

Learning Semantic Constraints for the Automatic Discovery of Part-Whole Relations

Proceedings of HLT-NAACL 2003
Main Papers, pp. 1-8
Edmonton, May-June 2003

Roxana Girju

Human Language Technology
Research Institute,
University of Texas at Dallas
and
Computer Science Department,
Baylor University
girju@ecs.baylor.edu

Adriana Badulescu and Dan Moldovan

Human Language Technology
Research Institute,
University of Texas at Dallas
adriana@hlt.utdallas.edu,
moldovan@utdallas.edu

Abstract

The discovery of semantic relations from text becomes increasingly important for applications such as Question Answering, Information Extraction, Text Summarization, Text Understanding, and others. The semantic relations are detected by checking selectional constraints. This paper presents a method and its results for learning semantic constraints to detect *part-whole* relations. Twenty constraints were found. Their validity was tested on a 10,000 sentence corpus, and the targeted part-whole relations were detected with an accuracy of 83%.

1 Introduction

1.1 Problem description

An important semantic relation for several NLP applications is the *part-whole* relation, or *meronymy*. Consider the text:

The car's mail messenger is busy at work in the mail car as the train moves along. Through the open side door of the car, moving scenery can be seen. The worker is alarmed when he hears an unusual sound. He peeks through the door's keyhole leading to the tender and locomotive cab and sees the two bandits trying to break through the express car door.

There are six part-whole relations in this text: 1) the *mail car* is part of the *train*, 2) the *side door* is part of the *car*, 3) the *keyhole* is part of the *door*, 4) the *cab* is part of the *locomotive*, 5) the *door* is part of the *car*, and 6) the *car* is part of the *express train* (the last two in the compound noun *express car door*).

Understanding part-whole relations allows Question Answering systems to address questions such as “*What are the components of X?*”, “*What is X made of?*” and others. Question Answering, Information Extraction and Text Summarization systems often need to identify relations between entities as well as synthesize information gathered from multiple documents. More and more knowledge intensive techniques are used to augment statistical methods when building advanced NLP applications.

This paper provides a method for deriving semantic constraints necessary to discover part-whole relations.

1.2 Semantics of part-whole relation

There are different ways in which we refer to something as being a part of something else, and this led many researchers to claim that *meronymy* is a complex relation that “should be treated as a collection of relations, not as a single relation” (Iris et al., 1988).

Based on linguistic and cognitive considerations about the way parts contribute to the structure of the wholes, Winston, Chaffin and Hermann (Winston et al., 1987) determined in 1987 six types of part-whole relations: Component-Integral object (*wheel - car*), Member-Collection (*soldier - army*), Portion-Mass (*meter - kilometer*), Stuff-Object (*alcohol - wine*), Feature-Activity (*paying - shopping*), and Place-Area (*oasis - desert*).

The part-whole relations in WordNet are classified into three basic types: Member-of (e.g., *UK IS-MEMBER-OF NATO*), Stuff-of (e.g., *carbon IS-STUFF-OF coal*), and all other part-whole relations grouped under the general name of Part-of (e.g., *leg IS-PART-OF table*). In this paper we lump together all the part-whole relation types, but if necessary, one can train the system separately on each of the six meronymy types to increase the learning accuracy.

1.3 Previous work

Although part-whole relations were studied by philosophers, logicians, psychologists and linguists, not much work has been done to automatically identify the

meronymy relation in text. Hearst (Hearst, 1998) developed a method for the automatic acquisition of hypernymy relations by identifying a set of frequently used and unambiguous lexico-syntactic patterns. Then, she tried applying the same method to meronymy, but without much success, as the patterns detected also expressed other semantic relations.

In 1999, Berland and Charniak (Charniak, 1999) applied statistical methods on a very large corpus to find part-whole relations. Using Hearst's method, they focused on a small set of lexico-syntactic patterns that frequently refer to meronymy and a list of 6 seeds representing whole objects. Their system's output was an ordered list of possible parts according to some statistical metrics. The accuracy obtained for the first 50 parts was 55%.

