Automatic extraction of semantic relationships for WordNet by means of pattern learning from Wikipedia*

Maria Ruiz-Casado, Enrique Alfonseca and Pablo Castells

Computer Science Dep., Universidad Autonoma de Madrid, 28049 Madrid, Spain
{Maria.Ruiz,Enrique.Alfonseca,Pablo.Castells}@uam.es

Abstract. This paper describes an automatic approach to identify lexical patterns which represent semantic relationships between concepts, from an on-line encyclopedia. Next, these patterns can be applied to extend existing ontologies or semantic networks with new relations. The experiments have been performed with the Simple English Wikipedia and WordNet 1.7. A new algorithm has been devised for automatically generalising the lexical patterns found in the encyclopedia entries. We have found general patterns for the hyperonymy, hyponymy, holonymy and meronymy relations and, using them, we have extracted more than 1200 new relationships that did not appear in WordNet originally. The precision of these relationships ranges between 0.61 and 0.69, depending on the relation.

1 Introduction

The exponential growth of the amount of data in the World Wide Web (WWW) requires the automatising of processes like searching, retrieving and maintaining information. One of the difficulties that prevents the complete automatising of those processes [1] is the fact that the contents in the WWW are presented mainly in natural language, whose ambiguities are hard to be processed by a machine.

The Semantic Web (SW) constitutes an initiative to extend the web with machine readable contents and automated services far beyond current capabilities [2]. A common practise is the annotation of the contents of web pages using an ontology. One of the most accepted definitions of an ontology is "an explicit specification of a conceptualisation" [3]. In most of the cases, ontologies are structured as hierarchies of concepts, by means of the relation called hyponymy (*is-a*, class inclusion or subsumption) and its inverse hyperonymy, which arranges the concepts from the most general to the most specific one. Additionally, there may be other relationships, such as meronymy (the part-whole relation) and its inverse holonymy; telicity (*purpose*), or any other which may be of interest, such as *is-author-of*, *is-the-capital-of*, *is-employee-of*, etc. In many cases, ontologies distinguish nodes that represent concepts (classes of things, e.g. *person*) from nodes that represent instances (examples of concepts, e.g. *John*) [4].

Like the web itself, sometimes, these ontologies have to include a high amount of information, or they undergo a rapid evolution. Therefore, it would be also highly desirable to automatise or semi-automatise the acquisition of the ontologies as well.

* This work has been sponsored by CICYT, project numbers TIC2002-01948 and TIN2004-03140.

This problem has been object of recent increasing interest, and new approaches for automatic ontology enrichment and population are being developed, which combine resources and techniques from Natural Language Processing, Information Extraction, Machine Learning and Text Mining [5, 6].

In this paper, we present a procedure for automatically enriching an existing lexical semantic network with new relationships extracted from on-line encyclopedic information. The semantic network chosen is WordNet [7], given that it is currently used in many applications, although the procedure is general enough to be used with other ontologies. The encyclopedia used is the Wikipedia, a collaborative web-based resource which is being constantly updated by its users. In this experiments, we have worked with its Simple English version[1], because the vocabulary and syntactic structures found in Simple English are easier to handle by a parser than those in fully unrestricted text. In addition, the fact that it is written with less supervision than an academic encyclopedia means that the language used is freer, sometimes colloquial, and the techniques that work well here are expected to be easier to port to the web than if we worked with a more structured reference text.

This paper is structured in the following way: Section 2 describes related work; Sections 3 and 4 detail the approach followed, and the evaluation performed; finally, Section 5 concludes and points out open lines for future work.

2 Related work

Automatic extraction of information from textual corpora is now a well-known field with many different applications. Concerning automatic ontology enrichment, we may classify current approaches in the following groups:

- Systems based on distributional properties of words: it consists in studying co-occurrence distributions of terms in order to calculate a semantic distance between the concepts represented by those terms. This distance metric can next be used for conceptual clustering [8, 9], Formal Concept Analysis [10] or for classifying words inside existing ontologies [11–14]. The previous are usually applied to enrich the ontologies with new concepts. On the other hand, [15] learn association rules from dependency relations between terms which, combined with heuristics, are used to extract non-taxonomic relations.
- Systems based on pattern extraction and matching: these rely on lexical or lexicosemantic patterns to discover ontological and non-taxonomic relationships between concepts in unrestricted text. [16–18] manually define regular expressions to extract hyponymy and part-of relationships. [19] learns such patterns for company merge relationships. [20] quantifies the error rate of a similar approach as 32%. [21] describes a combination of a pattern-based and a distributional-based approach, also for hyperonymy. [22] describe a whole framework which incorporates terminology extraction and ontology construction and pruning which takes into account, amongst other things, substring relationships for identifying hyperonyms.

