Exploring Noun-Modifier Semantic Relations

Vivi Nastase and Stan Szpakowicz
School of Information Technology and Engineering
University of Ottawa
Ottawa, ON, Canada
{vnastase, szpak}@site.uottawa.ca

Abstract

We explore the semantic similarity between base noun phrases in clusters determined by a comprehensive set of semantic relations. The attributes that characterize modifiers and nouns are extracted from *WordNet* and from *Roget's Thesaurus*. We use various machine learning tools to find combinations of attributes that explain the similarities in each category. The experiments gave promising results, with a good level of generalization and interesting sets of rules.

1 Introduction

We consider the nature of semantic relations in base noun phrases (base NPs) consisting of a head noun and one modifier (adverb, adjective, noun). Such relations capture the interaction between the two elements. Base NPs in the same semantic relation should, intuitively, share some characteristics. Finding features that make base NPs in the same semantic relation close to one another should also indirectly validate the list of semantic relations. The list with which we work consists of 50 semantic relations, and was developed by unifying three separate lists of relations, for the syntactic levels of multi-clause sentences, clauses and noun phrases (Nastase and Szpakowicz, 2001a). Each of the three initial lists was tested and validated (Barker, 1998). The list with examples is presented in the Appendix.

We look at descriptions of modifiers and head nouns in lexical resources to find the attributes that make them similar with respect to our semantic relations. We explore available lexical resources (*WordNet* and *Roget's Thesaurus*) to provide the attributes, and machine learning (ML) tools to find the most salient combinations of attributes. The choice of ML tools is driven by the type of output we want - symbolic rules, easy to understand - and the type of processing of the attributes imposed by our task. In principle we look for a generalization in the ontology that underlies *WordNet* or *Roget's*. We experiment with memory-based learning (MBL), decision tree induction (C5.0), rule induction (RIPPER) and relational learners (FOIL).

The most common methods of assessing word similarity compute a distance (Budanitsky and Hirst, 2001), or find the information content of the most specific subsuming concept in the IS-A hierarchy in a lexical resource (Resnik, 1999). Similarity between noun-modifier pairs is more complex. The distance between the two heads or two modifiers in a pair of base NPs can be zero, yet there may be no similarity between them, with respect to semantic relations. For example: *snow blindness* - **effect** versus *snow report* - **topic** (sense 2 of the noun *snow* in *WordNet 1.6* - {*layer*}) or : *pressure cooker*

- **instrument** versus *heavy cooker* - **property** (sense 1 of the noun *cooker* in *WordNet 1.6* - {*cooking utensil, cookware*}). There may be a way of combining the semantic distance between the heads with those between the modifiers. We are not looking for such a formula. Instead, we use the same lexical resources that are employed in finding distance metrics, and we extract features that characterize the words in base NPs. There may be several reasons why base NPs are similar. The similarity may be between the components of the base NPs, for example: **agent** – *student protest* and *animal attack* - the modifiers are sentient beings, the head nouns express actions. Otherwise, there may be a relational similarity, for example: **type** – *oak tree* and *cumulus cloud* - in both NPs the head noun is a hypernym of the modifier. Each relation may have its own signature, as far as such characteristics as described above are concerned. We will therefore let the machine learning tool find the appropriate combination of attributes for the purpose of characterization.

A review of related work presented in *Section 2* is followed by an overview of the data used in these experiments in *Section 3*. A discussion of the attributes that characterize the data, collected from *WordNet1.6* and *Roget's* is presented in *Section 5*. As an intermediate step we need a mapping between word senses in these two lexical resources. The algorithm that disambiguates senses in *Roget's* based on information extracted from *WordNet* is presented in *Section 4*. The learning experiments are the core of this work; their results are discussed in *Section 6*.

2 Related work

Several attempts have been made to learn the assignment of semantic relations to modifier-noun pairs, without necessarily seeking insight into their nature. The domains, the lists of relations and the methods all vary.

Rosario and Hearst (2001) perform ML using neural networks. They learn semantic relations between a noun and its modifier in a medical domain, to which the list of semantic relations and the lexical resource have been tailored.

Rosario et al. (2002) presents a continuation of that research. The authors look manually for rules that classify correctly noun compounds in the medical domain, based on the MeSH lexical hierarchy (Medical Subject Headings). The noun compounds are extracted automatically, and sampled for manual analysis. The hierarchy is traversed in a top to bottom manner to find a level at which the noun compounds displaying different relations are properly separated. Analysis has shown that finding the appropriate level of generalization depends on the relation involved; some are easier to capture in rules than others.

