Improving an Ontology Refinement Method with Hyponymy Patterns

Enrique Alfonseca* **and Suresh Manandhar†**

* Departamento de Ingeniería Informática, Universidad Autónoma de Madrid
28049 Madrid, Spain
Enrique.Alfonseca@ii.uam.es

† Computer Science Department, University of York
York YO10 5DD, United Kingdom
suresh@cs.york.ac.uk

Abstract

We describe here a procedure to combine two different existing techniques for Ontology Enrichment with domain-specific concepts. The resulting algorithm is fully unsupervised, and the level of precision is higher than when they are used separately, so we believe that both algorithms benefit from each other. The experiments have been performed by extending WordNet with concepts extracted from *The Lord of the Rings*.

1. Introduction

Lexical semantic ontologies, such as WordNet (Miller, 1995), have proven very useful with many applications in Natural Language Applications. However, they usually only include general terms, as it would be impossible to extend them with every concept used in every domain of knowledge, and there are few automatic methods for enriching them with domain-specific concepts, a subtask of what Maedche and Staab (2001) call *Ontology Refinement* (OR). In a previous paper (Alfonseca and Manandhar, 2002) we describe an unsupervised algorithm for enriching an ontology such as WordNet with concepts extracted from particular domains. Our method was a deterministic top-down algorithm that proceeded down the taxonomy, selecting at each level the concept that is distributionally most similar to the unknown concept.

We present in this paper a way to combine our OR system with the method described by Hearst (1992), consisting in looking for patterns inside texts from which we can extract information about how to extend an ontology. In our initial experiments, the accuracy of both algorithms have increased, which indicates that both of them can benefit from each other.

1.1. Related work

Because a comprehensive review of learning applied to ontologies is beyond the scope of this paper, we shall focus some systems for OR on lexical ontologies that have influenced most on our research.

One of the most widely used lexical ontologies is WordNet (Miller, 1995), in which concepts (also called synsets, sets of synonym words) are structured through the *hypernymy* relationship from the most general to the most specific. If a concept c subsumes a concept d, we say that c is a hypernym of d, and that d is a hyponym of c.

One of the first attempts to extend WordNet with domain-specific information was reported by O'Sullivan et al. (1995), who added new synsets about word processors and software applications, although the work was all done by hand by domain experts.

Concerning automatic systems for enriching existing ontologies with new concepts, two very similar systems were reported by Hastings (1994) and Hahn and Schnattinger (1998). Both of these systems start with an initial ontology of nouns and verbs, a set of domain-dependent texts, and restrictions about the selectional preferences of the verbs, e.g. that the object of *arson* is known to be a *building* and the object of *kill* is known to be a *person*. At the beginning, the hypothesis space of possible generalisations for a new, unknown concept is initialised as any possible concept in the ontology. When the new concept is seen in the text as subject or object of the verbs, the selectional restrictions are used to shrink that hypothesis space. The more times a concept appears in the text, the more information the system has to classify it in the ontology. Hastings (1994) worked on the terrorist domain, while Hahn and Schnattinger (1998) did his experiments with texts from an I.T. magazine and an ontology about electronics.

A different approach was described by Hearst (1992), and used again by Kietz et al. (2000) for his OR system. In this approach, the aim is to find regular expression patterns from free texts by looking at pairs of (hypernym, hyponym) that co-occur in the same sentence, and then to use them to learn new hypernymy relations. For example, from sentence (1), taken from (Hearst, 1998), a system can discover that the pattern `such NPs as {NP,}* NP` usually states a hypernymy relation, if *Herrick*, *Goldsmith* and *Shakespeare* appear as hyponyms of *author* in the initial ontology. That pattern can be used later to learn relationships between new concepts. We call these patterns *hyponymy patterns*.

(1) ...works by such authors as Herrick, Goldsmith and Shakespeare...

Kietz et al. (2000) applied hand-coded patterns for extending GermaNet (a German equivalent of WordNet) with concepts from a corporate intranet, and quantified the error rate of this procedure in 32%. As described by him, this procedure has several drawbacks:

- The list of patterns was compiled by hand.

attach(u)
u is the unknown synset,

1. Let r be the root synset in the ontology.
2. return analyseLevel(u, r)

analyseLevel(u, s)
s is the candidate synset most similar to u.

1. Get s's synset children, $\{s_1, s_2, ..., s_n\}$.
2. Calculate $d_s \leftarrow distance(u, s)$
3. For every child s_i, calculate $d_{s_i} \leftarrow distance(u, s_i)$
4. Find the concept whose semantic distance to u is the lowest
    4.1 If that concept is s, return s
    4.2 Otherwise, if that concept is s_i, return analyseLevel(u, s_i)

Figure 1: Pseudo-code of the original algorithm for finding the correct place where the unknown synset u will be attached in the ontology

- If a concept never appears inside one of these patterns, the system cannot classify it.
- The error rate is high, so it is necessary that a user validates the program's output.