[1] http://simple.wikipedia.org/wiki/Main_Page

– Systems based on dictionary definitions analysis [23–25], take advantage of the particular structure of dictionaries in order to extract hyperonymy relationships with which to arrange the concepts in an ontology. Concept definitions and glosses have been found very useful, as they are usually concise descriptions of the concepts and include the most salient information about them [26]. There are also several works which extract additional relationships from WordNet glosses, by disambiguating the words in the glosses [26–29].

3 Procedure

The procedure followed consists in crawling the Simple English version of Wikipedia, collecting all the entries, disambiguating them, and associating each other with relations. The steps carried out, similar to those described in [16, 19], are the following:

1. *Entry Sense Disambiguation*: This step consists in preprocessing the Wikipedia definitions and associating each Wikipedia entry to its corresponding WordNet synset, so the sense of the entry is explicitly determined.
2. *Pattern extraction*: For each entry, the definition is processed looking for words that are connected with the entry in Wikipedia by means of a hyperlink. If there is a relation in WordNet between the entry and any of those words, the context is analysed and a pattern is extracted for that relation.
3. *Pattern generalisation*: In this step, the patterns extracted in the previous step are compared with each other, and those that are found to be similar are automatically generalised.
4. *Identification of new relations*: the patterns are applied to discover new relations other than those already present in WordNet.

The following sections detail all the steps in the procedure:

3.1 Entry sense disambiguation

The goal of this step is to mark each entry in the Wikipedia with its corresponding synset in WordNet. To this aim, the entries are downloaded, and they are processed in the following way:

1. Those web pages which contain more than one definition are divided in separate files.
2. Most of the HTML tags are removed.
3. The definitions are processed with a sentence splitter, a part-of.speech-tagger and a stemmer [30].
4. For each entry, choose the WordNet synset whose sense is nearer according to the definition.

The disambiguation procedure, described in detail in [31], is mainly based on the Vector Space Model and the dot-product similarity metric, co-occurrence information and some heuristics. Approximately one third of the entries in Wikipedia are not found in WordNet, one third appear with just one sense (they are monosemous), and one third have multiple possible senses (they are polysemous).

The output of this pre-processing step is a list of Wikipedia disambiguated entries.

3.2 Pattern extraction

In the previous step, every entry from the encyclopedia has been disambiguated using WordNet as the sense dictionary. The aim of this step is the extraction of patterns relating two concepts such that they have already been disambiguated and they share a relation in WordNet. The process is the following:

1. For each term t in the Wikipedia, with a definition d, we select every term f such that there is a hyperlink within d pointing to f. This assures that f's entry also exists in Wikipedia, and its sense has been disambiguated in the previous step.

   The reason why we only select the terms which have an entry in Wikipedia is that we have obtained a higher accuracy disambiguating the entry terms than attempting a disambiguation of every word inside the definitions. In this way, we expect the patterns to be much more accurate.

   If a particular entry is not found in the disambiguated set, it is ignored, because it means that either the entry is not yet defined in the Wikipedia[2], or it was not found in WordNet and was not disambiguated previously.
2. Once we have found a hyperlink to other disambiguated entry, the following process is carried out:
   (a) Look up in WordNet relationships between the two terms.
   (b) If any relation is found, collect the sentence where the hyperlink appears (with part-of-speech tags).
   (c) Replace the hyperlink by the keyword TARGET.
   (d) If the entry term appears in the sentence, replace it by the keyword ENTRY.

This work uses WordNet 1.7, in which there are six possible relationships between nouns. The first four, hyperonymy, hyponymy, holonymy and meronymy have been included in this study. Concerning antonymy, this relationship in WordNet does not always refer to the same feature, as sometimes it relates nouns that differ in gender (e.g. *king* and *queen*), and, other times, in a different characteristic (e.g. *software* and *hardware*), so it would be very difficult to find a consistent set of patterns for it. With respect to synonymy, we found that there are very few sentences in Wikipedia that contain two synonyms together, as they are expected to be known by the reader and they are used indistinctly inside the entries.