Vanderwende (1994) uses a dictionary built from texts to find clues about possible semantic relations in which the word might be involved (for example, finding *for* in some definition indicates that, in combination with another word, it could display the **purpose** relation). In this work words are taken one by one, with no interest in generalization.

For general NPs, Barker and Szpakowicz (1998) use a simplified case of memory based learning. They store noun-modifier-indicator-relation tuples (the indicator is usually a preposition), and match a new NP with previously stored patterns. No lexical resource is used.

In an experiment that does not involve modifier-noun pairs, Li and Abe (1998) generalize case frames of specific verbs to concepts using *WordNet's* ontology. The experiment aims to find generalizations for the fillers of each syntactic argument of a specific verb, by finding an appropriate cut in the tree structure (defined by the hypernym/hyponym relations in the resource) that covers the examples extracted from a corpus. The best of several possible cuts in the tree is chosen according to the MDL principle.

Clark and Weir (2001) present a similar approach in choosing the sense of a noun in *WordNet*. The choice is constrained by the predicate whose argument the noun is, and

Table 1: Distribution of Semantic Relations in the data set

Rel	Occur	Rel	Occur	Rel	Occur
cause	19	agent	73	whole	10
effect	37	beneficiary	11	product	20
purpose	44	object	45	source	21
detraction	4	object-property	15	content	17
frequency	17	instrument	44	container	3
time at	31	state	11	topic	54
time through	6	property	52	measure	31
direction	8	possessor	43	equality	17
location	7	part	15	type	16
location at	24	location from	28	material	44

by the probability of the semantic class to which the noun can belong according to its senses in *WordNet*.

Lauer (1995) maps words in noun compounds onto categories in *Roget's Thesaurus*, in order to find probabilities of occurrence of certain noun compounds and their paraphrases. There is no automatic process in finding the best level of generalization.

All these approaches consider the generalization level of *one* concept. In this process, only words are used. Our approach is different. We look at generalizations of two connected concepts. There are several features which preliminary analysis has shown to be relevant to recognizing the relation between the concepts: is any of the words the result of nominalization or adjectivalization, is it an -er nominal, is it a noun, adjective or adverb. The aim is to find rules which justify the existence of certain type of interaction between the two elements of the base NP, through the analysis of information extracted from publicly available resources for a general domain, more general semantic relations, and ML methods that present an insightful look into the nature of the data.

3 The data

For the experiments described in this paper we will use a data set consisting of 600 modifier-noun pairs. The modifiers are nouns, adjectives or adverbs. These examples were gathered manually from (Levi, 1978), automatically from (Larrick, 1961), semi-automatically from *SemCor* (the version annotated with *WordNet 1.6* senses). Some examples were constructed and added for relations for which few or no examples were found in these texts. The examples that were not extracted from *SemCor* were manually annotated with *WordNet 1.6* senses. All the pairs were manually annotated with 30 semantic relations from our set of 50.

This is a rather small data set, especially compared with the richness of noun phrases in language. Using a larger set brings about a very labour-intensive task of annotating data with semantic relations, and maybe *WordNet* senses. What we look for is a set of rules to constitute the core of a semi-automatic learning system, which will use these rules to tag other examples with semantic relations. The analysis we perform using this small set of data will reveal which relations are harder to characterize and need more data, and which of them have indicators that are easier to capture in rules.

The distribution of semantic relations in this data set is presented in Table 1.

4 Word Sense Disambiguation in *Roget's* Using *WordNet*

Including *Roget's* in our experiments introduces a subsidiary task - disambiguating word senses in *Roget's*. Doing it all manually is unrealistic, so we have to look for a method of bootstrapping the disambiguation process. We turn to a resource that contains analogous

Table 2: Distribution of Semantic Relations after filtering with *Roget's*

Rel	Occur	Rel	Occur	Rel	Occur
cause	15	agent	35	whole	7
effect	31	beneficiary	9	product	16
purpose	31	object	27	source	9
detraction	4	object-property	13	content	15
frequency	12	instrument	33	container	3
time at	26	state	7	topic	43
time through	6	property	45	measure	30
direction	8	possessor	30	equality	5
location	5	part	8	type	12
location at	22	location from	19	material	29

information. We propose an algorithm that suggests a sense using the information in *WordNet*. Suggestions are then manually corrected; the idea is for the algorithm to reduce significantly the effort of manual annotation. We also had the option of using contextual information (adjacent words, for example) from the corpora we experiment with, but we have decided against that, as an algorithm which uses only *WordNet* information would be more general. The results obtained encouraged us to keep our simple algorithm.