2 Lexico-syntactic patterns expressing meronymy

2.1 Variety of meronymy expressions

Since there are many ways in which something can be part of something else, there is a variety of lexico-syntactic structures that can express the meronymy semantic relation. Expressions that reflect semantic relations are either explicit or implicit. The explicit ones are further broken down into unambiguous and ambiguous.

A. Explicit part-whole constructions

There are *unambiguous* lexical expressions that always convey a part-whole relation. For example:

The substance *consists of* two ingredients.

The cloud was *made of* dust.

Iceland is a *member of* NATO.

In these cases the simple detection of the patterns leads to the discovery of part-whole relations.

On the other hand, there are many *ambiguous* expressions that are explicit but convey part-whole relations only in some contexts. These expressions can be detected only with complex semantic constraints.

Examples are:

The horn is *part of* the car.

(whereas ``He is part of the game'' is not meronymic).

B. Implicit part-whole constructions

In addition to the explicit patterns, there are other patterns that express part-whole relations implicitly. Examples are: girl's mouth, eyes of the baby, door knob, oxygen-rich water, high heel shoes.

2.2 An algorithm for finding lexico-syntactic patterns

In order to identify lexical forms that express part-whole relations, the following algorithm was used:

Step 1. Pick pairs of WordNet concepts C_i , C_j among which there is a part-whole relation.

We selected 100 pairs of *part-whole* concepts that were evenly distributed over all nine WordNet noun hierarchies.

Step 2. Extract lexico-syntactic patterns that link the two selected concepts of each pair by searching a collection of texts.

For each pair of part-whole concepts determined above, search a collection of documents and retain only the sentences containing that pair. We chose two distinct text collections: SemCor 1.7 and LA Times from TREC-9. From each collection 10,000 sentences were selected randomly. We manually inspected these sentences and picked only those in which the pairs referred to meronymy.

The result of this step is a list of lexico-syntactic expressions that reflect meronymy. From syntactic point of view, these patterns can be classified in two major categories:

- *Phrase-level patterns*, where the part and whole concepts are included in the same phrase. For example, in the pattern " $NP_X PP_Y$ " the noun phrase that contains the *part* (X) and the prepositional phrase that contains the *whole* (Y) form a noun phrase (NP). Throughout this paper, X represents the part, and Y represents the whole.
- *Sentence-level patterns*, where the part-whole relation is intrasentential. A frequent example is the pattern " $NP_Y \text{ verb } NP_X$ ".

From the 20,000 SemCor and LA Times sentences, 535 part-whole occurrences were detected. Of these 493 (92.15%) were phrase-level patterns and only 42 sentence-level patterns. There were 54 distinct meronymic lexico-syntactic patterns, of which 36 phrase-level patterns and 18 sentence-level patterns. The most frequent phrase-level patterns were:

" $NP_1 \text{ of } NP_2$ " occurring 173 of 493 times or 35%;

" NP_1 's NP_2 " occurring 71 of 493 times or 14%;

The most frequent sentence-level pattern was

" $NP_1 \text{ Verb } NP_2$ " occurring 18 of 42 times (43%).

These observations are consistent with the results in (Evens et al. , 1980). Based on these statistics, we decided to focus in this paper only on the three patterns

above. The problem, however, is that these are some of the most ambiguous part-whole relation patterns. For example, in addition to meronymic relations, the genitives can express POSSESSION (*Mary's toy*), KINSHIP (*Mary's brother*), and many other relations. The same is true for "*NP₁ Verb NP₂*" patterns ("*Kate has green eyes*" is meronymic, while "*Kate has a cat*" is POSSESSION).

As it can be seen, the genitives and the have-verb patterns are ambiguous. Thus we need some semantic constraints to differentiate the part-whole relations from the other possible meanings these patterns may have.

3 Learning Semantic Constraints

3.1 Approach

The learning procedure proposed here is *supervised*, for the learning algorithm is provided with a set of inputs along with the corresponding set of correct outputs. Based on a set of positive and negative meronymic training examples provided and annotated by the user, the algorithm creates a decision tree and a set of rules that classify new data. The rules produce constraints on the noun constituents of the lexical patterns.