For illustration, if the entry for *Lisbon* contains the sentence *Lisbon is part of Portugal*, the pattern produced would be the following: `ENTRY is/VBZ part/NN of/IN TARGET`. Note that the words are annotated with part-of-speech tags, using the labels defined for the Penn Treebank[32].

The output of this step consists of as many lists as relationships under study, each list containing patterns that are expected to model each particular relation for diverse pairs of words.

[2] The Wikipedia is continuously refreshing its contents and growing, and some of the links of the definitions fail to bring to another definition.

3.3 Pattern generalisation (I): Edit distance calculation

In order to generalise two patterns, the general idea is to look for the similarities between them, and to remove all those things that they do not have in common.

The procedure used to obtain a similarity metric between two patterns, consists of a slightly modified version of the dynamic programming algorithm for *edit-distance* calculation [33]. The *edit distance* between two strings A and B is defined as the minimum number of changes (character insertion, addition or replacement) that have to be done to the first string in order to obtain the second one. The algorithm can be implemented as filling in a matrix $\mathcal{M}$ with the following procedure:

$$\mathcal{M}[0,0] = 0 \tag{1a}$$

$$\mathcal{M}[i,0] = M[\ i-1,0] + 1 \tag{1b}$$

$$\mathcal{M}[0,j] = M[0,j-1] + 1 \tag{1c}$$

$$\begin{aligned}\mathcal{M}[i,j] = min(&\mathcal{M}[i-1,j-1] + d(A[i],B[j]),\\ &\mathcal{M}[i-1,j] + 1,\\ &\mathcal{M}[i,j-1] + 1)\end{aligned} \tag{1d}$$

where $i\epsilon[1...|A|], j\epsilon[1...|B|]$
and

$$d(A[i],B[j]) = \begin{cases} 0 & \text{if } A[i] = B[j] \\ 1 & \text{otherwise} \end{cases}$$

In these equations, M[i,j] will contain the edit distance between the first i elements of A and the first j elements of B. Equation (1a) indicates that, if A and B are both empty strings, the edit distance should be 0. Equations (1b) and (1c) mean that the edit distance between an empty string, and a string with N symbols must be N. Finally, equation (1d) uses the fact that, in order to obtain a string[3] $A\sigma$ from a string $B\gamma$, we may proceed in three possible choices:

- We may obtain $A\gamma$ from $B\gamma$, and next substitute γ by σ. If γ and σ are the same, no edition will be required.
- We may obtain $A\sigma\gamma$ from $B\gamma$, and next delete γ at the end.
- We may obtain A from $B\gamma$, and next insert the symbol σ in the end.

In the end, the value at the rightmost lower position of the matrix is the edit distance between both strings. The same algorithm can be implemented for word patterns, if we consider that the basic element of each pattern is not a character but a whole token.

At the same time, while filling matrix $\mathcal{M}$, it is possible to fill in another matrix $\mathcal{D}$, in which we record which of the choices was selected as minimum in equation (1d). This can be used afterwards in order to have in mind which were the characters that both strings had in common, and in which places it was necessary to add, remove or replace characters. We have used the following four characters:

[3] $A\sigma$ represents the concatenation of string A with character σ.

A: `It is a kind of`
B: `It is nice of`

$\mathcal{M}$	**0**	**1**	**2**	**3**	**4**
0	0	1	2	3	4
1	1	0	1	2	3
2	2	1	0	1	2
3	3	2	1	1	2
4	4	3	2	2	2
5	5	4	3	3	2

$\mathcal{D}$	**0**	**1**	**2**	**3**	**4**
0		I	I	I	I
1	R	E	I	I	I
2	R	R	E	I	I
3	R	R	R	U	I
4	R	R	R	R	U
5	R	R	R	R	E

Fig. 1. Example of the edit distance algorithm. A and B are two word patterns; $\mathcal{M}$ is the matrix in which the edit distance is calculated, and $\mathcal{D}$ is the matrix indicating the choice that produced the minimal distance for each cell in $\mathcal{M}$.

