

Using WordNet to Automatically Deduce Relations between Words in Noun-Noun Compounds

Fintan J. Costello,

School of Computer Science,
University College Dublin,
Dublin 6, Ireland.

fintan.costello@ucd.ie

Tony Veale,

Department of Computer Science,
University College Dublin,
Dublin 6, Ireland.

tony.veale@ucd.ie

Simon Dunne,

Department of Computer Science,
University College Dublin, Dublin 6, Ireland.

sdunne@inismor.ucd.ie

Abstract

We present an algorithm for automatically disambiguating noun-noun compounds by deducing the correct semantic relation between their constituent words. This algorithm uses a corpus of 2,500 compounds annotated with WordNet senses and covering 139 different semantic relations (we make this corpus available online for researchers interested in the semantics of noun-noun compounds). The algorithm takes as input the WordNet senses for the nouns in a compound, finds all parent senses (hypernyms) of those senses, and searches the corpus for other compounds containing any pair of those senses. The relation with the highest proportional co-occurrence with any sense pair is returned as the correct relation for the compound. This algorithm was tested using a 'leave-one-out' procedure on the corpus of compounds. The algorithm identified the correct relations for compounds with high precision: in 92% of cases where a relation was found with a proportional co-occurrence of 1.0, it was the correct relation for the compound being disambiguated.

Keywords: Noun-Noun Compounds, Conceptual Combination, Word Relations, WordNet

1 Introduction

Noun-noun compounds are short phrases made up of two or more nouns. These compounds are common in everyday language and are especially frequent, and important, in technical documents

(Justeson & Katz, 1995, report that such phrases form the majority of technical content of scientific and technical documents surveyed). Understanding these compounds requires the listener or reader to infer the correct semantic relationship between the words making up the compound, inferring, for example, that the phrase 'flu virus' refers to a virus that *causes* flu, while 'skin virus' describes a virus that *affects* the skin, and *marsh virus* a virus *contracted in* marshes. In this paper we describe a novel algorithm for disambiguating noun-noun compounds by automatically deducing the correct semantic relationship between their constituent words.

Our approach to compound disambiguation combines statistical and ontological information about words and relations in compounds. Ontological information is derived from WordNet (Miller, 1995), a hierarchical machine readable dictionary, which is introduced in Section 1. Section 2 describes the construction of an annotated corpus of 2,500 noun-noun compounds covering 139 different semantic relations, with each noun and each relation annotated with its correct WordNet sense.¹

Section 3 describes our algorithm for finding the correct relation between nouns in a compound, which makes use of this annotated corpus. Our general approach is that the correct relation between two words in a compound can be deduced by finding other compounds containing words from the same semantic categories as the words in the compound to be disambiguated: if a particular relation occurs frequently in those other compounds, that relation is probably also the correct relation for the compound in question. Our al-

¹A file containing this corpus is available for download from <http://inismor.ucd.ie/~fintanc/wordnet.compounds>

Table 1: Thematic relations proposed by Gagné.

relation	example
head <i>causes</i> modifier	flu virus
modifier <i>causes</i> head	college headache
head <i>has</i> modifier	picture book
modifier <i>has</i> head	lemon peel
head <i>makes</i> modifier	milk cow
head <i>made of</i> modifier	chocolate bird
head <i>for</i> modifier	cooking toy
modifier <i>is</i> head	dessert food
head <i>uses</i> modifier	gas antiques
head <i>about</i> modifier	travel magazine
head <i>located</i> modifier	mountain cabin
head <i>used by</i> modifier	servant language
modifier <i>located</i> head	murder town
head <i>derived from</i> modifier	oil money

gorithm implements this approach by taking as input the correct WordNet senses for the constituent words in a compound (both base senses and parent or hypernyms of those senses), and searching the corpus for other compounds containing any pair of those base or hypernym senses. Relations are given a score equal to their proportional occurrence with those sense pairs, and the relation with the highest proportional occurrence score across all sense-pairs is returned as the correct relation for the compound. Section 4 describes two different leave-one-out tests of this ‘Proportional Relation Occurrence’ (PRO) algorithm, in which each compound is consecutively removed from the corpus and the algorithm is used to deduce the correct sense for that compound using the set of compounds left behind. These tests show that the PRO algorithm can identify the correct relations for compounds, and the correct senses of those relations, with high precision. Section 6 compares our algorithm for compound disambiguation with one recently presented alternative, Rosario et al.’s (2002) rule-based system for the disambiguation of noun-noun compounds. The paper concludes with a discussion of future developments of the PRO algorithm.