Yarowsky (1995) selected *Roget's* senses for words using collocation information from corpora. We wanted to use only information about the word itself and its sense in *WordNet*. Kwong (1998) has shown that it is possible to determine the sense of a word in *Roget's*. She manually applied a simple algorithm that uses synsets, hypernyms and *WordNet* glosses, to a small set of words (36, divided into 3 test groups). Her experiment was carried out for nouns only.

Word sense disambiguation that we propose is automatic. It handles nouns, adjectives and adverbs extracted from the base noun phrases in the data set with which we work.

Word senses in *Roget's* were disambiguated using information about the word in *WordNet*. Specifically, the paragraph corresponding to each possible sense in *Roget's* was intersected with the *mini-net* (the ordered set of hypernyms, hyponyms, meronyms and holonyms) of the word sense in *WordNet*; the paragraph with the best overlap was taken to provide the context for the correct sense of the word under analysis.

The results obtained show that the correct *Roget's* sense can be selected from the first two senses indicated by our simple algorithm in 86.02% of the cases. The percentages shown are computed using as a base 880 – the number of unique words/senses from our data set that appear in *Roget's Thesaurus*. The average number of senses in *Roget's* for the words in the data set is 7.4. The results and the disambiguating algorithm are presented in detail in (Nastase and Szpakowicz, 2001b).

Because of the words that do not have a corresponding sense in *Roget's*, our data set is reduced to 555 base noun phrases, with the distribution presented in Table 2.

5 The Attributes

5.1 Roget's Thesaurus

Roget's Thesaurus has a strict organization. It is grouped into 6 classes, two of which are further divided into two subclasses. Since no information is lost by disregarding the class name given the more specific subclass name, we promoted 4 subclasses to classes (Jarmasz and Szpakowicz, 2000). A class has sections, a section has subsections. Subsections consists of heads which contain paragraphs for different parts of speech. A paragraph groups words and phrases. The words and the phrase heads have the same part of speech.

Roget's hierarchy is very regular, as opposed to *WordNet*'s. All the words and phrases are located at the same level. Therefore all vectors of attributes that describe each of the words in the data set have the same length. As input data for ML, we extract paths from each *Roget's* sense to the root of the ontology. For each modifier and noun in a base NP, we extract the following attributes, illustrated with an example for the adjective *parental*:

parental – the word w;
a – part of speech of w;
denominal-adj – information about the source of the word (deverbal/ denominal/true adjective or adverb, deverbal/true noun);
parent – the word w_p to which w pertains (or is derived from), according to *WordNet*. If there is no such word, then $w_p = w$;
n – part of speech of w_p;
parentage – first word in the paragraph that best fits w_p's sense;
parentage – headword;
causation – section;
abstract relations – class.

5.2 WordNet

For each modifier and noun in a base NP we extract from *WordNet* the same information as from *Roget's Thesaurus*. The only part that is changed is the information extracted from *WordNet's* ontology.

For the same adjective *parental*, here is the information extracted from *WordNet*:

parental,	**parental**,
a,	a,
denominal-adj,	denominal-adj,
parent,	parent,
n,	n,
genitor,	genitor,
progenitor primogenitor,	progenitor primogenitor,
ancestor ascendant ascendent antecedent,	ancestor ascendant ascendent antecedent,
relative relation,	relative relation,
person individual someone somebody mortal human soul,	person individual someone somebody mortal human soul,
life-form organism being living-thing,	causal-agent cause causal-agency,
entity something	entity something

We have two vectors because the noun *parent* to which *parental* pertains is a hyponym of the noun *person*, which has two hypernym sets:

$\{$ *life form, organism, being, living thing* $\}$ and
$\{$ *causal agent, cause, causal agency* $\}$.

Both these vectors will be used in learning.

WordNet's hierarchy is not regular, and vectors as those above can have varying lengths. C5.0, one of the tools used, requires input vectors to have the same length. The formatting process is described at length in (Nastase, 2001). The input for RIPPER and FOIL is obtained by reformatting the input files for C5.0.

6 Learning Noun-Modifier Relations

The purpose of these experiments, more than trying to obtain good precision and recall, is to show potential in extracting from this data rules that give an interesting and intuitive characterization of the semantic relations.