For the discovery of the semantic constraints we used C4.5 *decision tree learning* (Quinlan, 1993). The learned function is represented by a decision tree, or a set of if-then rules. The decision tree learning searches a complete hypothesis space from simple to complex hypotheses until it finds a hypothesis consistent with the data. Its bias is a preference for the shorter tree that places high information gain attributes closer to the root. The error in the training examples can be overcome by using different training and a test corpora, or by cross-validation techniques.

C4.5 receives in general two input files, the NAMES file defining the names of the attributes, attribute values and classes, and the DATA file containing the examples. The output of C4.5 consists of two types of files, the TREE file containing the decision tree and some statistics, and the RULES file containing the rules extracted from the decision tree and some statistics for training and test data. This last file also contains a default rule that is usually used to classify unseen instances when no other rule applies.

3.2 Preprocessing Part-Whole Lexico-Syntactic Patterns

Since our constraint learning procedure is based on the semantic information provided by WordNet, we need to preprocess the noun phrases (NPs) extracted and identify the *part* and the *whole*. For each NP we keep only the largest word sequence (from left to right) that is defined in WordNet as a concept.

For example, from the noun phrase "*brown carving*

knife" the procedure retains only "*carving knife*", as it is the WordNet concept with the largest number of words in the noun phrase. For each such concept, we manually annotate it with its corresponding sense in WordNet, for example *carving knife*#1 means sense number 1.

3.3 Building the Training Corpus and the Test Corpus

In order to learn the constraints, we used the SemCor 1.7 and TREC 9 text collections. From the first two sets of the SemCor collection, 19,000 sentences were selected. Another 100,000 sentences were extracted from the LA Times articles of TREC 9. A corpus "A" was thus created from the selected sentences of each text collection. Each sentence in this corpus was then parsed using the syntactic parser developed by Charniak (Charniak, 2000).

Focusing only on the sentences containing relations indicated by the three patterns considered, we manually annotated all the noun phrases in the 53,944 relationships matched by these patterns with their corresponding senses in WordNet (with the exception of those from SemCor). 6,973 of these relationships were part-whole relations, while 46,971 were not meronymic relations.

We used for training a corpus of 34,609 positive examples (6,973 pairs of NPs in a part-whole relation extracted from the corpus "A" and 27,636 extracted from WordNet as selected pairs) and 46,971 negative examples (the non-part-whole relations extracted from corpus "A").

3.4 Learning Algorithm

Input : positive and negative meronymic examples of pairs of concepts.

Output : semantic constraints on concepts.

Step 1. Generalize the training examples

Initially, the training corpus consists of examples that have the following format:

```
<part#sense; whole#sense; target>,
```

where *target* can be either "Yes" or "No", as the relation between the *part* and *whole* is meronymy or not. For example, <oasis#1; desert#1; Yes> indicates that between oasis and desert there is a meronymic relation.

From this initial set of examples an intermediate corpus was created by expanding each example using the following format:

```
<part#sense, class_part#sense;
whole#sense, class_whole#sense;
target> ,
```

where *class_part* and *class_whole* correspond to the WordNet semantic classes of the *part*, respectively *whole* concepts. For instance, the initial example <aria#1; opera#1; Yes> becomes <aria#1, entity#1; opera#1,

abstraction#6; Yes> , as the *part* concept *aria#1* belongs to the *entity#1* hierarchy in WordNet and the *whole* concept *opera#1* is part of the *abstraction#6* hierarchy.

From this intermediate corpus a generalized set of training examples was built, retaining only the semantic classes and the target value. At this point, the generalized training corpus contains three types of examples:

1. Positive examples

<X_hierarchy#sense; Y_hierarchy#sense; Yes>

2. Negative examples

<X_hierarchy#sense; Y_hierarchy#sense; No>

3. Ambiguous examples

<X_hierarchy#sense; Y_hierarchy#sense; Yes/No>

The third situation occurs when the training corpus contains both positive and negative examples for the same hierarchy types. For example, both relations <apartment#1; woman#1; No> and <hand#1; woman#1; Yes> are mapped into the more general type <entity#1; entity#1; Yes/No>. However, the first example is negative (a POSSESSION relation), while the second one is a positive example.