1.2. Structure of this document

In next section, we describe our system before we extended it with Hearst-like hyponymy patterns. Next, in section 3 we describe the way in which we have combined our previous system with the patterns. Finally, sections 4 and 5 contain the results of our preliminary experiments and our conclusions.

2. Description of the previous system

The aim of our system is the correct classification of unknown concepts in the WordNet lexical ontology.

For finding which is the correct place where a new unknown synset u should be attached to the ontology, we have devised an algorithm that performs a top-down search, and it stops at the synset that is most similar to u. The procedure is detailed in Figure 1. The search starts at the most general synset s, and compares u with it and with all of its immediate hyponyms. If s is more similar to u than any of s's children, then u is assumed to be a hyponym of s. Otherwise, we proceed downwards along the most similar child found.

The semantic distance used is based on the Distributional Semantics model, which assumes that there is a strong correlation between the semantics of a word and the set of contexts in which that word appears Rajman and Bonnet (1992). This idea motivated the use of topic signatures, that have been applied to text summarisation (Lin, 1997) and word-sense disambiguation (Yarowsky, 1992) (Agirre et al., 2000). A topic signature (Yarowsky, 1992) of a word w is the list of the words that co-occur with it, together with their respective frequencies or weights. Because WordNet does not include topic signatures we used the method proposed by Agirre et al. (2000) to acquire them, in an unsupervised way, from Internet.

First decision: entity		
synset	**synset Id**	$total$
being, organism	n00002908	0.3207
causal agency	n00004753	0.3121
location	n00018241	0.1383
body of water	n07411542	0.1112
thing (anything)	n03781420	0.0457
thing (object)	n00002254	0.0442
cell	n00004081	0.0087
(15 more)	...	...
Second decision: being		
synset	**synset Id**	$total$
human	n00005145	0.6161
animal	n00010787	0.2790
host	n01015823	0.0243
parasite	n01015154	0.0192
flora	n00011740	0.0169
(34 more)	...	...

Table 1: Similarity values for each of the decisions that have been taken when classifying the unknown concept *orc*. In the first place, when deciding between *entity* and its children, the chosen one was *being, life form*. In the second decision, the chosen synset was *human*. Both were correct.

These signatures can be used, at each iteration of our top-down algorithm to decide which is the synset most similar to the unknown concept u (step 4).

For example, table 1 show how the concept *orc*, which appears in *The Lord of the Rings* (Tolkien, 1968) but not in WordNet, was classified. The root of the hierarchy where it was classified is the synset *entity*, so the first decision consisted in choosing the synset, amongst *entity* and its hyponyms, that had the most similar context to *orc*, in the test set. There were two synsets with the maximal value: *being* and *causal agency*, both of which are hypernyms of *human*.

In the second decision, when deciding between *being* and its children synsets, the chosen synset was *human*, with a high degree of confidence. The context words for *animal* were also found somewhat similar to those of *orc*, and the rest of the synsets had much lower values.

3. Learning hyponymy patterns

3.1. Motivation

If we examine the mistakes committed by the previous algorithm, we find that it is difficult for it to distinguish between concepts that can appear in similar contexts. For example, the topic signatures of *adult_male* and *adult_female*, in WordNet, are very similar, and many mistakes were due to people classified in the wrong sex. Due to the same reason, when processing excerpts from *The Lord of the Rings* (Tolkien, 1968), all *hobbits*[1] were classified as *male* men.

The approach taken by Hearst (1998), by looking for hyponymy patterns and then extracting the hyponymy relationships can help improve this weak point in our algorithm because, when the extracted relationship is correct, it is usually relevant. However, as he notes, the hyponymy patterns used to find new hypernymy relationships can generate a large number of mistakes, either because the extracted relation is far too general (e.g. `hypernym(exercise, thing)`); because they are subjective opinions with little interest (e.g. `hypernym(Gaslight, classic)`, referring to the film *Gaslight*); or because of parsing errors.

Our new approach proposes the following:

- The use of the hypernymy patterns only as a support for the top-down classifier for making the decisions, when the topic signature gives a similar weight to several concepts.
- The automatic extraction of a different set of hypernymy patterns for every level of the WordNet hierarchy.