When attempting to learn all semantic relations in the same experiment, C5.0 does not give good results. We have therefore decided to split this problem into 30 binary learning problems for all the learning tools used (C5.0, RIPPER, FOIL), in order to be able to compare them. For each relation, the data is split into positive and negative examples – positive are the base NPs in the semantic relation that we want to learn, negative are all the others.

A parameter that influences the results of ML experiments is the balance between the number of examples for the different classes in the data set. Our experiments have shown that C5.0 is quite sensitive to balance, whereas RIPPER and FOIL are not. There is no standard in the literature for balancing an imbalanced set. In a comparative study, Japkowicz (2000) observes that both down-sizing and resampling the data set may have a positive effect on the outcome of learning, but all this depends both on the problem and the tool used.

When balance is a factor, misclassification costs will be one as well. We might choose a misclassification cost to compensate for the ratio of negative/positive examples. This cost will further balance the influence that the ratio has on the outcome of the experiment, by giving more importance to the less numerous class. In experiments with C5.0 both the ratio and the misclassification costs were varied. For reasons of space we only report on the best performance observed. In the case of RIPPER and FOIL, introducing misclassification costs will generate a more detailed and precise set of rules by adding new rules to the set generated with no misclassification costs.

The two sets of input data, one corresponding to *Roget's Thesaurus*, the other to *WordNet*, are used in separate learning processes.

6.1 Decision trees - C5.0

C5.0 is an ML tool that builds decision trees or rules. Data has two parts: values and attributes, each included in a separate file. All possible values for each attribute must be specified (RuleQuest, 2000).

The data is split into a training set and a test set (if desired). C5.0 uses the training part of the data to build a decision tree or rule set model of the data, which is then applied to the test set. Cross-validation is also an option. One can set a parameter to the number N of cross-validation sessions to be performed. The data set will be split into a number N of subsets. At each turn one of the subsets will serve as a test set; the rest will be used for training. The system will preserve the ratio of positive/negative examples (or in the general case, the ratio between examples in each class) in each subset.

At every step, the system picks the attribute which best discriminates between positive and negative examples (or in a general case, that best discriminates between examples in different classes/categories). Each partition of the training set thus obtained is further split according to the same principle, until a predefined depth is reached, the data overfits the model built, or the final sets are pure enough, according to some parameter.

We perform machine learning that builds decision trees and rule sets, because of the insight that these methods give into the nature of the data. We can look at the attributes, and combinations of attributes, picked by the system as best discriminating between positive and negative examples, and understand the connection between data clumped together by decision trees or rules.

We have run the learning algorithm for several partitions of the data set, with the following ratios of negative to positive examples: 1:1, 2:1, 5:1. To obtain these ratios we balance the data set by randomly down-sizing the class of negative examples. For each such partition we perform two experiments, one with no misclassification cost, the other with a misclassification cost that compensates for the ratio. The misclassification costs will penalize false negatives.

We perform three-fold cross-validation on these data sets. Although 10 is a common value, we chose 3 because most relations have few examples. Any larger value would mean that in the partition used for testing, there might be only one or two positive examples.

We should make an observation about the results of the learning process using the information extracted from *WordNet*. *WordNet*'s hierarchy is not as uniform as *Roget's*, in that there are several paths from a certain sense of a word to the root, that is why there may be several occurrences of the same noun-modifier pair, one for each of these paths.

Figure 1 illustrates the best results obtained. The ratio is 1:1, misclassification costs have no effect. The figure represents the error in classification for 10 relations, randomly picked out of the 30 relations used in this experiment. The baseline for comparison is 50% – classifying everything either as negative or as positive (the respective semantic relation). Most of the relations are well below the baseline, with one relation actually having a 0% error.

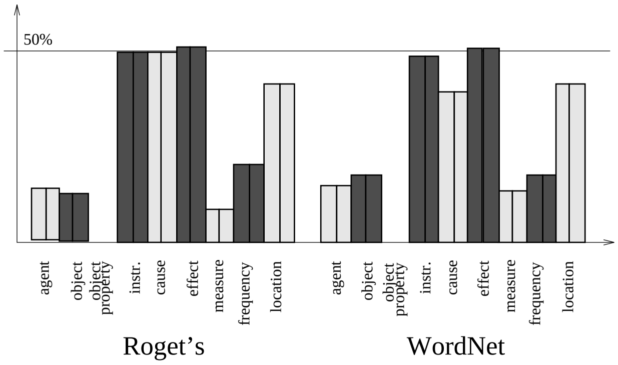

Figure 1: Errors in learning for the 1:1 ratio

The results are promising. They show that learning is possible, but the decision trees obtained are very large, and the corresponding rules hard to follow. Our main purpose is to find rules that give an insight into the nature of semantic relations. The results obtained with C5.0 were not appropriate from this point of view. Also, C5.0 treats the negative and positive classes equally with respect to learning. We are only interested in rules that characterize the positive class – the semantic relation. We have therefore moved on to RIPPER.