Step 2. Learning constraints for unambiguous examples

For the unambiguous examples in the generalized training corpus (those that are either positive or negative), constraints are determined using C4.5. In this context, the features are the components of the relation (the *part* and, respectively the *whole*) and the values of the features are their corresponding WordNet semantic classes (the furthest ancestor in WordNet of the corresponding concept).

With the first two types of examples, the unambiguous ones, a new training corpus was created on which we applied C4.5 using a 10-fold cross validation. The output is represented by 10 sets of rules generated from these unambiguous examples.

The rules in each set were ranked according to their frequency of occurrence and average accuracy obtained for that particular set. In order to use the best rules, we decided to keep only the ones that had a frequency above a threshold (occur in at least 7 of the 10 sets of rules) and an average accuracy greater than 50%.

Step 3. Specialize the ambiguous examples

A part of the generalized training corpus contains ambiguous examples. These examples refer to the same semantic classes in WordNet, but their target value is in some cases “Yes” and in others “No”. Since C4.5 cannot be applied in this situation, we recursively specialize these examples to eliminate the ambiguity.

The specialization procedure is based on the IS-A information provided by WordNet. Initially, each semantic class represented the root of one of the noun hierarchies in WordNet. By specialization, a semantic class is replaced with its first hyponym, i.e. the concept immediately below in the hierarchy. For this task, we considered again the intermediate training corpus of examples.

For instance, the examples <apartment#1, entity#1; woman#1, entity#1; No> and <hand#1, entity#1; woman#1, entity#1; Yes> that caused the ambiguity <entity#1; entity#1; Yes/No>, were replaced with <apartment#1, whole#2; woman#1, causal_agent#1; No>, respectively <hand#1, part#7; woman#1, causal_agent#1; Yes>. These two intermediate examples are thus generalized in the less ambiguous examples <whole#2; causal_agent#1; No> and <part#7; causal_agent#1; Yes>. This way, we specialize the ambiguous examples with more specific values for the attributes. The specialization process for this particular example is shown in Figure 1.

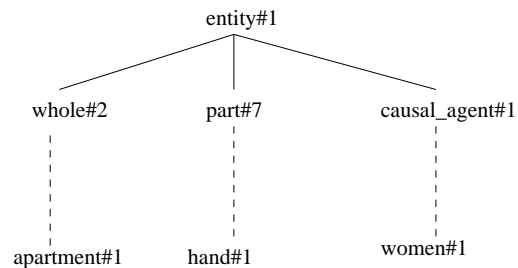


Figure 1: The specialization of the ambiguous training examples with the corresponding WordNet hyponyms as new semantic classes

Although this specialization procedure eliminates a part of the ambiguous examples, there is no guarantee it will work for all the ambiguous examples of this type. This is because the specialization splits the initial hierarchy into smaller distinct subhierarchies, and thus, the examples are distributed over this new set of subhierarchies. For the examples described above, the procedure eliminates the ambiguity through specialization of the semantic classes into two new ones: *whole* - *causal agent*, and respectively *part* - *causal agent*. However, if the training corpus contained the examples <leg#2; insect#1; Yes> and <world#7; insect#1; No>, the procedure specializes them in the ambiguous example <part#7; organism#1; Yes/No> and the ambiguity still remains.

Steps 2 and 3 are repeated until there are no more ambiguous examples. The general architecture of this procedure is shown in Figure 2. Here is an example