2 Introduction to WordNet

In both our annotated corpus of 2,500 noun-noun compounds and our proportional relation selection algorithm we use WordNet (Miller, 1995). The basic unit of WordNet is the sense. Each word in WordNet is linked to a set of senses, with each sense identifying one particular meaning of that word. For example, the noun ‘skin’ has senses representing (i) the cutis or skin of human beings, (ii)

the rind or peel of vegetables or fruit, (iii) the hide or pelt of an animal, (iv) a skin or bag used as a container for liquids, and so on. Each sense contains an identifying number and a ‘gloss’ (explaining what that sense means). Each sense is linked to its parent sense, which subsumes that sense as part of its meaning. For example, sense (i) of the word ‘skin’ (the cutis or skin of human beings) has a parent sense ‘connective tissue’ which contains that sense of skin and also contains the relevant sense of ‘bone’, ‘muscle’, and so on. Each parent sense has its own parents, which in turn have their own parent senses, and so on up to the (notional) root node of the WordNet hierarchy. This hierarchical structure allows computer programs to analyse the semantics of natural language expressions, by finding the senses of the words in a given expression and traversing the WordNet graph to make generalisations about the meanings of those words.

3 Corpus of Annotated Compounds

In this section we describe the construction of a corpus of noun-noun compounds annotated with the correct WordNet noun senses for constituent words, the correct semantic relation between those words, and the correct WordNet verb sense for that relation. In addition to providing a set of compounds to use as input for our compound disambiguation algorithm, one aim in constructing this corpus was to examine the relations that exist in naturally occurring noun-noun compounds. This follows from existing research on the relations that occur between noun-noun compounds (e.g. Gagné & Shoben, 1997). Gagné and her colleagues provide a set of ‘thematic relations’ (derived from relations proposed by, for example, Levi, 1978) which, they argue, cover the majority of semantic relations between modifier (first word) and head (second word) in noun-noun compounds. Table 1 shows the set of thematic relations proposed in Gagné & Shoben (1997). A side-effect of the construction of our corpus of noun-noun compounds was an assessment of the coverage and usefulness of this set of relations.

3.1 Procedure

The first step in constructing a corpus of annotated noun-noun compounds involved selection of a set of noun-noun compounds to classify. The source used was the set of noun-noun compounds

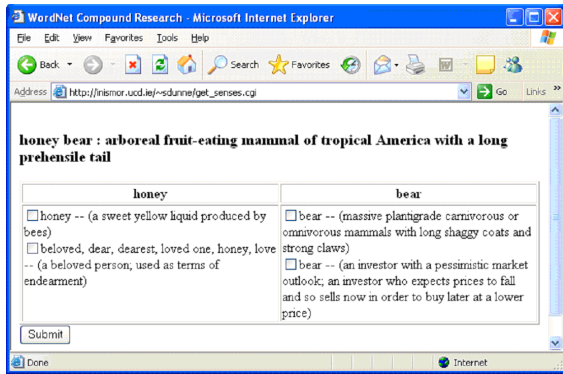


Figure 1: Selecting WordNet senses for nouns.

defined in WordNet. Compounds from WordNet were used for two reasons. First, each compound had an associated gloss or definition written by the lexicographer who entered that compound into the corpus: this explains the relation between the two words in that compound. Sets of compounds from other sources would not have such associated definitions. Second, by using compounds from WordNet, we could guarantee that all constituent words of those compounds would also have entries in WordNet, ensuring their acceptability to our compound disambiguation algorithm. An initial list of over 40,000 two-word noun-noun compounds were extracted from WordNet version 2.0. From this list we selected a random subset of compounds and went through that set excluding all compounds using scientific latin (e.g. *ocimum basilicum*), idiomatic compounds (e.g. *zero hour*, *ugli fruit*), compounds containing proper nouns (e.g. *Yangtze river*), non-english compounds (e.g. *faux pas*), and chemical terminology (e.g. *carbon dioxide*).