6.2 Rule induction - RIPPER

RIPPER is a rule induction system (Cohen, 1995). It has the option of producing rules that concentrate on the first N-1 classes in an N-classification problem. In our binary situation, the system will produce only rules that characterize the cluster defined by the given semantic relation. Experiments have shown that RIPPER is not affected by imbalance, and its input is similar to that for C5.0. The rules obtained are quite different, however, and although the data is sparse, the learner correctly characterizes aspects of the semantic relation exhibited by the available data, judged against our intuition. The relations will be tested in a large scale experiment. They will constitute the core of a classifier system. We expect them to change as more data is processed, and we will monitor these changes. This is part of future work.

RIPPER was used in experiments which had three parameters - lexical resource (possible values: *WordNet*/*Roget's*), misclassification costs (possible values: used/not used), nominalization/adjectivalization information (possible values: used/not used). All possible combinations of values for these parameters were tried. We will present a sample of the results. In all cases, misclassification costs only increased the number of rules, without modifying the ones obtained without misclassification costs. This parameter had no influence on the rules presented.

The rules generated are presented in the following format:

$$Class : -Attr_1 = Value_{Attr1}, ...Attr_N = Value_{AttrN}(NC/NM)$$

where $Class$ in our binary classification problems will be the relation that is being analysed. $Attr_X$ is an attribute that characterises the data, $Value_{AttrX}$ is one of the possible values of $Attr_X$, NC is the number of examples that the rule classifies correctly, and NM is the number of examples that the rule misclassifies. The names of the attributes indicate their source. For example: *hypernyms_depth_3_head* means the hypernym at depth three, in *WordNet*'s hypernym/hyponym hierarchy, of the head in the base NP (counting down from the most general level). The value of such an attribute is a synset.

Cause and **Effect**

Best results were obtained with *WordNet*. We present them partially here. Information about the source of the words was used.

cause – *flu virus* – H is the cause of M (H denotes the head of the noun-phrase, M the modifier).

cause :- hypernyms_depth_3_modifier = {physiological state} (9/2)

effect – *exam anxiety* – H is the effect of M.
effect :- hypernyms_depth_2_head = {condition, status},
hypernyms_depth_4_head = {ill health, unhealthiness, health problem}. (7/1)
effect :- hypernyms_depth_2_head = {happening, occurrence, natural event}
head_source = deverbal_noun. (6/1)

It might seem a mistake to have in the same rule several hypernyms of the head word or of the modifier. The structure defined by IS-A links in *WordNet* is a graph. A synset may have several hypernyms and hyponyms. Specifying two hypernyms at different levels in the hierarchy for the same word sense serves as disambiguation among the possible senses (represented as paths in this graph). The same is true of *Roget's*. A word can appear in different paragraphs, and a paragraph keyword is not unique.

Agent

Information about the source of the words improved quite dramatically the precision and quality of the rule set. Considering that syntactic indicators play a major role in the identification of this relation, it is surprising to see a big difference in the performance of the system depending on the lexical resource used – *WordNet* performed much better, and quite well even without word-source information. For comparison, we present a sample of the rules built using *WordNet*, without (1) and with (2) word-source information:

agent – *student protest* – M is the agent of H.

(1)
agent :- hypernyms_depth_3_modifier = {person, individual, ...},
hypernyms_depth_4_modifier = {leader}. (22/1)
agent :- hypernyms_depth_3_modifier = {person, individual, ...},
hypernyms_depth_1_head = {act, human action, ...}. (18/4)
agent :- hypernyms_depth_3_modifier = {person, individual, ...},
hypernyms_depth_4_head = {communication}. (8/0)
agent :- hypernyms_depth_2_modifier = {social group},
hypernyms_depth_1_head = {act, human action, ...}. (6/0)
(2)
agent :- head_source = deverbal_noun,
hypernyms_depth_3_modifier = {person, individual, ...}. (50/4)
agent :- hypernyms_depth_2_modifier = {social group},
head_source = deverbal_noun, modifier_pos = noun. (8/0)

The information that the head is a deverbal noun seems to subsume the fact that its hypernym is the synset {*act, human action, ...*} , which is the criterion used by Hull and Gomez (1996) in deciding whether a noun is a deverbal noun.