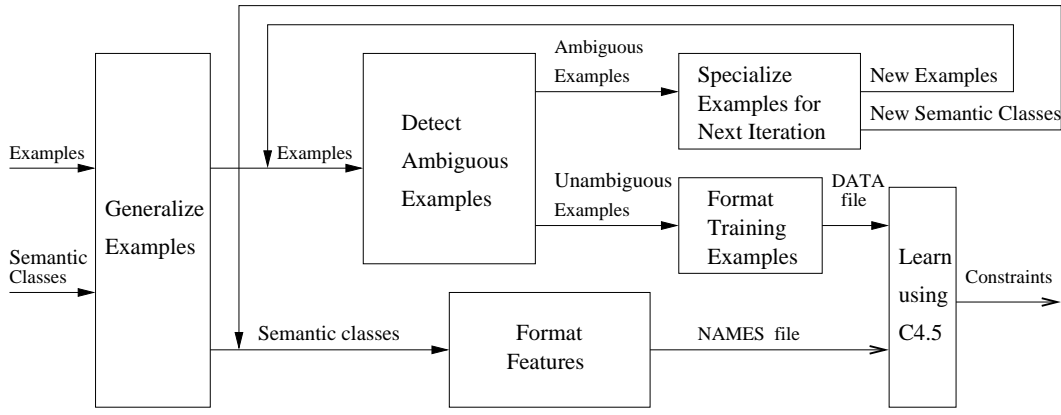


Figure 2: Architecture of the constraint learning procedure.

of iterations produced by the program to specialize ambiguous data:

```

entity#1 - entity#1
  body_of_water#1 - location#1
  location#1 - body_of_water#1
  location#1 - object#1
    region#1 - artifact#1
    region#3 - artifact#1

```

Each indentation represents a new iteration.

3.5 The Constraints

Table 1 summarizes the constraints learned by the program. The meaning of a constraint with the part *Class-X*, the whole *Class-Y* and the value 1 is “if *Part is a Class-X* and *Whole is a Class-Y* then it is a *part-whole relation*” and for the value 0 is “if *Part is a Class-X* and *Whole is a Class-Y* then it is not a *part-whole relation*”. For example, “if *Part is an entity#1* and the *Whole is a whole#2* then it is not a *part-whole relation*”. (*whole#2* is the WordNet concept meaning “an assemblage of parts that is regarded as a single entity”).

When forming larger, more complex rules, if the *part* and the *whole* contain more than one value, one of these values is negated (preceded by !). For example for the part *object#1* and the whole *organism#1* the constraint is “if the *Part is object#1* and not *substance#1* and not *natural_object#1* and the *Whole is organism#1* and not *plant#2* and not *animal#1* then *NO part-whole relation*”.

4 Results for discovering part-whole relations

To validate the constraints for extracting part-whole relations, a new test corpus “B” was created from other

10,000 sentences of TREC-9 LA Times news articles. This corpus was parsed and disambiguated using a Word Sense Disambiguation system that has an accuracy of 81% when disambiguating nouns in open-domain (Mihalcea and Moldovan, 2001). The results provided by the part-whole relation discovery procedure were validated by a human annotator.

Let us define the *precision* and *recall* performance metrics in this context:

$$precision = \frac{\text{Number of correctly retrieved relations}}{\text{Number of relations retrieved}}$$

$$recall = \frac{\text{Number of correctly retrieved relations}}{\text{Number of correct relations}}$$

On the test corpus there were 119 meronymy relations expressed by the three patterns considered. The system retrieved 140 relations, of which 117 were meronymy relations and 23 were non-meronymy relations, yielding a precision of 83% and a recall of 98%. Table 2 shows the results obtained for each of the three patterns and for all of them.

However, there were other 43 manner relations found in the corpus, expressed by other than the three lexico-syntactic patterns considered in this paper, yielding a global meronymy relation coverage (recall) of 72%. [117/119+43]

The errors are explained mostly by the fact that the genitives and the verb *have* are very ambiguous. These lexico-syntactic patterns encode numerous relations which are very difficult to disambiguate based only on the nouns they connect. The errors were also caused by the incorrect parsing of a few s-genitives, the use of the rules with smaller accuracy (e.g. 50%), the wrong word sense disambiguation of some concepts, and the lack of named entity recognition in WordNet (e.g., proper names of persons, places, etc.).