3.2. Automatic extraction of hypernymy patterns

As (Hearst, 1998) proposes, hypernymy patterns can be extracted automatically from texts by looking at sentences that contain a pair $(hypernym, hyponym)$ from WordNet. We have defined First Order Predicate Logic (FOPL) predicates to represent several kinds of syntactic dependencies, and we extract the dependencies between the hypernym and the hyponym. The following are some of the rules out system generated:

To obtain the hyponymy patterns that apply to each WordNet synset, we followed the following steps:

- For each WordNet synset, a query is automatically constructed for the Altavista Internet search engine, following the procedure detailed in (Agirre et al., 2000), and a set of documents is collected that contain the words in that synset.
- The documents are processed with a Flex tokeniser, a sentence splitter, the TnT part-of-speech tagger (Brants, 2000), a Flex stemmer, and a transformation-list Noun Phrase chunker (Ramshaw and Marcus, 1995) written in Java.
- The sentences from those documents that contain both any of the synset words and any of its hypernym's words were selected.
- The system extract the hyponymy patterns from them, using the FOPL predicates, and pruned the low-frequency ones.

For example, the following are some of the patterns that were extracted from the texts. The first one shows the case in which the verb *to be* functions as a copula; the second and the third phrases show appositive constructions; and the last case shows how a prepositional phrase can indicate a hypernymy relationship.

(1) `Shakespeare was a first-class poet`
hypernym(N2, N1) :- subject(N1, be), object(N2, be)
(2) `Shakespeare, the poet, ...`
hypernym(N2, N1) :- appositive(N2, N1)
(3) `The English dramatist, Shakespeare, ...`
hypernym(N2, N1) :- appositive(N1, N2)
(4) `...the city of Seville...`
hypernym(N2, N1) :- pp_modifier(of, N1, N2).

These patterns are extracted at each level of the WordNet hierarchy, from documents downloaded from Internet corresponding to nearly one thousand of the WordNet synsets. From our experiments we observe that some rules such as (1), (2) and (3) are general and appear at every level, but rule (4) applies only in a few cases, specially for geographic regions such as *city*, *kingdom* or *valley*.

3.3. Modifications to the original algorithm

The top-down algorithm is modified so, at each level, if one of the possible decision synsets has one descendant in the ontology which had been suggested by the patterns, the support for choosing that synset is multiplied by a factor which decreases with the depth of that descendant. For example, in the classification for *orc* shown in table 1, if the patterns had suggested that *orc* could be a hyponym of *animal*, its weight would have been multiplied by 5, and *animal* would have been the decision taken; if they had suggested that *orc* could be a hyponym of *domestic animal* (a child of animal), it would have been multiplied by 2.5; and so on.

In this way, we fulfil a double objective:

1. The directed search of the top-down algorithm helps in that most of the erroneous hypernyms suggested by mistakes of the patterns are never considered, because the search does not proceed near them.
2. The hypernyms suggested by the patterns help the top-down algorithm when the decision is difficult because two concepts appear in very similar contexts, such as the male-female distinction.

4. Experiments and Results

We have worked with the WordNet taxonomy that is rooted on the node *entity*, and which includes locations, people, life beings, and artifacts, amongst others.

[1] Hobbits are a race of small people

We have evaluated the algorithm using three metrics: (1) the percentage of unknown concepts that were **finally** attached to one of the correct hypernyms, i.e. the overall accuracy; (2) the percentage of times that the correct synset was chosen, at each iteration of the top-down search; (3) The average position that the correct synset ranked in those decisions.

We also used a fourth metric, called **Learning Accuracy** (Hahn and Schnattinger, 1998), that takes into consideration the distance, in the ontology, between the place where the new concept should have been classified and the place where the algorithm placed it. Let us suppose that the target answer for classifying the unknown concept u_i is s_i, and the system returns instead the concept f_i. Let us call c_i the lowest concept that is a hypernym of both s_i and f_i. If we call CP_i, SP_i and FP_i the lengths of the shortest paths from the top of the hierarchy to c_i, s_i and f_i, respectively; and DP_i the distance between c_i and f_i, then the Learning Accuracy for u_i is

$$LA_i = \begin{cases} \frac{CP_i}{SP_i} & \text{if } f_i = c_i \\ \frac{CP_i}{FP_i} & \text{if } f_i = s_i \\ \frac{CP_i}{FP_i+DP_i} & \text{otherwise} \end{cases} \tag{1}$$

The overall learning accuracy is the mean of the computed values:

$$LA = \sum_{i \epsilon \{1 \ldots n\}} \frac{LA_i}{n} \tag{2}$$

If the output is correct, Learning Accuracy will have a value of 1.Figure 2 (a) and (b) show the value of the learning accuracy in two different cases.

Because WordNet is not a tree, i.e. a synset can have more than one hypernym, it may be the case that there are several ways to calculate Learning Accuracy, such as that in Figure 2 (c). We have redefined LA as the maximum of all of them, which corresponds to the shortest path between s_i and f_i. Therefore, LA in the example displayed would be 0.6.