The remaining compounds were placed in random order, and the third author annotated each compound with the WordNet noun senses of the constituent words, the semantic relation between those words, and the WordNet verb sense of that relation (again, with senses extracted from WordNet version 2.0). A web page was created for this annotation task, showing the annotator the compound to be annotated and the WordNet gloss (meaning) for that compound (see Figure 1). This page also showed the annotator the list of possible WordNet senses for the modifier noun and head noun in the compound, allowing the annotator to select the correct WordNet sense for each word. After selecting correct senses for the words in the compound, another page was presented (Figure 2)

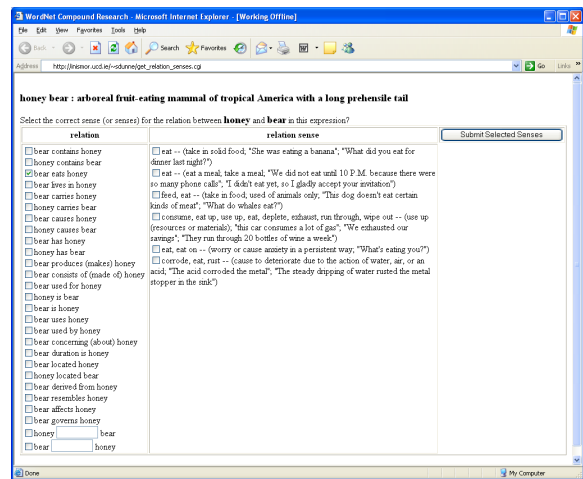


Figure 2: Selecting relation and relation senses.

allowing the annotator to identify the correct semantic relation for that compound, and then to select the correct WordNet sense for the verb in that relation.

We began by assuming that Gagné & Shoben's (1997) set of 14 relations was complete and could account for all compounds being annotated. However, a preliminary test revealed some common relations (e.g., *eats*, *lives in*, *contains*, and *resembles*) that were not in Gagné & Shoben's set. These relations were therefore added to the list of relations we used. Various other less commonly-occurring relations were also observed. To allow for these other relations, a function was added to the web page allowing the annotator to enter the appropriate relation appearing in the form "noun (insert relation) modifier" and "modifier (insert relation) noun". They would then be shown the set of verb senses for that relation and asked to select the correct sense.

3.2 Results

Word sense, relation, and relation sense information was gathered for 2,500 compounds. Relation occurrence was well distributed across these compounds: there were 139 different relations used in the corpus. Frequency of these relations ranged widely: there were 86 relations that occurred for just one compound in the corpus, and 53 relations that occurred more than once. For the relations that occurred more than once in the corpus, the average number of occurrences was 46. Table 2 shows the 5 most frequent relations in the corpus: these 5 relations account for 54% of compounds. Note that 2 of the 5 relations in Table 2 (*head con-*

Table 2: 5 most frequent relations in the corpus.

relation	frequency	number of relation senses
head <i>used for</i> modifier	382	3
head <i>about</i> modifier	360	1
head <i>located</i> modifier	226	3
head <i>contains</i> modifier	217	3
head <i>resembles</i> modifier	169	1

tains modifier and *head resembles modifier*) are not listed in Gagné’s set of taxonomic relations. This suggests that the taxonomy needs to be extended by the addition of further relations.

