by the search engine. The value of n is provided by the user in advance. *GlossOnt* then tries to find domain concepts semantically related to the target query and extracts relationships from the search documents. Section 3 and section 4 present the algorithms for finding domain concepts semantically related to the target concept, and section 5 describes relationship extraction methods in detail.

3 Domain Entity Recognition

Domain entity recognition identifies words or phrases that name and describe domain concepts in documents for a particular domain. In this work, we call these domain-specific terms "glossary items". Concepts are generally represented by nouns and verbs in texts. However, recognizing all nouns and verbs in the given documents is insufficient because domain-specific documents contain a lot of generic terms as well as domain terms.

We developed an automatic glossary extraction system which identifies and organizes glossary items from domain documents (see (Park, Byrd, & Boguraev 2002) for a complete description). The glossary extraction algorithm consists of four steps: candidate glossary item recognition, pre-

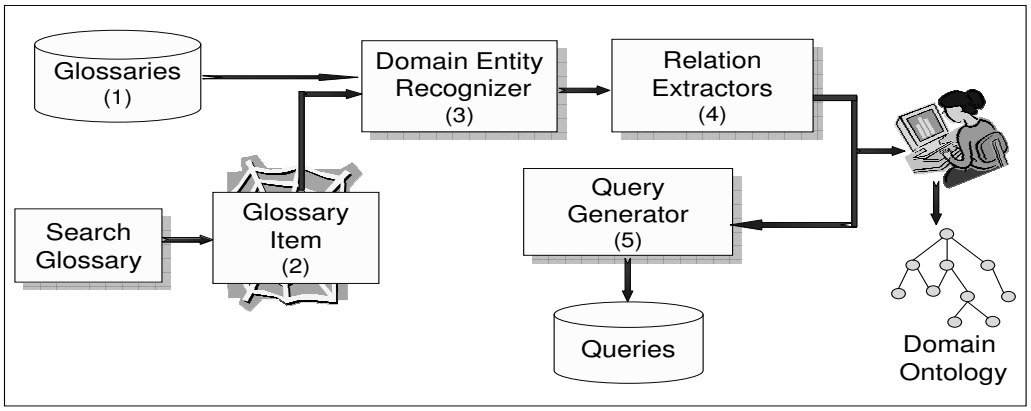

Figure 1: Phase I: *GlossOnt* takes glossary definitions for a target concept from the user's local storage or from the Web. It extracts domain concepts and their relationships to build ontologies. It also produces advanced queries from the result of the analysis of glossary definitions

modifier filtering, glossary item aggregation and confidence value computation.

Candidate Glossary Item Recognition

We first identify single- and multi-word phrases (up to 6 words) by applying a Finite State Transducer (FST)-based phrase recognizer. This FST-based recognition is purely syntactic and is run on part-of-speech (POS) tagged documents. This step recognizes nouns, noun phrases and verbs since they mainly express concepts.

However, the POS-based term recognition method is prone to out-of-vocabulary (OOV) words.[2] Domain documents tend to contain a lot of OOV words because many technical words are not found in general corpora or dictionaries. We process these OOV words and decide if an OOV word is a real word or not on the basis of its morphological constituents and the entropy of the character sequences in the word (see (Park 2002) for detailed description for OOV word processing). If an OOV word is regarded as a real word, and its POS is decided to be a noun or a verb, it is also considered as a candidate glossary item.

Pre-modifier Filtering

Pre-modifier filtering distinguishes domain-specific modifiers from generic modifiers. Many domain-specific noun phrases contain generic modifiers, which do not contribute to the domain concept. For instance, "psychiatric" in "psychiatric disorder" is domain-specific, but "related" in "related disorder" is not domain-specific. There are two problems in including all pre-modifiers in glossary items. First, these pre-modifiers weaken the domain specificity of the glossary items. Second, there may exist many essentially identical domain concepts with slightly different modifiers. This step aims to filter out generic modifiers from glossary items to make the glossary items more coherent and domain-specific.