- `I` means that it is necessary to insert a token, in order to transform the first string into the second one.
- `R` means that it is necessary to remove a token.
- `E` means that the corresponding tokens are equal, so it is not necessary to edit them.
- `U` means that the corresponding tokens are unequal, so it is necessary to replace one by the other.

Figure 1 shows an example for two patterns, A and B, containing respectively 5 and 4 tokens. The first row and the first column in $\mathcal{M}$ would be filled during the initialisation, using Formulae (1b) and (1c). The corresponding cells in matrix $\mathcal{D}$ are filled in the following way: the first row is all filled with `I`'s, indicating that it is necessary to insert tokens to transform an empty string into B; and the first column is all filled with `R`'s indicating that it is necessary to remove tokens to transform A into an empty string. Next, the remaining cells would be filled by the algorithm, looking, at each step, which is the choice that minimises the edit distance. $\mathcal{M}(5,4)$ has the value 2, indicating the distance between the two complete patterns. For instance, the two editions would be replacing `a` by `nice`, and removing `kind`.

3.4 Pattern generalisation (II): Algorithm

After calculating the edit distance between two patterns A and B, we can use matrix $\mathcal{D}$ to obtain a generalised pattern, which should maintain the common tokens shared by them. The procedure used is the following:

1. Initialise the generalised pattern G as the empty string.
2. Start at the last cell of the matrix $\mathcal{M}(i,j)$. In the example, it would be $\mathcal{M}(5,4)$.
3. While we have not arrived to $\mathcal{M}(0,0)$,
   (a) If $(\mathcal{D}(i,j) = \texttt{E})$, then the two patterns contained the same token A[i]=B[j].
      – Set $G = A[i]\ G$
      – Decrement both i and j.
   (b) If $(\mathcal{D}(i,j) = \texttt{U})$, then the two patterns contained a different token.
      – $G = A[i]|B[j]\ G$, where `|` represents a disjunction of both terms.
      – Decrement both i and j.

(c) If ($\mathcal{D}(i,j) = \texttt{R}$), then the first pattern contained tokens not present in the other.
  - Set $G = * G$, where $*$ represents any sequence of terms.
  - Decrement i.

(d) If ($\mathcal{D}(i,j) = \texttt{I}$), then the second pattern contained tokens not present in the other.
  - Set $G = * G$
  - Decrement j

If the algorithm is followed, the patterns in the example will produced the generalised pattern

```
It is a kind   of
It is nice     of
-------------------
It is a|nice * of
```

This pattern may match phrases such as *It is a kind of, It is nice of, It is a hyperonym of,* or *It is a type of.* As can be seen, the generalisation of these two rules produces one that can match a wide variety of sentences, and which may be indicating different kinds of relationships between concepts.

3.5 Pattern generalisation (III): Generalisation with part-of-speech tags

The previous example shows that, when two patterns are combined, sometimes the result of the generalisation is far too general, and matches a wide variety of sentences that don't share the same meaning. Therefore, in order to restrict the kinds of patterns that can combine to produce a generalisation, the algorithm has been extended to handle part-of-speech tags. Now, a pattern will be a sequence of terms, and each term will be annotated with a part-of-speech tag, as in the following examples:

(a) `It/PRP is/VBZ a/DT kind/NN of/IN`
(b) `It/PRP is/VBZ nice/JJ of/IN`
(c) `It/PRP is/VBZ the/DT type/NN of/IN`

The edit distance algorithm is modified in the following way: the system only allows replacement actions if the words from the two patterns A and B belong to the same general part-of-speech (nouns, verbs, adjectives, adverbs, etc.). Also, if this is the case, we consider that there is no edit distance between the two patterns. In this way, two patterns that do not differ in the part-of-speech of any of their words will be considered more similar than other pairs of patterns differing in the part-of-speech of one word. The d function, therefore, is redefined as:

$$d(A[i], B[j]) = \begin{cases} 0 & \text{if } PoS(A[i]) = PoS(B[j]) \\ 1 & \text{otherwise} \end{cases} \quad (2)$$

The insertion and deletion actions are defined as before. Therefore, patterns (a) and (b) above would have an edit distance of 2, and the result of their generalisation is:

`It/PRP is/VBZ * of/IN`

On the other hand, the patterns (a) and (c) would have an edit distance of 0, and the result of their generalisation would be the following:

```
It/PRP is/VBZ a|the/DT kind|type/NN of/IN
```

Once the generalisation procedure has been defined, the following algorithm is used in order to generate the final set of generalised patterns:

1. Collect all the patterns from the Wikipedia entries in a set $\mathcal{P}$.
2. For each possible pair of patterns, calculate the edit distance between them.
3. Take the two patterns with the smallest edit distance, p_i and p_j.
4. If the edit distance between them exceeds a threshold θ, stop.
5. Otherwise,
   (a) Remove them from $\mathcal{P}$.
   (b) Calculate the more general pattern p_g from them.
   (c) Add p_g to $\mathcal{P}$.
6. Go back to step 2.

The previous algorithm is repeated for each relationship (e.g. hyponymy or meronymy). The output of the algorithm is the set containing all the rules that have been obtained by combining pairs of original rules. The purpose of the parameter θ is the following: if we set no limit to the algorithm, ultimately all the rules can be generalised to a single generalisation containing just one asterisk, which would match any text. Thus, it is desirable to stop merging rules when the outcome of the merge is too general and would be source of a large quantity of errors. The value of θ was set empirically to 5. For higher values of θ, the system tried to generalise very different rules, resulting in rules with many asterisks and few lexical terms.

3.6 Identification of new relations

Finally, given a set of patterns for a particular relation, they can be applied to all the entries in the Wikipedia corpus. Whenever a pattern matches, the target word is identified, and a candidate relationship is produced.

In this step, we took into account the fact that most relations of holonymy and meronymy are either between instances or between concepts, but not between an instance and a concept. For instance, it is correct to say that *Lisbon* is part of *Portugal*, but it does not sound correct to say that *Lisbon* is part of the concept *country*, even though Portugal is a country. Therefore, all the results obtained for holonymy and meronymy in which one of the two concepts related was an instance and the other was a concept were removed from the results. We have used the classification of WordNet synsets as instances or concepts provided by [34].

The output of this step is a list of extracted related pairs of entries for each relation.

4 Evaluation and Results

The algorithm has been evaluated with the whole Simple English Wikipedia entries, as available on November 15, 2004. Each of the entries was disambiguated using the procedure described in [31]. An evaluation of 360 entries, performed by two human

judges, indicates that the precision of the disambiguation is 92% (87% for polysemous words). The high figure should not come as a surprise, given that, as can be expected, it is an easier problem to disambiguate the title of an encyclopedia entry (for which there exist much relevant data) than a word inside unrestricted text.

The next step consisted in extracting, from each Wikipedia entry e, a list of sentences containing references to other entries f which are related with e inside WordNet. This resulted in 270 sentences for hyponymy, 158 for hyperonymy, 247 for holonymy and 222 for meronymy. When analysing these patterns, however, we found that, both for hyperonymy and meronymy, most of the sentences extracted only contained the name of the entry f (the target of the relationship) with no contextual information around it. The reason was unveiled by examining the web pages:

- In the case of hyponyms and holonyms, it is very common to express the relationship with natural language, with expressions such as *A dog is a mammal*, or *A wheel is part of a car*.
- On the other hand, when describing hyperonyms and meronyms, their hyponyms and holonyms are usually expressed with enumerations, which tend to be formatted as HTML bullet lists. Therefore, the sentence splitter chunks each hyponym and each holonym as belonging to a separate sentence.

Extraction of hyponymy relations A total of 1204 relations of hyponymy have been automatically extracted from the Wikipedia entries using the patterns that were found in the previous step (excluding repetitions). 352 out of them already appeared in WordNet (including those which appear through transitive closure), and the remaining 852 relations were evaluated by two human judges. Inter-judge agreement was very high for the case of hyponymy, reaching 99%. The overall precision is 0.69.

Table 1 shows some of the rules extracted, which were evaluated separately. The rule that applied most often is the classical hyponymy copular expression, `ENTRY is a TARGET`, which relates a concept with its hyperonym. There are five versions of this rule (numbered 1, 2, 3, 4 and 6) in the Table, allowing for extra tokens before and in between, and providing a long list of adjectives that may appear in the definition. Many of the errors produced by these patterns can be explained, given that, for sequences such as *the man with the telescope is the leader* in the corpus, the word *telescope* would be chosen as hyponym of *leader*, because the patterns do not have syntactic information.