In addition to identifying the relations used in compounds in our corpus, we also identified the WordNet verb sense of each relation. In total 146 different relation senses occurred in the corpus. Most relations in the corpus were associated with just 1 relation sense. However, a significant minority of relations (29 relations, or 21% of all relations) had more than one relation sense; on average, these relations had three different senses each. Relations with more than one sense in the corpus tended to be the more frequently occurring relations: as Table 2 shows, of the 5 most frequent relations in the corpus, 3 were identified as having more than one relation sense. The two relations with the largest number of different relation senses occurring were *carry* (9 senses) and *makes* (8 senses). Table 3 shows the 3 most frequent senses for both relations. This diversity of relation senses suggests that Gagné’s set of thematic relations may be too coarse-grained to capture distinctions between relations.

4 Compound Disambiguation Algorithm

The previous section described the development of a corpus of associations between word-sense and relation data for a large set of noun-noun compounds. This section presents the ‘Proportional Relation Occurrence’ (PRO) algorithm which makes use of this information to deduce the correct relation for a given compound.

Our approach to compound disambiguation works by finding other compounds containing words from the same semantic categories as the words in the compound to be disambiguated: if a particular relation occurs frequently in those other compounds, that relation is probably also the correct relation for the compound in question. We take WordNet senses to represent semantic cate-

Table 3: Senses for relations *makes* and *carries*.

relation	relation sense gloss	example
Makes	bring forth or yield;	spice tree
Makes	cause to occur or exist;	smoke bomb
Makes	create or manufacture a man-made product;	cider mill
Carries	contain or hold, have within;	pocket watch
Carries	move while supporting, in a vehicle or one’s hands;	passenger van
Carries	transmit or serve as the medium for transmission;	radio wave

gories. Once the correct WordNet sense for a word has been identified, that word can placed a set of nested semantic categories: the category represented by that WordNet sense, by the parent sense (or hypernym) of that sense, the parent of that parent, and so on up to the (notional) root sense of WordNet (the semantic category which subsumes every other category in WordNet). Our algorithm uses the set of semantic categories for the words in a compound, and searches for other compounds containing words from any pair of those categories.

Figure 3 shows the algorithm in pseudocode. The algorithm uses a corpus of annotated noun-noun compounds and, to disambiguate a given compound, takes as input the correct WordNet sense for the modifier and head words of that compound, plus all hypernyms of those senses. The algorithm pairs each modifier sense with each head sense (lines 1 & 2 in Figure 3). For each sense-pair, the algorithm goes through the corpus of noun-noun compounds and extracts every compound whose modifier sense (or a hypernym of that sense) is equal to the modifier sense in the current sense-pair, and whose head sense (or a hypernym of that sense) is equal to the head sense in that pair (lines 5 to 8). The algorithm counts the number of times each relation occurs in that set of compounds, and assigns each relation a Proportional Relation Occurrence (PRO) score for that sense-pair (lines 10 to 12). The PRO score for a given relation R in a sense-pair S is a tuple with two components, as in Equation 1:

$$PRO(R, S) = \langle \frac{|R \cap S|}{|S|}, \frac{|R \cap S|}{|D|} \rangle. \quad (1)$$

The first term of this tuple is the proportion of times relation R occurs with sense-pair S (in other words, the conditional probability of relation R

Preconditions:

The entry for each compound C in corpus D contains:
 $C_{modList}$ = sense + hypernym senses for modifier of C ;
 $C_{headList}$ = sense + hypernym senses for head of C ;
 C_{rel} = semantic relation of C ;
 $C_{relSense}$ = verb sense for semantic relation for C ;

Input:

X = compound for which a relation is required;
 $modList$ = sense + hypernym senses for modifier of X ;
 $headList$ = sense + hypernym senses for head of X ;
 $finalResultList$ = ();