The filtering decision is based on the modifier's domain specificity and the association of the modifier and its head noun. The domain specificity of a word (w) is calculated by the relative probability of the occurrence of the word in the given domain text (p_d) versus a general corpus (p_c) as follows:

$$domain\_specificity(w) = \frac{p_d(w)}{p_c(w)}$$

The association of an adjective (a) and a noun (n) specifies how likely the head noun is n when a is seen in a glossary item and is calculated by the conditional probability of the occurrence of the head noun given the pre-modifier; that is, $p(n|a)$.

An experiment with a 1-megabyte computer manual shows that the pre-modifier filtering step reduces 142 different kinds of "server"s into 103 "server"s. We think this benefit would be greater when we process larger documents.

Glossary Item Aggregation

Glossary item aggregation recognizes different expressions of a concept and aggregates them into a single glossary item. A single concept may be expressed in different ways in texts, such as abbreviations ("nuclear/biological/chemical" and "nbc"); spelling errors or alternative spellings ("anesthesia" and "anaesthesia"); and orthographic variants ("Attention-Deficit Hyperactivity Disorder", "attention deficit hyperactivity disorder" and "attention deficit/hyperactivity disorder"). Aggregation of different expressions of a concept unifies terminological differences and helps to compute the domain-specificity of a concept more accurately.

Note that discovering some of the variants is not trivial—especially abbreviations and misspellings. We developed a machine-learning approach to recognize abbreviations and

[2] words that are not found in a given dictionary

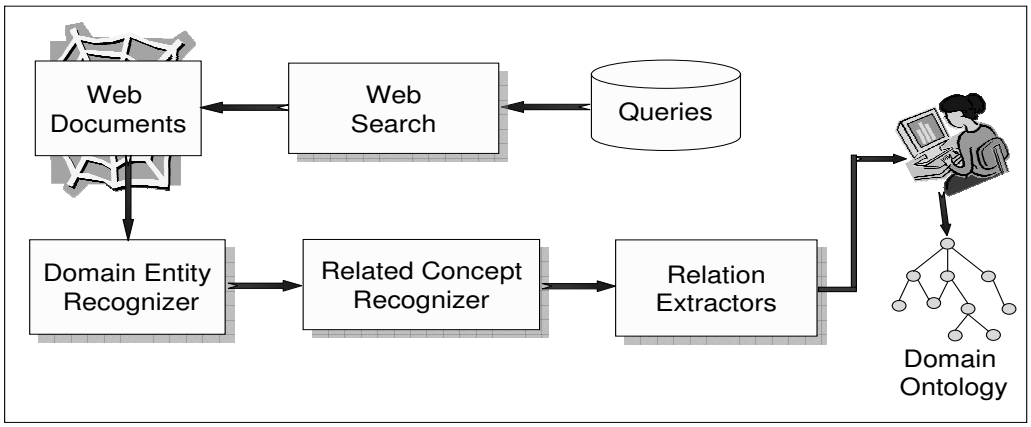

Figure 2: Phase II: *GlossOnt* activates a Web search with query terms generated in Phase I. It first recognizes domain-specific entities in the search documents, and then it extracts other domain concepts that are semantically related to the query terms. Finally, it discovers relationships in which the domain concepts participate

their definitions in documents (Park & Byrd 2001). This algorithm learns *abbreviation pattern-based rules* from many examples of abbreviations and their definitions, and applies the rules to match abbreviations and their full forms in texts. This algorithm also exploits textual cues (for instance, phrases such as "for short", "is acronym of" and patterns such as "multi-word (single word)") to recognize new types of abbreviations which existing *abbreviation pattern rules* fail to match.

Misspellings are recognized as follows. First, if a word is not found in a dictionary, we suppose the word is misspelled. Then, we retrieve words from the dictionary for which string edit distance with the misspelled word is less than or equal to 2. String edit distance (also known as Levenshtein distance) is the smallest number of insertions, deletions, and substitutions required to change one string into another (Levenshtein 1962). If one of these words appears in the document, we consider the target word as a misspelling of the word.