Secondly, there are also patterns which have been extracted because of the characteristics of Wikipedia. For instance, there are several entries about months in the years, and all of them contain a variant of the sentence `XXX is the` n`th month in the year`. Therefore, rule 5 shows a pattern extracted from those sentences. Other example is that of colours, and all of which contain the same sentence, `List of colors`, in their definition.

Finally, rule 23 has been displayed as an example of a too specific rule that, because it is not very general, has not been able to identify any hyponymy relationship apart from those that were already in WordNet. This rule has been created, mostly, from definitions of planets in the Solar System (e.g. *Venus is the second planet from the Sun*) and from definitions of months in the year (e.g. *March is the third month in the Year*). When

No.	Match	Prec.	Rule
1	2	1.0	ENTRY/NN is/VBZ a/DT type/NN of/IN TARGET
2	1	1.0	ENTRY/NNP (/(*/* :/, Jawa|Kernow/NNP)/) is/VBZ */* a|the/DT TARGET in|of/IN England|Indonesia/NNP
3	139	0.86	*/* is|was/VBD */* British|English|Greek|alcoholic|baked| deadly|non-metal|old|oldest/JJ TARGET
4	370	0.7	ENTRY/NN is/VBZ a/DT TARGET
5	23	0.57	TARGET of/IN the/DT Year/NN
6	18	0.44	*/* ENTRY/NN is/VBZ a/DT TARGET founded|used/VBN */*
...			(up to 22 rules)
23	0	N/A	*/* ENTRY|Isotopes|Jupiter|Mercury|Neptune|Saturn|Uranus|Venus/NNP are|is/VBZ */* big|common|different|eighth| fifth|first|largest|nearest|ninth|second|seventh|sixth|small |third/JJ TARGET from|in|of|on/IN */* Earth|Ocean|Sun|days|earth|element|sun|year|years/NNP

Table 1. Some of the rules obtained for the relation of hyponymy. Columns indicate the number of the rules, the new results produced by each rule, its precision and the text of the rule.

matched with the Wikipedia entries, it is just able to extract the known relationships for instances of planets and instances of months, so it does not generate new knowledge.

Extraction of hyperonymy relations Concerning hyperonymy, as commented before, there were very few patterns to use, and they were very specific. Just four patterns were matched in the texts, resulting in four already known relationships.

Extraction of holonymy relations Twenty rules for identification of holonymy relations matched the texts in 418 places. 115 were already present in WordNet (including transitive closure), and the remaining 303 were evaluated by two judges. In this case, inter-judge agreement reached 95%. In order to unify the criteria, in the doubtful cases, similar cases were looked inside WordNet, and the judges tried to apply the same criteria as shown by those examples. The final precision for these patterns is 0.61.

Table 2 shows some of the rules for holonymy. Most of the *member part-of* and *substance part-of* relations were rightly extracted by rules 4, 5 and 6, which match sentences such as *X is in Y* or *X is a part of Y*. However, they also produced some wrong relations. Many patterns focused on locations, such as rules 1, 2, 3, 4, 6 and 7.

In the case of holonymy, an important source of errors was the lack of a multi-word expression recogniser. Many of the part-of relations that appear in Wikipedia are relations between instances, and a large portion of them have multi-word names. For instance, the application of the set of patterns to the sentence

Oahu is the third largest of the Hawaiian Islands

returns the relation *Oahu is part of Islands*, because *Hawaiian_Islands* has not been previously identified as a multi-word named entity.

Other errors were due to orthographic errors in the Wikipedia entry (e.g. *Lourve* instead of *Louvre*) and relations of holonymy which held in the past, but which are not true by now, such as *New York City is part of Holland* or *Caribbean Sea is part of Spain*.