RIPPER has found rule sets that characterize well all **Temporal** relations, especially **frequency** (*daily news*). The rules are mostly based on attributes that establish the modifier as a temporal indicator.

In the case of **property** (*blue book* - H has the property M), rules in the set obtained using either lexical resource characterize different property aspects - colour, size, weight, etc. - according to the sense of the modifier.

6.3 Relational learner - FOIL

RIPPER cannot extract relations between attributes, but takes their values and uses each of them separately. Our intuition is that certain relations, for example **type** (*oak tree* - M is a type of H), **equality** (*composer arranger* - M is also H), **part** (*board member* - H is a part of M) and **whole** (*molecular chain* - M is a part of H), will be better explained by a system that can extract relations between attributes. FOIL is such a relational learner (Quinlan, 1989).

On most relations FOIL produces results quite similar to RIPPER. It did not discover the rules to characterize the relations mentioned above. We will explain why, taking **type** as an example. In this relation, the head noun is a hypernym of the modifier, but not necessarily the first hypernym, as in the following NPs:
nervous system – nervous (sense 3) → (pertains to noun) nervous system → system
oak tree – oak (sense 2) → tree
Because of the inconsistency in the position of the attributes that should be compared, FOIL could not produce the set of rules we expected.

6.4 Memory based learning (MBL)

The MBL process starts with the data described in Section 4.1. Learning consists in recording the available instances. Testing will assign relations to unseen data. This is accomplished by computing distances between the test data and the recorded instances. The relation of the example closest to our test instance will be the assigned relation (Cover and Hart, 1967).

In our first attempt at MBL, the distance between examples was computed using a very simple formula. The examples are represented as:

$$[root_{mod}, POS_{mod}, src_{mod}, WNsense_{mod}, root_{head}, POS_{head}, src_{head}, WNsense_{head}]$$

The formula to compute the distance between examples i and j is:

$$Dist(i,j) = \sum_k d(a_{ik}, a_{jk})$$

$$d(a_{ik}, a_{jk}) = \begin{cases} WNd(a_{ik}, a_{jk}) & : \; a_{ik} = root_{mod} \\ & \quad\; a_{jk} = root_{head} \\ 0 & : \; a_{ik} = a_{jk} \\ 1 & : \; a_{ik} \neq a_{jk} \end{cases}$$

where the *WordNet* distance $WNd(w_1, w_2)$ is the length of the path that connects the two synsets to which the words belong, following hypernym links.

The results obtained are quite ambiguous. This learning process will assign a list of possible relations to each test example, according to the examples in the data set that are at the same distance from the test example. The MBL process does not perform well on our examples because some attributes are more significant than others. We do not know which of them are, so we cannot adapt the distance formula to give more important attributes more weight. We prefer a learning tool such as RIPPER, which identifies the more relevant attributes.

7 Conclusions

Although the amount of data is not sufficient to find a proper set of rules that will characterize noun phrases, the results obtained by the rule induction system show that generalization is possible. Interesting sets of rules were produced for the relations which are characterized by fewer and more consistent attributes of the head and the modifier.

The purpose of this experiment was twofold. First of all, we wanted to know if it is possible to automate the assignment of semantic relations between a noun and its modifier. For this we have used different lexical resources and ML tools. The results obtained show that examples for some relations have certain attributes in common that are easily identified. The relation with nearly perfect scores, **object-property** (ex: *sunken ship* - H acquired the property M, as a result of an action described by M), is identified by just one characteristic that distinguishes it from the other relations - the modifier is the past participle of a verb. This semantic relation is situated at one end of the spectrum. Certain relations, such as **agent, object** and **instrument** are characterized by a combination of syntactic and semantic attributes. Some relations, including **cause** and **effect**, seem to rely almost exclusively on the semantics of the words in a base NP. Some of the relations that require only semantic information are nonetheless more easily identified. This is because either the modifier or the noun belongs to a certain semantic class (for example **measure** - the modifiers are mostly adjectives that denote size).

Second, we wanted to see how the two ontologies compare in the same learning task. *WordNet* performs better in almost all cases. Its more fine-grained hierarchy seems more appropriate for this generalization task. For the semantic relations that rely mostly on syntactic indicators, the lexical resource used does not influence the results much.