4 Semantically Related Concept Recognition

This section describes how to choose glossary items that are semantically related to a target concept. The domain entity recognition method described in the previous section may produce many candidate glossary items. For concept-focused ontology building, we aim to focus on a small number of domain concepts which are strongly related to each other.

Let the target concept be t, and a candidate glossary item be g. Our hypotheses for finding glossary items semantically relevant to t are as follows:

1. t and g share some words.
   When two glossary items have some words in common, the two glossary items can be regarded as sharing some concept. For instance, if a person is interested in *ballistic missile*, he may also want to know about *short-range ballistic missile* and *medium-range ballistic missile*.
2. t and g appear together in same sentences. If two glossary items co-occur often, they may be a semantic indicator for each other.

Among the glossary items which satisfy the above conditions, we select the final set of domain concepts which have high frequency and domain specificity. Note that frequency and domain specificity are computed on aggregated glossary items. The reason we use domain specificity (a rather complicated measure) as a decision factor is that many high frequency glossary items are generic terms.

We compute the domain-specificity of a glossary item as the average of the domain specificity of all the words in the glossary item:

$$\frac{\sum_{w_i \in T} \log \frac{P_d(w_i)}{P_c(w_i)}}{\mid T \mid} \quad (1)$$

where, $\mid T \mid$ is the number of words in glossary item T, $p_d(w_i)$ is the probability of word w_i in a domain-specific document, and $p_c(w_i)$ is the probability of word w_i in a general document collection.

Table 1 shows the 30 most relevant domain concepts for *agroterrorism* extracted from 100 Web search documents. The search query was created by processing a glossary definition of *agroterrorism* in Table 2.

Note that the first phase of *GlossOnt*, domain glossary processing, does not have this process. We treat all the glossary items in glossary definitions for a concept as semantically relevant to the target concept. This assumption is reasonable as glossary definitions are specially written to define the concept.

The following relation extraction procedure only concentrates on this set of glossary items.

agriculture	agroterrorist attack	al-qaida
animal	Biological Terrorist Attack	attack
bioterrorism	bioterrorist attack	chemical
counter-terrorism	crop	ecoterrorist
event	farm	food
food supply	homeland security	livestock
Terror Attack	Terrorist Threat	terrorism
Terrorist Act	terrorist group	terrorist
Terrorist motive	terrorist organization	threat
Terrorist Incident	United State	weapon

Table 1: Domain concepts relevant to *agroterrorism*. These were extracted from 100 Web documents from a search query "*agroterrorism* and *terrorist attack*".

5 Relation Extraction

The relation extraction module extracts binary relations such as "IS-A", "PART-OF", "HAS-ALIAS", and other verbal relations. Verbal relations mean relations represented by verbs and their arguments. There may be many possible relations in documents, but we only extract the relations where at least one of the arguments is the target concept or a glossary item selected by the process described in the previous section.

We apply a set of different methods for relation extraction according to the type of relations. For instance, we apply several light-weight FST-based pattern recognizers for "IS-A" relation, and a deep syntactic parser for extracting verbal relations. The following sections describe how to extract these relations. Throughout this section, we use a glossary definition in Table 2 as our example text.

agroterrorism .

Terrorist attacks aimed at reducing the food supply by destroying crops using natural pests such as the potato beetle, animal diseases such as hoof and mouth disease and anthrax, molds and other plant diseases, or chemicals that defoliate vegetation, such as Agent Orange.

Table 2: A glossary definition for *agroterrorism*

IS-A Relation

This section describes three different ways for extracting IS-A, or hypernym/hyponym relation. In this work, we don't distinguish "classes" and "instances"; thus, throughout this paper, IS-A relation includes INSTANCE-OF relation. In future work, we will investigate a method for automatically distinguishing classes and instances.