Nr	Class-X	Class-Y	Value	Accuracy	Frequency	Example
1	object#1	social_event#1	1	69.84	9	<i>scene#4 - movie#1</i>
2	whole#2	social_event#1	1	63.00	7	<i>sequence#3 - movie#1</i>
3	entity#1	group#1	1			<i>academician#1 - academy#2</i> <i>king#1 - royalty#1</i>
4	location#1	people#1	0	50.00	8	<i>section#3 - nation#1</i>
5	organism#1	system#1	0	50.00	8	<i>archbishop#1 - York#1</i>
6	group#1	group#1	1			<i>military_reserve#1 - military_unit#1</i> <i>amoebida#1 - genus_amoeba#1</i>
7	collection#1	arrangement#2	0	92.60	10	<i>data#1 - table#1</i>
8	arrangement#2	social_group#1	0	85.70	10	<i>hierarchy#1 - church#1</i>
9	system#1	collection#1	0	85.70	10	<i>economy#1 - selection#2</i>
10	entity#1	entity#1	1			<i>door#4 - car#4</i> <i>point#15 - knife#1</i>
11	point#2	object#1	0	89.55	10	<i>place#1 - wall#2</i>
12	location#1	object#1	1			<i>base#16 - box#1</i>
13	geographic_area#1	instrumentality#3	0	80.75	8	<i>graveyard#1 - ship#1</i>
14	person#1	person#1	0	89.55	10	<i>child#1 - woman#1</i>
15	object#1 !substance#1 !natural_object#1	organism#1 !plant#2 !animal#1	0			<i>desk#1 - man#1 - No</i> <i>feather#1 - bird#1 - Yes</i> <i>blade#1 - grass#1 - Yes</i> <i>body#1 - man#1 - Yes</i>
16	organism#1 !plant#2	object#1 !land#3	0			<i>author#1 - book#1 - No</i> <i>Romanian#1 - Romania#1 - Yes</i>
17	entity#1	whole#2	0	98.82	10	<i>author#1 - book#1</i> <i>attorney#1 - hotel#1</i>
18	entity#1	location#1	0	98.16	10	<i>ambassador#1 - Romania#1</i> <i>conqueror#1 - Denmark#1</i>
19	entity#1	thing#12	0	97.46	10	<i>cup#3 - something#1</i>
20	entity#1	body_of_water#1	0	91.76	10	<i>commissioner#1 - pacific#1</i>

Table 1: The list of all the constrains accompanied by examples (! means “is not”, 1 means “Is a part-whole relation,” and 0 means “Is not a part-whole relation”)

5 Application to Question Answering

The part-whole semantic relation occurs with high frequency in open text. Its discovery is paramount for many applications. In this section we mention only Question Answering. For many questions such as “*What parts does General Electric manufacture?*”, “*What are the components of X?*”, “*What is Y made of?*”, etc., the discovery of part-whole relations is necessary to assemble the right answer.

The concepts and part-whole relations acquired from a collection of documents can be useful in answering difficult questions that normally cannot be handled based solely on keywords matching and proximity. As the level of difficulty increases, Question Answering systems need richer semantic resources, including ontologies and larger knowledge bases. Consider the question:

What does the AH-64A Apache helicopter consist of?

For questions like this, the system must extract all the components the war helicopter has. Unless an ontology of such army attack helicopter parts exists in the knowl-

edge base, which in an open domain situation is highly unlikely, the system must first acquire from the document collection all the direct and indirect pieces the helicopter is made of. These parts can be scattered all over the text collection, so the Question Answering system has to gather together these partial answers into a single and concise hierarchy of parts. This technique is called Answer Fusion.

Using a state-of-the-art Question Answering system (Harabagiu et al. , 2001) adapted for Answer Fusion (Girju, 2001) and including a meronymy module, the question presented above was answered by searching the Internet at the website for Defence Industries - Army (www.army-technology.com). The system started with the question focus *AH-64A Apache helicopter* and extracted and disambiguated all the meronymy relations using the part-whole module. The following taxonomic ontology was created for this question:

AH-64A Apache Helicopter
 Helfire air-to-surface missile
 millimetre wave seeker
 70mm Folding Fin Aerial rocket
 30mm Cannon camera
 armaments

Number of Relations	Y verb X	Y's X	X of Y	All patterns
Number of patterns	280	225	962	1467
Number of correct relations	18	23	78	119
Number of relations retrieved	25	24	91	140
Number of correctly retrieved relations	18	22	77	117
Precision	72%	91.16%	84.61%	83.57%
Recall	100%	95.65%	98.71%	98.31%