WordNet is one of the most commonly used lexical resources in the NLP community. *WordNet* and *Roget's Thesaurus* are lexical resources and not ontologies per se. Their IS-A hierarchy can be used as an ontology, which is what we did in this experiment. These resources were constructed starting from words, and arranging and linking them according to certain principles. However, our experiment is looking for concepts. Concepts that may be closely related semantically and that we can mentally group together for a certain reason are not found together in *WordNet* and *Roget's*. As (Barriere and Popowich, 2000) show, there are nameless concepts, which generalize concepts that would otherwise be unrelated in an IS-A hierarchy. Other resources will be tried in this task, closer to an ontology of concepts rather than words.

The ML tools used showed that a rule induction system (in our case RIPPER) performs best. Contrary to our expectations the relational learner did not bring any improvement, especially on the semantic relations on which, intuitively, it should have performed better. We have noticed that due to the granularity of the lexical resource, the distance between the two components is not constant.

A side effect of this experiment is word-sense disambiguation in *Roget's* using *WordNet*. We saw that a quite simple algorithm can significantly reduce the work load in annotating a corpus with *Roget's* senses.

We have used information about the source of the words in our experiments. Denominal adjective information is given by the *pertainym* link in *WordNet*. The rest of the information - deverbal adjective (past participle), deverbal noun - was added by the data processing scripts from a small manually built data base. We are looking at a process of automatically detecting deverbal nouns and denominal verbs using word definitions in LDOCE.

Having rules that characterize the entities involved in semantic relations is interesting for several reasons. Firstly, it gives an insight into the nature of the relation, and the categories of entities that can interact in the manner described by the relation. Secondly, it has potential use for word sense disambiguation, by choosing from possible senses for each word in the pair the combination whose interaction can be described by a semantic relation.

8 Acknowledgments

This work has been partially supported by NSERC. The 1987 edition of Penguin's *Roget's Thesaurus* has been licensed to us by Pearson Education. We thank Rada Mihalcea and Jane Morris for their comments.

References

Ken Barker and Stan Szpakowicz. 1998. Semi-automatic recognition of noun-modifier relationships. In *COLING-ACL*.

Ken Barker. 1998. *Semi-Automatic Recognition of Semantic Relationships in English Techincal Texts*. Ph.D. thesis, Department of Computer Science, University of Ottawa, Ottawa, Canada.

Caroline Barriere and Fred Popowich. 2000. Expanding the type hierarchy with nonlexical concepts. In *Advances in Artificial Intelligence, 13th Conference of the Canadian Society for Computational Studies of Intelligence*, pages 53–68, Montreal, Quebec, Canada.

Alexander Budanitsky and Graeme Hirst. 2001. Semantic distance in wordnet: An experimental, application-oriented evaluation of 5 measures. In *Workshop on WordNet and Other Lexical Resources, NAACL*.

Stephen Clark and David Weir. 2001. Class based probability estimation using a semantic hierarchy. In *Proceedings of the 2nd Meeting of the NAACL*, Pittsburg, PA.

William Cohen. 1995. Fast effective rule induction. In *12th International Conference on Machine Learning*, Lake Tahoe, California.

T. Cover and P. Hart. 1967. Nearest neighbour pattern classification. *IEEE Transactions on Information Theory*, 13:21–27.

Richard Hull and Fernando Gomez. 1996. Semantic interpretation of nominalizations. In *The 13th National Conference on Artificial Intelligence*, pages 1062–1068, Portland, Oregon, USA.

Nathalie Japkowicz. 2000. The class imbalance problem: Significance and strategies. In *2000 International Conference on Artificial Intelligence*, pages 111–117.

Mario Jarmasz and Stan Szpakowicz. 2000. The design and implementation of an electronic lexical knowledge base. In *Advances in Artificial Intelligence, 14th Conference of the Canadian Society for Computational Studies of Intelligence*, pages 325–334, Ottawa, Ontario, Canada.

Oi Yee Kwong. 1998. Aligning wordnet with additional lexical resources. In *COLING-ACL Workshop on Usage of WordNet in NLP Systems*, pages 73–79.

N. Larrick. 1961. *Junior Science Book of Rain, Hail, Sleet and Snow*. Garrard Publishing Company, Champain,IL.

Mark Lauer. 1995. *Designing Statistical Language Learners: Experiments on Noun Compounds*. Ph.D. thesis, Department of Computing, Macquarie University, Australia, December.

Judith Levi. 1978. *The Syntax and Semantics of Complex Nominals*. Academic Press, New York.