No.	Match	Prec.	Rule
1	10	1.00	`ENTRY/NNP is/VBZ a/DT city|province/NN in/IN TARGET`
2	5	1.00	`ENTRY/NNP */* the/DT */* capital|city/NN */* capital|city/NN of/IN TARGET`
3	1	1.00	`*/* is|was/VBZ one/CD of/IN the/DT */* States|countries/NNPS in/IN the/DT TARGET`
4	25	0.84	`*/* ENTRY/NNP is/VBZ */* a|the/DT */* Lakes|Republic|canal|capital|city|coast|country|northeast|province|region|southwest|state|west/NN in|of/IN TARGET`
5	120	0.59	`ENTRY/NNP is/VBZ a|an|the/DT */* in|of/IN the/DT TARGET`
6	97	0.49	`*/* Things|city|member|north|part|planets|state/NNS in|of/IN the/DT TARGET`
7	4	0.75	`*/* ENTRY/NNP is|was/VBZ a/DT */* country|part|river/NN in|of/IN */* eastern|north|northern/JJ TARGET`
...			(up to 20 rules)

Table 2. Rules obtained for the relation of holonymy, ordered by precision. Columns indicate the rules' number, number of new results found, precision and pattern.

Extraction of meronymy relations Concerning the last kind of relationship studied, meronymy, 184 new relations were found, out of which 115 already were known, 42 were judged correct, and 27 were judged wrong, which results in an overall precision of 0.61. As is the case with hyperonymy, the number of patterns and relations extracted is much lower than for their inverse relations.

5 Conclusions and future work

This work addresses the problem of automatically identifying semantic relationships in free text. Some of the conclusions that can be drawn from this work are the following:

- A new algorithm for generalising lexical patterns has been described, implemented and evaluated. It is based on the edit distance algorithm, which has been modified to take into account the part-of-speech tags of the words. This algorithm is fully automatic, as it requires no human supervision.
- The set of patterns which has been found automatically from the Wikipedia entries, is able to extract new relations from text for each of the four relationships: hyperonymy, hyponymy, meronymy and holonymy. More than 1200 new relationships have been provided.
- The precision of the generated patterns is similar to that of patterns written *by hand* (although they are not comparable, as the experimental settings differ). The kind of hyponymy lexicosyntactic patterns as described by [16] were evaluated, in different settings, by [20] and [10], who report a precision of 0.65 and 0.39, respectively. [18] reports a 0.55 accuracy for a set of patterns that identify holonyms. Only [19] reports much higher accuracies (0.72, 0.92 and 0.93), when identifying relationships of merging between companies.

This work opens the following research lines: (a) to extract other kinds of relations, such as *location*, *instrument*, *telic* or *author*; (b) to generalise the experiment to other

ontologies and encyclopedias, and even to apply it to fully unrestricted texts; and (c) to extend the formalism used to represent the patterns, so they can encode syntactic features as well.

References

1. Ding, Y., Fensel, D., Klein, M.C.A., Omelayenko, B.: The semantic web: yet another hip? Data Knowledge Engineering **41** (2002) 205–227
2. Berners-Lee, T., Hendler, J., Lassila, O.: The semantic web - a new form of web content that is meaningful to computers will unleash a revolution of new possibilities. Scientific American **284** (2001) 34–43
3. Gruber, T.R.: A translation approach to portable ontologies. Knowledge Acquisition **5** (1993) 199–220
4. Degen, W., Heller, B., Herre, H., Smith, B.: Gol: Towards an axiomatized upper-level ontology. In: Proceedings of the International Conference on Formal Ontology in Information Systems, FOIS-2001. (2001)
5. Gómez-Pérez, A., Macho, D.M., Alfonseca, E., nez, R.N., Blascoe, I., Staab, S., Corcho, O., Ding, Y., Paralic, J., Troncy, R.: Ontoweb deliverable 1.5: A survey of ontology learning methods and techniques (2003)
6. Maedche, A., Staab, S.: Ontology learning for the semantic web. IEEE Intelligent systems **16** (2001)
7. Miller, G.A.: WordNet: A lexical database for English. Communications of the ACM **38** (1995) 39–41
8. Lee, L.: Similarity-Based Approaches to Natural Language Processing. Ph.D. thesis. Harvard University Technical Report TR-11-97 (1997)
9. Faure, D., Nédellec, C.: A corpus-based conceptual clustering method for verb frames and ontology acquisition. In: LREC workshop on Adapting lexical and corpus resources to sublanguages and applications, Granada, Spain (1998)
10. Cimiano, P., Staab, S.: Clustering concept hierarchies from text. In: Proceedings of LREC-2004. (2004)
11. Hastings, P.M.: Automatic acquisition of word meaning from context. University of Michigan, Ph. D. Dissertation (1994)
12. Hahn, U., Schnattinger, K.: Towards text knowledge engineering. In: AAAI/IAAI. (1998) 524–531
13. Pekar, V., Staab, S.: Word classification based on combined measures of distributional and semantic similarity. In: Proceedings of Research Notes of the 10th Conference of the European Chapter of the Association for Computational Linguistics, Budapest (2003)
14. Alfonseca, E., Manandhar, S.: Extending a lexical ontology by a combination of distributional semantics signatures. In: Knowledge Engineering and Knowledge Management. Volume 2473 of Lecture Notes in Artificial Intelligence. Springer Verlag (2002) 1–7
15. Maedche, A., Staab, S.: Discovering conceptual relations from text. In: Proceedings of the 14th European Conference on Artifial Intelligence. (2000)
16. Hearst, M.A.: Automatic acquisition of hyponyms from large text corpora. In: Proceedings of COLING-92, Nantes, France (1992)
17. Hearst, M.A. In: Automated Discovery of WordNet Relations. In Christiane Fellbaum (Ed.) WordNet: An Electronic Lexical Database. MIT Press (1998) 132–152
18. Berland, M., Charniak, E.: Finding parts in very large corpora. In: Proceedings of ACL-99. (1999)