Table 2: The number relations obtained and the accuracy for each pattern and for all the patterns used for this research

General Electric 1700-GE engine
4-rail launchers
four-bladed main rotor
anti-tank laser guided missile
Longbow millimetre wave fire control radar
integrated radar frequency interferometer
rotating turret
tandem cockpit
Kevlar seats

For example, the relation “*AH-64A Apache helicopter* has part *Hellfire air-to-surface missile*” was determined from the sentence “*AH-64A Apache helicopter has a Longbow-millimeter wave fire control radar and a Hellfire air-to-surface missile*”. For validation only the heads of the noun phrases were considered as they occur in WordNet (i.e., *helicopter* and *air-to-surface missile*, respectively).

6 Conclusions

The method presented in this paper for the detection and validation of part-whole relation is semi-automatic and has a better accuracy than the previous attempts (Charniak, 1999). It discovers semi-automatically the part-whole lexico-syntactic patterns and learns (automatically) the semantic constraints needed for the disambiguation of these generally applicable patterns.

We combined the results of the decision tree learning with an IS-A hierarchy (the WordNet IS-A relation) specialization for a more accurate learning.

The method presented in this paper can be generalized to discover other semantic relations. The only part-whole elements used in this algorithm were the patterns and the training examples. Thus the learning procedure and the validation procedure are generally applicable and we intend to use the method for the detection of other semantic relations such as manner, influence, and others. The inconvenience of the method is that for a very precise learning the number of examples (both positive and negative) should be very large.

We also intend to automate the detection of lexico-

syntactic patterns and to discover constraints for all the part-whole patterns.

References

- Matthew Berland and Eugene Charniak. 1999. Finding Parts in Very Large Corpora. In *Proceedings of the 37th Annual Meeting of the Association for the Computational Linguistics (ACL-99)*, pages 57-64, College Park, MD.
- Eugene Charniak. 2000. A maximum-entropy-inspired parser. In *Proceedings of the North American Chapter of the Association for Computational Linguistics (NAACL 2000)*, Seattle, WA.
- M. Evens, B. Litowitz, J. Markowitz, R. Smith, and O. Werner. 1980. Lexical-semantic relations: A Comparative Survey. In *Proceedings of the Linguistic Research*, pages 187-190, Edmonton, Canada.
- Peter Gerstl and Simone Pribbenow. 1996. A conceptual theory of Part-Whole Relations and Its Applications. In *Data and Knowledge Engineering*, 20(3), pages 305-322.
- Roxana Girju. 2001. Answer Fusion with On-Line Ontology Development. In *Proceedings of the North American Chapter of the Association for Computational Linguistics (NAACL) - Student Research Workshop, (NAACL 2001)*, Pittsburgh, PA.
- Marti Hearst. 1998. Automated Discovery of WordNet Relations. In *An Electronic Lexical Database and Some of its Applications*, MIT Press, Cambridge, MA.
- Madelyn Iris, Bonnie Litowitz and Martha Evens. 1988. Problems of the Part-Whole Relation. In *Relational Models of the Lexicon: Representing Knowledge in Semantic Networks*, Cambridge University Press, Evens (ed.), pages 261-288.
- Rada Mihalcea and Dan Moldovan. 2001. A Highly Accurate Bootstrapping Algorithm for Word Sense Disambiguation. In the *International Journal on Artificial Intelligence Tools*, vol.10, no.1-2, pages 5-21.

- S. Harabagiu, D. Moldovan, M. Pasca, M. Surdeanu, R. Mihalcea, R. Girju, V. Rus, F. Lacatusu, P. Moraescu, and R. Bunescu. 2001. Answering Complex, List and Context Questions with LCC's Question-Answering Server. In Proceedings of the *Text Retrieval Conference for Question Answering (TREC 10)*.
- J. Ross Quinlan. 1993. C4.5: Programs for Machine Learning. *Morgan Kaufmann*.
- Morton Winston, Roger Chaffin, and Douglas Herrmann. 1987. A Taxonomy of Part-Whole Relations. In *Cognitive Science*, vol. 11, pages 417-444.