In our preliminary experiments, we have classified 42 concepts taken from *The Lord of the Rings* (Tolkien, 1968).

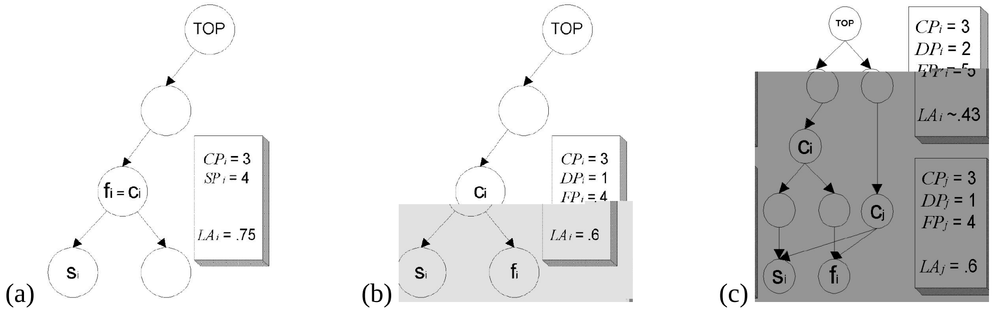

Figure 2: Learning accuracy in three different cases. (a) When the proposed concept is correct, but too general. (b) When the proposed concept is incorrect. (c) When there are different ways to compute Learning Accuracy.

5. Conclusions

We describe here a way to improve our unsupervised *Ontology Refinement* algorithm by finding *hypernymy patterns* in domain-specific texts. The integration of the two different algorithms produces a more robust classification algorithm.

The top-down classifier, based on the context words, suggests a path from the root of the ontology down to the concept that will be suggested as the maximally specific generalisation of an unknown concept. The patterns help this algorithm in selecting a concept when the context does not give much information, such as for male-female distinctions.

The result is a deterministic unsupervised system that also allows the attachment of new concepts to any intermediate level in an ontology, not only at the leaves. We have shown that it is able to tackle big ontologies with the size of WordNet.

Because it does not require any previous hand-coding of patterns, and the concept contexts are also automatically collected from the Internet, we believe it could be ported to other languages, if the syntactic processing tools used are available for them.

6. Acknowledgements

This work has been partially sponsored by CICYT, project number TIC2001-0685-C02-01.

7. References

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

E. Alfonseca and S. Manandhar. 2002. An unsupervised method for ontology refinement. In *Poceedings of the First International Conference on General WordNet*, Mysore, India, january.

T. Brants. 2000. *TnT - A Statistical Part-of-Speech Tagger*. User manual.

Udo Hahn and Klemens Schnattinger. 1998. Towards text knowledge engineering. In *AAAI/IAAI*, pages 524–531.

P. M. Hastings. 1994. *Automatic acquisition of word meaning from context*. University of Michigan, Dissertation.

M. A. Hearst. 1992. Automatic acquisition of hyponyms from large text corpora. In *Proceedings of COLING-92*, Nantes, France.

M. A. Hearst, 1998. *Automated Discovery of WordNet Relations. In (Christiane Fellbaum (Ed.) WordNet: An Electronic Lexical Database*, pages 132–152. MIT Press.

J. Kietz, A. Maedche, and R. Volz. 2000. A method for semi-automatic ontology acquisition from a corporate intranet. In *Workshop "Ontologies and text", co-located with EKAW'2000*, Juan-les-Pins, French Riviera.

Chin-Yew Lin. 1997. *Robust Automated Topic Identification*. Ph.D. Thesis. University of Southern California.

A. Maedche and S. Staab. 2001. Ontology learning for the semantic web. *IEEE Intelligent systems*, 16(2).

George A. Miller. 1995. WordNet: A lexical database for English. *Communications of the ACM*, 38(11):39–41.

D. O'Sullivan, A. McElligott, and R. F. E. Sutcliffe. 1995. Augmenting the princeton wordnet with a domain specific ontology. In *Proceedings of the Workshop on Basic Issues in Knowledge Sharing at the 14th International Joint Conference on Artificial Intelligence*, Montreal.

M. Rajman and A. Bonnet. 1992. Corpora-based linguistics: new tools for natural language processing. In *1st Annual Conference of the Association for Global Strategic Information*, Germany. Bad Kreuznach.

L. A. Ramshaw and M. P. Marcus. 1995. Text chunking using transformation-based learning. In *Third ACL Workshop on Very Large Corpora*, pages 82–94. Kluwer.

J. R. R. Tolkien. 1968. *The Lord of the Rings*. Allen and Unwin.

David Yarowsky. 1992. Word-sense disambiguation using statistical models of roget's categories trained on large corpora. In *Proceedings of COLING-92*, pages 454–460, Nantes, France.