19. Finkelstein-Landau, M., Morin, E.: Extracting semantic relationships between terms: supervised vs. unsupervised methods. In: Proceedings of the International Workshop on Ontological Engineering on the Global Information Infrastructure. (1999)
20. Kietz, J., Maedche, A., Volz, R.: 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 (2000)
21. Alfonseca, E., Manandhar, S.: Improving an ontology refinement method with hyponymy patterns. In: Language Resources and Evaluation (LREC-2002), Las Palmas (2002)
22. Navigli, R., Velardi, P.: Learning domain ontologies from document warehouses and dedicated websites. Computational Linguistics **30** (2004)
23. Wilks, Y., Fass, D.C., Guo, C.M., McDonald, J.E., Plate, T., Slator, B.M.: Providing machine tractable dictionary tools. Journal of Computers and Translation (1990)
24. Rigau, G.: Automatic Acquisition of Lexical Knowledge from MRDs. PhD Thesis, Departament de Llenguatges i Sistemes Informàtics, Universitat Politècnica de Catalunya (1998)
25. Richardson, S.D., Dolan, W.B., Vanderwende, L.: MindNet: acquiring and structuring semantic information from text. In: Proceedings of COLING-ACL'98. Volume 2., Montreal, Canada (1998) 1098–1102
26. Harabagiu, S., Moldovan, D.I.: Knowledge processing on an extended wordnet. In: WordNet: An Electronic Lexical Database. MIT Press (1998) 379–405
27. Harabagiu, S., Miller, G., Moldovan, D.: Wordnet 2 - a morphologically and semantically enhanced resource. In: Proc. of the SIGLEX Workshop on Multilingual Lexicons, ACL Annual Meeting, University of Maryland (1999)
28. Novischi, A.: Accurate semantic annotation via pattern matching. In: Proceedings of FLAIRS-2002. (2002)
29. DeBoni, M., Manandhar, S.: Automated discovery of telic relations for wordnet. In: Poceedings of the First International Conference on General WordNet, Mysore, India (2002)
30. Alfonseca, E.: Wraetlic user guide version 1.0 (2003)
31. Ruiz-Casado, M., Alfonseca, E., Castells, P.: Automatic assignment of wikipedia encyclopedic entries to wordnet synsets. In: press. (2005)
32. Marcus, M.P., Santorini, B., Marcinkiewicz, M.A.: Building a large annotated corpus of english: the penn treebank. Computational Linguistics **19** (1993) 313–330
33. Wagner, R., Fischer, M.: The string-to-string correction problem. Journal of Assoc. Comput. Mach. **21** (1974)
34. Alfonseca, E., Manandhar, S.: Distinguishing instances and concepts in wordnet. In: Poceedings of the First International Conference on General WordNet, Mysore, India (2002)