Begin:

```
1 for each modifier sense  $M \in modList$ 
2   for each head sense  $H \in headList$ 
3      $relCount = ()$ ;
4      $matchCount = 0$ ;
5     for each compound  $C \in corpus D$ 
6       if ( $(M \in C_{modList})$  and ( $H \in C_{headList}$ ))
7          $relCount[C_{rel}] = relCount[C_{rel}] + 1$ ;
8          $matchCount = matchCount + 1$ ;
9     for each relation  $R \in relCount$ 
10       $condProb = relCount[R] / matchCount$ ;
11       $jointProb = relCount([R] / D)$ ;
12       $scoreTuple = (relProb, jointProb)$ ;
13       $prevScoreTuple = finalResultList[R]$ ;
14      if ( $scoreTuple[1] > prevScoreTuple[1]$ )
15         $finalResultList[R] = relScoreTuple$ ;
16      if ( $scoreTuple[1] = prevScoreTuple[1]$ )
17        if ( $scoreTuple[2] > prevScoreTuple[2]$ )
18           $finalResultList[R] = scoreTuple$ ;
19 sort  $finalResultList$  by relation score tuples;
20 return  $finalResultList$ ;
```

End.

Figure 3: Compound disambiguation algorithm.

given sense-pair S); the second term is simply the proportion of times the relation co-occurs with the sense pair in the database of compounds D (in other words, the joint probability of relation R and sense-pair S). The algorithm compares the PRO score obtained for each relation R from the current sense-pair with the score obtained for that relation from any other sense-pair, using the first term of the score tuple as the main key for comparison (lines 14 and 15), and using the second term as a tie-breaker (lines 16 to 18). If the PRO score for relation R in the current sense-pair is greater than the PRO score obtained for that relation with some other sense pair (or if no previous score for the relation has been entered), the current PRO tuple is recorded for relation R . In this way the algorithm finds the maximum PRO score for each relation R across all possible sense-pairs for the compound in question. The algorithm returns a list of candidate relations for the compound, sorted by PRO score (lines 19 and 20). The relations at the front of that list (those with the highest PRO scores) are those most likely to be the correct relation for that

compound.

Tests of this algorithm suggest that, in many cases, candidate relations for a given compound will be tied on the first term of their PRO score tuple. The use of the second score-tuple term is therefore an important part of the algorithm. For example, suppose that two competing relations for some compound have a proportional occurrence of 1.0 (both relations occur in every occurrence of some sense-pair in the compound corpus). If the first relation occurs 20 times with its selected sense pair (i.e. there are 20 occurrences of the sense-pair in the corpus, and the relation occurs in each of those 20 occurrences), but the second relation only occurs 2 times with its selected sense pair (i.e. there are 2 occurrences of that sense-pair in the corpus, and the relation occurs in each of those 2 occurrences), the first relation will be preferred over the second relation, because there is more evidence for that relation being the correct relation for the compound in question.

The algorithm in Figure 3 returns a list of candidate semantic relations for a given compound (returning relations such as ‘head carries modifier’ for the compound *vegetable truck* or ‘modifier causes head’ for the compound *storm damage*, for example). This algorithm can also return a list of relation senses for a given compound (returning the WordNet verb sense ‘carries: moves while supporting, in a vehicle or one’s hands’ for the relation for the compound *vegetable truck* but the verb sense ‘carries: transmits or serves as the medium for transmission’ for the compound *radio wave*, for example). To return a list of relation senses rather than relations, we replace C_{rel} with $C_{relSense}$ throughout the algorithm in Figure 3. Section 5 describes a test of both versions of the algorithm.

5 Testing the Algorithm

To test the PRO algorithm it was implemented in a Perl program and applied to the corpus of compounds described in Section 3. We applied the program to two tasks: computing the correct relation for a given compound, and computing the correct relation sense for that compound. We used a ‘leave-one-out’ cross-validation approach, in which we consecutively removed each compound from the corpus (making it the ‘query compound’), recorded the correct relation or relation sense for that compound, then passed the correct