Lexico-syntactically suggested IS-A relations: Some linguistic patterns indicate hypernym/hyponym relations and appear frequently enough to be useful. The first major attempt to automatically extract hyponymy relations of terms from text was that of (Hearst 1992). It presents a set of lexico-syntactic patterns that indicate a hyponymy relation. These patterns occur frequently and in many text genres, and can be recognized with *not-very-complex* text analytics.

Table 3 shows the lexicon-syntactic patterns for IS-A relations. We added more patterns which occur frequently in Web documents in addition to the patterns proposed in (Hearst 1992).

1. such $noun_0$ as $noun_1$, $noun_2$,..., {and|or} $noun_n$
2. $noun_0$ such as $noun_1$, $noun_2$,..., {and|or} $noun_n$
3. $noun_0$ {including|especially} $noun_1$,...,{and|or} $noun_n$
4. $noun_1$, $noun_2$,..., $noun_n$ {and|or} other $noun_0$

5. $noun_1$ is {a|the} {kind | type} of $noun_0$
6. $noun_1$ is a {term|word|concept} {[used] to verb|for verb-ing} $noun_0$
7. $noun_0$ except $noun_1$, $noun_2$,..., {and|or} $noun_n$

Table 3: The lexico-syntactic patterns for hypernym/hyponym relationship. The patterns indicate that $noun_i, 1 \leq i \leq n$, are hyponyms of $noun_0$.

When these syntactic patterns occur in text (either search documents or glossaries), we find that the $noun_i, 1 <= i <= n$, are hyponyms of $noun_0$. In these patterns, note that at least one of $noun_i$ belongs to the set of glossary items semantically related to the target concept.

However, purely pattern-based IS-A relation extraction is not very accurate. Hearst (1992) reports that 52% of relations extracted by the "*and|or other*" pattern were judged to be good when the pattern was extracted from the text of *Grolier's Encyclopedia*. Another evaluation with general texts shows worse performance. Cederberg & Widdows (2003) applies the same set of patterns as in (Hearst 1992) in the British national Corpus (BNC) and reports 40% of the relations extracted were exactly correct, or would be correct with the use of minor post-processing consisting of lemmatization and removal of common types of modifier words. Cederberg & Widdows (2003) suggests a method for filtering out less similar hypernym and hyponym pairs; thus, enhancing precision but reducing recall.

A frequent reason for incorrect hyponymy relations is that the real hypernym appears several words away from the cue phrases. In other words, $noun_0$ in Table 3 is not always the hypernym, but we need to look at wider context to find the correct hypernym. Based on our manual inspection of 2000 sentences, we set the window to 3 noun phrases. For instance, the second pattern in Table 3 extends as follows:

$$noun_{-2},\ \cdots,\ noun_{-1}, \cdots,\ noun_0\ such\ as\ noun_1,\ noun_2, \cdots,\ \{and \mid or\}\ noun_n.$$

More formally, for the extended patterns, we compute the similarity of $noun_i$ (from $i = 1$ to n) and $noun_j, -2 \leq j \leq 0$, and decide $noun_j$ with the highest similarity with $noun_i$ to be the hypernym of $noun_1, \ldots, noun_n$. If the decision is not made, $noun_0$ is selected to be the hypernym.

We compute the similarity of $noun_i$ and $noun_j$, $S(noun_i, noun_j)$, by using a Web search engine. The notion is that if $noun_j$ is the correct hypernym of $noun_i$, then $noun_j$ and $noun_i$ would appear together in those patterns in Table 3 (i.e., local context) more often than other nouns. We can confidently reach this conclusion if we have enough

of these examples to justify it. The largest textual resource is the Web since many Web search engines index billions of web pages. We submit the following three queries to a search engine and choose $noun_j$ with the most search documents returned by the search engine.

$$Q1: \ noun_0 \ \ such \ as \ noun_i$$
$$Q2: \ noun_{-1} \ such \ as \ noun_i$$
$$Q3: \ noun_{-2} \ such \ as \ noun_i$$

From the example gloss in Table 2, the system extracts the following 6 IS-A relationships.

potato beetle	IS-A	*natural pest*
hoof	IS-A	*animal disease*
mouth disease	IS-A	*animal disease*
anthrax	IS-A	*animal disease*
mold	IS-A	*plant disease*
Agent Orange	IS-A	*chemical*

Note that *GlossOnt* identifies *chemical* rather than *vegetation* as the hypernym of *Agent Orange*. However, the system makes a mistake for this example due to syntactic ambiguity in the sentence. The system identifies *hoof* and *mouth disease* as two different domain concepts, but in fact *hoof and mouth disease* is a domain concept.