Hang Li and Naoki Abe. 1998. Generalizing case frames using a thesaurus and the mdl principle. *Computational Linguistics*, 24(2):217–244.

Vivi Nastase and Stan Szpakowicz. 2001a. Unifying semantic relations across syntactic levels. In *2nd RANLP Conference*, pages 194–198.

Vivi Nastase and Stan Szpakowicz. 2001b. Word sense disambiguation in Roget's thesaurus using WordNet. In *Workshop on WordNet and Other Lexical Resources, NAACL*, pages 17–22.

Vivi Nastase. 2001. Preparing data for learning noun-modifier semantic relations in base noun phrases, tr-2001-05. Technical report, SITE, University of Ottawa.

Ross Quinlan. 1989. Learning relations: A comparison of a symbolic and connectionst approach. Technical report, University of Sydney, Sydney, Australia.

Philip Resnik. 1999. Semantic similarity in a taxonomy: An information-based measure and its application to problems in natural language. *Journal of Artificial Intelligence*, 11:95–130.

Barbara Rosario and Marti Hearst. 2001. Classifying the semantic relations in noun-compounds via a domain specific hierarchy. In *2001 Conference on EMNLP*, pages 82–90.

Barbara Rosario, Marti Hearst, and Charles Fillmore. 2002. The descent of hierarchy and selection in relational semantics. In *ACL 2002*.

Research RuleQuest. 2000. Data mining tools : C5.0 (tutorial). http://www.rulequest.com.

Lucy Vanderwende. 1994. Algorithm for automatic interpretation of noun sequences. In *15th ACL*, pages 782–788.

David Yarowsky. 1995. Unsupervised word sense disambiguation rivaling supervised methods. In *33rd Annual Meeting of the Association for Computational Linguistics*, pages 189–196.

Appendix Noun-Modifier Relations (NMRs)
The relations that were not represented in our data sets are marked with a star(⋆).

Relation	Abbr.	Example	Paraphrase
CAUSALITY			
cause	cs	flu virus	H makes M occur or exist, H - necess. and suff.
effect	eff	exam anxiety	M makes H occur or exist, M - necess. and suff.
purpose	prp	concert hall	H is for V-ing M, M - not necess. occurs or exists
⋆entailment	ent		H makes M occur or exist, H - not known to exist
detraction	detr	headache pill	H opposes M, H - not suff. to prevent M
⋆prevention	prev		H opposes M, H - suff. to prevent M
TEMPORALITY			
⋆co-occurrence	cooc		H and M occur or exist at the same time
frequency	freq	daily exercise	H occurs every time M occurs
⋆precedence	prec		H (begins to) occurs or exists before M
time at	tat	morning exercise	H occurs when M occurs
⋆time from	tfr		H began to occur when M became true
time through	tthr	six-hour meeting	H existed while M existed, M - interval of time
⋆time to	tto		H existed until M started to exist
SPATIAL			
direction	dir	outgoing mail	H is directed towards M, M is not the final point
location	loc	home town	H is the location of M
location at	lat	desert storm	H is located at M
location from	lfr	foreign capital	H originates at M
⋆location to	lto		the destination of H is M
⋆location through	lthr		H occurred through M (M is a space)
CONJUNCTIVE			
⋆conjunction	conj		both H and M exist
⋆disjunction	disj		either one or both H and M exist
PARTICIPANT			
⋆accompaniment	acc		H is accompanied by M (co-agent)
agent	ag	student protest	M perfomrs H, M - animate or natural phen.
beneficiary	ben	student discount	M benefits from H
⋆exclusion	excl		M is excluded from H, or H replaces M
instrument	inst	laser printer	H uses M
object	obj	metal separator	M is acted upon by H
object property	obj prop	sunken ship	H underwent M
part	part	printer tray	H is part of M
posessor	posr	national debt	M has H
property	prop	blue book	H is M
product	prod	plum tree	H produces M
source	src	olive oil	M is the source of H
stative	st	sleeping dog	H is in a state of M
whole	whl	daisy chain	M is part of H
QUALITY			
container	cntr	film music	M contains H
content	cont	apple cake	M is contained in H
equative	eq	player coach	H is also M
manner	man	stylish writing	H occurs in the way indicated by M
material	mat	brick house	H is made of M
measure	meas	expensive book	M is a measure of H
⋆order	ord		H is before M in physical space
topic	top	weather report	H is concerned with M
type	type	oak tree	M is a type of H