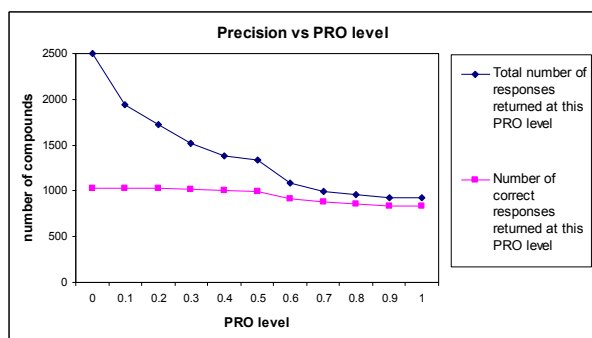


Figure 4: Graph of precision versus PRO value for returned relations

head and modifier senses of that query compound (plus their hypernyms), and the corpus of remaining compounds (excluding the query compound), to the Perl program. We carried out this process for each compound in the corpus. The result of this procedure was a list, for each compound, of candidate relations or relation senses sorted by PRO score.

We assessed the performance of the algorithm in two ways. We first considered the rank of the correct relation or relation sense for a given compound in the sorted list of candidate relations/relation senses returned by the algorithm. The algorithm always returned a large list of candidate relations or relation senses for each compound (over 100 different candidates returned for all compounds). In the relation selection task, the correct relation for a compound occurred in the first position in this list for 41% of all compounds (1,026 out of 2,500 compounds), and occurred in one of the first 5 positions (in the top 5% of returned relations or relation senses) for 72% of all compounds (1780 compounds). In the relation-sense selection task, the correct relation for a compound occurred in the first position in this list for 43% of all compounds, and occurred in one of the first 5 positions for 74% of all compounds. This performance suggests that the algorithm is doing well in both tasks, given the large number of possible relations and relation senses available.

Our second assessment considered the *precision* and the *recall* of relation/relation senses returned by the algorithm at different proportional occurrence levels (different levels for the first term in PRO score tuples as described in Equation 1). For each proportional occurrence level between 0 and 1, we assumed that the algorithm would only return a relation or relation sense when the first rela-

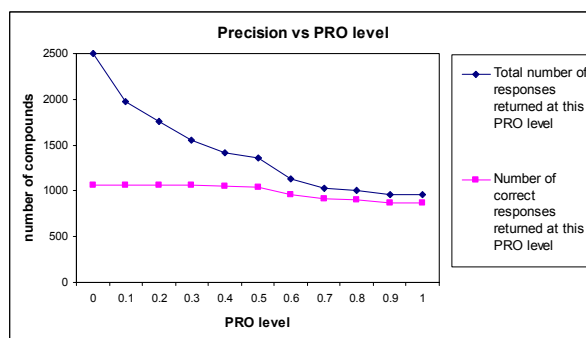


Figure 5: Graph of precision versus PRO value for returned relation senses

tion in the list of candidate relations returned had a score at or above that level. We then counted the total number of compounds for which a response was returned at that level, and the total number of compounds for which a correct response was returned. The precision of the algorithm at a given PRO level was equal to the number of correct responses returned by the algorithm at that PRO level, divided by the total number of responses returned by the algorithm at that level. The recall of the algorithm at a given PRO level was equal to the number of correct responses returned by the algorithm at that level, divided by the total number of compounds in the database (the total number of compounds for which the algorithm could have returned a correct response).

Figure 4 shows the total number of responses, and the total number of correct responses, returned at each PRO level for the relation selection task. Figure 5 shows the same data for the relation-sense selection task. As both graphs show, as PRO level increases, the total number of responses returned by the algorithm declines, but the total number of correct responses does not fall significantly. For example, in the relation selection task, at a PRO level of 0 the algorithm return a response (selects a relation) for all 2,500 compounds, and approximately 1,000 of those responses are correct (the algorithm's precision at this level is 0.41). At a PRO level of 1, the algorithm return a response (selects a relation) for just over 900 compounds, and approximately 850 of those responses are correct (the algorithm's precision at this level is 0.92). A similar pattern is seen for the relation sense responses returned by the algorithm. These graphs show that with a PRO level around 1, the algorithm makes a relatively small number of errors when selecting the correct relation or relation sense for a

given compound (an error rate of less than 10%). The PRO algorithm thus has a high degree of precision in selecting relations for compounds.