Structurally suggested IS-A relations: The genus term (i.e., the head noun of the first sentence) of a gloss usually represents the hypernym of the defined term (Vossen, Meijs, & den Broeder 1989). If a given document is a glossary definition, the system processes the first sentence to find the genus term. If there is no other indication for another relation, the genus is determined as the hypernym of the gloss term. For instance, the genus term in the example text is *terrorist attack* and we conclude that *agroterrorism* is a *terrorist attack*.

The common structures in which genus terms are found are:

- $noun_0$ which ....
- A {kind|type|category} of $noun_0$ ...
- A {term|concept} { [used] to verb|for verb-ing} $noun_0$ ...

When these patterns are found in a glossary definition, $noun_0$ is regarded as the hypernym of the target concept. The above patterns are also used for extracting IS-A relations from search documents as explained in the next section.

Lexically suggested IS-A relations: Hyponymy relations can also be found by analyzing multi-word noun phrases. When noun phrases contain one or more modifiers, the modifiers tend to add additional semantic attributes or restrictions to the head noun. Especially, adjectival modifiers usually introduce a hyponymic relation (Bodenreider, Burgun, & Rindflesch 2001).

We treat the head noun and the noun phrase as hypernym and hyponym relation. For instance, we conclude that *psychiatric disorder* IS-A *disorder* and *terrorist attack* IS-A *attack* by applying this heuristic.

5.2 Alias Relation

This relation specifies alternative names for a concept. The HAS-ALIAS relationship in this work is not the same as the synonymy relationship in a strict sense (for instance, "robber" is a synonym of "thief") but it specifies a subset of the synonymy relationship.

Abbreviations are the most common examples of the HAS-ALIAS relation. The system for matching abbreviations and their definitions (Park & Byrd 2001), which was described in section 4, is used for this purpose too. We consider an abbreviation and its definition as aliases for each other. For example, *International Business Machines* HAS-ALIAS *IBM*.

In addition to abbreviations, we apply lexico-syntactic patterns for recognizing other aliases. The patterns are "$noun_1$ ... [also] {known | called } as $noun_2$" as shown in the following examples.

- *Zomig*, formerly known as *311C90*
- *3,4-methylenedioxymethamphetamine* (also known as "*Ecstasy*")

When these patterns appear in text, we find that $noun_1$ and $noun_2$ are aliases. Note that the HAS-ALIAS relation is symmetric.

5.3 Verbal Relations

Verbal relations mean relations represented by verbs and their arguments. Syntactic structures are distinct from semantic structures. Nevertheless, syntactic dependency relations coincide closely with semantic relations between the entities (Maedche & Staab 2000a; Gamallo. *et al.* 2002). In general, lexical items, which express predicate relations, take, as their syntactic dependents, nominal expressions which name the predicate's arguments. These predicate-argument structures can often be interpreted as expressing relationships. A common example is that the subject and the object of a verb are the participants in the relation expressed by the verb.

We find the dependency relations in which the target concept or one of the selected glossary items appear and generate the relations named with the verbs. We process documents with a deep syntactic parser (McCord 1980) to obtain grammatical dependency relations of constituents and to recognize patterns for relations.

In our example gloss, the following 6 verbal relations are recognized:

agroterrorism	REDUCE	*food supply*
agroterrorism	DESTROY	*crop*
agroterrorism	USE	*natural pest*
agroterrorism	USE	*animal disease*
agroterrorism	USE	*plant disease*
agroterrorism	USE	*mold*
agroterrorism	USE	*chemical*

Finally, Figure 3 shows the domain entities and the relationships extracted from the glossary definition in Table 2 by the methods described in section 4 and section 5.

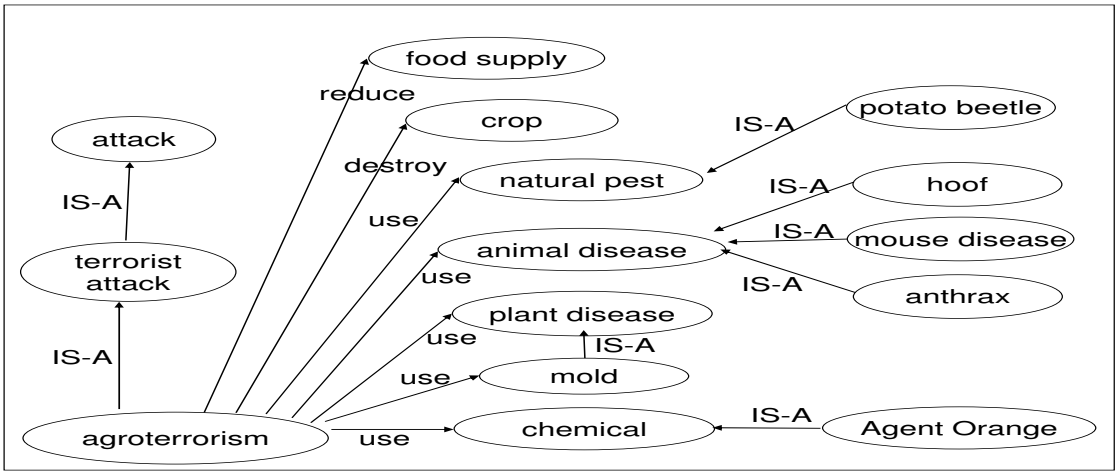

Figure 3: Domain concepts and relations extracted from a glossary definition for *agroterrorism* in Table 2

While these verbal relations represent relationships between domain concepts, it is unclear at this time how these verbal relations would be incorporated in the ontology.

6 Related Work

In this section, we briefly discuss previous efforts to construct or maintain ontologies (semi-)automatically from text.

Hahn & Schnattinger (1998) introduce a methodology for automating the maintenance of domain-specific taxonomies based on natural language text understanding. It starts with an ontology and incrementally updates the ontology as new concepts are acquired from text. Especially, the system records unknown lexical items occurring in certain syntactic constructions (genetive, apposition and verb case frame), and decides if the concept for the new lexical item should be added based on the concept hypotheses. The hypothesis space of possible generalizations for a new unknown concept is initialized as any possible concept in the ontology (e.g., the top concept). When the new concept is seen with more concept hypotheses, the system shrinks that hypothesis space. This system augments an ontology with newly found instances and classes but no attempt has made for extracting relationships.

Kietz, Volz, & Maedche (2000) present a cyclic process for extending an existing ontology with the results of processing a domain-specific dictionary and corporate intranet documents. The system starts with a generic ontology (they actually use a lexical-semantic net called GermaNet). It acquires domain-specific concepts and their taxonomic embedding from domain-specific documents. The newly acquired concepts are added into the generic ontology and all unspecific concepts are removed from the ontology. The system also introduces new conceptual relationships between concepts. This system is similar to the proposed system in this paper in that it also processes a dictionary of corporate terms and applies the lexico-syntactic patterns to find taxonomic relations. However, this system simply converts the headwords in the dictionary into concepts, and does not have an automatic domain concept recognition system from text.

Most previous approaches for automatic ontology construction concentrate only on identifying taxonomic relations (e.g., IS-A relation) by using statistical or linguistic information (Hahn & Schnattinger 1998; Hearst 1992; Pereira, Tishby, & Lee 1993). Maedche & Staab (2000a) presents a method for finding non-taxonomic conceptual relations based on co-occurrence information. Their method finds concept correlations based on co-occurrence statistics of two concepts in domain-specific text. It uses a data mining algorithm for discovering generalized association rules proposed by (Srikant & Agrawal 1995) to find associations between concepts and the appropriate level of abstraction for new relations. However, the relationships remain unnamed; that is, it finds if two concepts are related but it does not tell how they are related to each other. In this work, we discover *named* relationships between domain concepts.

Systems for automatically building semantic lexicons (Roark & Charniak 1998; Riloff & Jones 1999; Thelen & Riloff 2002) don't intend to construct ontologies. However, they are also useful for extending existing ontologies. These systems take as input several seed words for a target concept and use bootstrapping algorithms to find new words which supposedly belong to the same category with the seed words.

7 Conclusion

We presented a semi-automatic tool for building *partial* ontologies from domain glossaries and Web search documents which were obtained on-the-fly. We first run a glossary extraction tool for recognizing domain entities and select glossary items relevant to the target concepts based on co-occurrence information. Then, we apply deep syntactic parsing as well as shallow syntactic pattern matching to extract relationships in which the target concepts and the selected glossary items participate.

This approach differs from other approaches in that it focuses on one domain concept at a time and it obtains re-

lated source documents on demand. Therefore, this concept-focused approach bears several advantages over previous approaches. First, it is a light-weight tool for extracting information for a specific concept, which the user cares about at the time. Second, it produces an up-to-date ontology because it obtains source documents on demand from the Web, which is dynamically changing. Third, it suffers from less semantic ambiguity because the knowledge source is a set of relevant documents to the target concept, which are collected by a search engine with advanced queries.

In future work, we plan to investigate non-verbal binary relations and unary relations or properties. Non-verbal binary relations are relations expressed in noun phrases, for example, *X is the wife of Y* or Y's wife X for "WIFE-OF" OR "SPOUSE-OF" relation. Unary relations or properties of a concept usually specify characteristics of the concept, such as color, length, and so on. These are generally present as modifiers of nouns or as predicates of the verb *to–be*. For instance, from "Al Abbas is a 950 km maximum range ballistic missile"), we want to extract *Al Abbas* HAS-MAXRANGE *950 km* in addition to *Al Abbas* IS-A *ballistic missile*.

Most of the technologies described in this paper have been independently evaluated. However, a remaining challenge is to design and perform evaluations of the system as a whole and the quality of ontologies built by the system.

Finally, an automatic mechanism to identify and resolve contradictory information from different knowledge sources would be desirable. Currently, we entrust domain experts or ontology engineers to resolve the problems.

Acknowledgments

This work is supported by the Advanced Research and Development Activity under the Novel Intelligence and Massive Data (NIMD) program PNWD-SW-6059. I also thank Christopher Welty, Frank Elio and the anonymous reviewers for their comments that much improved the final manuscript.

References

Agirre, E.; Ansa, O.; Hovy, E.; and Martinez, D. 2000. Enriching very large ontologies using the www. In *Proceedings of the Ontology Learning Workshop, ECAI*.

AltaVista. 2002. Altavista prisma. `http://www.altavista.com/prisma`.

Berendt, B.; Hotho, A.; and Stumme, G. 2002. Towards semantic web mining. In *Proceedings of International Semantic Web Conference (ISWC02)*.

Bodenreider, O.; Burgun, A.; and Rindflesch, T. C. 2001. Lexically-suggested hyponymic relations among medical terms and their representation in the umls. In *Proceedings of Terminology and Artificial Intelligence (TIA'2001)*, 11—21.

Cederberg, S., and Widdows, D. 2003. Using lsa and noun coordination information to improve the precision and recall of automatic hyponymy extraction. In *Proceedings of Conference on Natural Language Learning (CoNLL)*, 111–118.

Cooper, J., and Byrd, R. J. 1997. Lexical navigation: visually prompted query expansion and refinement. In *Proceedings of the second ACM international conference on Digital libraries*, 237–246.

Fensel, D. 2000. *Ontologies: Silver Bullet for Knowledge Management and Electronic Commerce*. Springer-Verlag.

Gamallo., P.; Gonzalez, M.; Agustini, A.; p. Lopes, G.; and de Lima, V. S. 2002. Mapping syntactic dependencies into semantic relations. In *Workshop OLT'02 (ECAI'02)*, 15–22.

Guarino, N. 1998. Formal ontology and information systems. In *Proceedings of FOIS'98*, 3–15.

Guha, R.; McCool, R.; and Miller, E. 2003. Semantic search. In *Proceedings of WWW2003*.

Hahn, U., and Schnattinger, K. 1998. Ontology engineering via text understanding. In *Proceedings of the 15th World Computer Congress*.

Hearst, M. 1992. Automatic acquisition of hypernyms from large text corpora. In *Proceedings of the Fourteenth International Conference on Computational Linguistics*.

Kietz, J.-U.; Volz, R.; and Maedche, A. 2000. Extracting a domain-specific ontology from a corporate intranet. In *Proceedings of the Fourth Conference on Computational Natural Language Learning and of the Second Learning Language in Logic Workshop*.

Kumar, R.; Raghavan, P.; Rajagopalan, S.; and Tomkins, A. 2001. On semi-automated web taxonomy construction. In *Fourth International Workshop on the Web and Databases (WebDB'2001)*.

Lenat, D.; Guha, R. V.; Pittman, K.; Pratt, D.; and Shepherd, M. 1990. Cyc: Toward programs with common sense. *Communications of the ACM* 33(8):30–49.

Leroy, G.; Tolle, K. M.; and Chen, H. 1999. Customizable and ontology-enhanced medical information retrieval interfaces. Proceedings of IMIA Working Group 6 Medical Concept Representation and Natural Language Processing.

Levenshtein, V. 1962. Binary codes capable of correcting deletions, insertions and reversals. *Soviet Physics Doklady* 10(8):707–710.

Maedche, A., and Staab, S. 2000a. Discovering conceptual relations from text. In *Proceedings of ECAI*.

Maedche, A., and Staab, S. 2000b. Semi-automatic engineering of ontologies from text. In *Proceedings of Software Engineering and Knowledge Engineering*.

McCord, M. C. 1980. Slot grammars. *American Journal of Computational Linguistics* (1):255–286.

Miller, G. 1990. Wordnet: an on-line lexical database. *International Journal of Lexicography* 3(4).

National Library of Medicine. Unified medical language system (umls). `http://www.nlm.nih.gov/research/umls`.

Park, Y., and Byrd, R. J. 2001. Hybrid text mining for matching abbreviations and their definitions. In *Proceedings of Empirical Methods in Natural Language Processing*, 126–133.

Park, Y.; Byrd, R. J.; and Boguraev, B. K. 2002. Automatic glossary extraction: Beyond terminology identification. In *Proceedings of the Nineteenth International Conference on Computational Linguistics*.

Park, Y. 2002. Identification of probable real words: An entropy-based approach. In *Proceedings of ACL Workshop on Unsupervised Lexical Acquisition*, 1–8.

Pereira, F. C. N.; Tishby, N.; and Lee, L. 1993. Distributional clustering of english words. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 183–190.

Riloff, E., and Jones, R. 1999. Learning dictionaries for information extraction by multi-level bootstrapping. In *Proceedings of the Sixteenth National Conference on Artificial Intelligence*.

Roark, B., and Charniak, E. 1998. Noun-phrase co-occurrence statistics for semi-automatic semantic lexicon construction. In *Proceedings of COLING-ACL'98*.

Srikant, R., and Agrawal, R. 1995. Mining generalized association rules. In *Proceedings of VLDB'95*, 407–419.

Thelen, M., and Riloff, E. 2002. A bootstrapping method for learning semantic lexicons using extraction pattern contexts. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing*.

Vossen, P.; Meijs, W.; and den Broeder, M. 1989. Meaning and structure in dictionary definitions. In Boguraev, B., and Briscoe, T., eds., *Computational Lexicography for Natural Language Processing*, 171–192. Longman.