As Figures 4 and 5 show, the number of correct responses returned by the PRO algorithm did not vary greatly across PRO levels. This means that the recall of the algorithm remained relatively constant across PRO levels: in the relation selection task, for example, recall ranged from 0.41 (at a PRO level of 0) to 0.35 (at a PRO level of 1). A similar pattern occurred in the relation-sense selection task.

6 Related Work

Various approaches to noun-noun compound disambiguation in the literature have used the semantic category membership of the constituent words in a compound to determine the relation between those words. Most of these use hand-crafted lexical hierarchies designed for particular semantic domains. We compare our algorithm for compound disambiguation with one recently presented alternative, Rosario, Hearst, and Fillmore's (2002) rule-based system for the disambiguation of noun-noun compounds in the biomedical domain.

6.1 Rule-based disambiguation algorithm

Rosario et al.'s (2002) general approach to noun-noun compound disambiguation is based, as ours is, on the semantic categories of the nouns making up a compound. Rosario et al. make use of the MeSH (Medical Subject Headings) hierarchy, which provides detailed coverage of the biomedical domain they focus on. Their analysis involves automatically extracting a corpus of noun-noun compounds from a large set of titles and abstracts from the MedLine collection of biomedical journal articles, and identifying the MeSH semantic categories under which the modifier and head words of those compounds fall. This analysis generates a set of *category pairs* for each compound (similar to our sense pairs), with each pair consisting of a MeSH category for the modifier word and a MeSH category for the head.

The aim of Rosario et al.'s analysis was to produce a set of rules which would link the MeSH category pair for a given compound to the correct semantic relation for that compound. Given such a set of rules, their algorithm for disambiguating noun-noun compounds involves obtaining the MeSH category membership for the constituent

words of the compounds to be disambiguated, forming category pairs, and looking up those category pairs in the list of *category-pair*→*relation* rules. If a rule was found linking the category pair for a given compound to a particular semantic relation, that relation was returned as the correct relation for the compound in question.

To produce a list of *category-pair*→*relation* rules, Rosario et al. first selected a set of category pairs occurring in their corpus of compounds. For each category pair, they manually examined 20% of the compounds falling under that category pair, paraphrasing the relation between the nouns in that compound by hand, and seeing if that relation was the same across all compounds under that category pair. If that relation was the same across all selected compounds, that category pair was recorded as a rule linked to the relation produced. If, on the other hand, several different relations were produced for a given category pair, analysis descended one level in the MeSH hierarchy, splitting that category pair into several sub-categories. This repeated until a rule was produced assigning a relation to every compound examined. The rules produced by this process were then tested using a randomly chosen test set of 20% of compounds falling under each category pair, entirely distinct from the compound set used in rule construction, and applying the rules to those new compounds. An evaluator checked each compound to see whether the relation returned for that compound was an acceptable reflection of that compound's meaning. The results varied between 78.6% correct to 100% correct across the different category pairs.

6.2 Comparing the algorithms

In this section we first compare Rosario et al.'s algorithm for compound disambiguation with our own, and then compare the procedures used to assess those algorithms. While both algorithms are based on the association between category pairs (sense pairs) and semantic relations, they differ in that Rosario et al.'s algorithm uses a static list of manually-defined rules linking category pairs and semantic relations, while our PRO algorithm automatically and dynamically computes links between sense pairs and relations on the basis of proportional co-occurrence in a corpus of compounds. This gives our algorithm an advantage in terms of coverage: where Rosario et al.'s algorithm can

only disambiguate compounds whose constituent words match one of the *category-pair*→*relation* rules on their list, our algorithm should be able to apply to any compound whose constituent words are defined in WordNet. This also gives our algorithm an advantage in terms of extendability, in that while adding a new compound to the corpus of compounds used by Rosario et al. could potentially require the manual removal or re-definition of a number of *category-pair*→*relation* rules, adding a new compound to the annotated corpus used by our PRO algorithm requires no such intervention. Of course, the fact that Rosario et al.'s algorithm is based on a static list of rules linking categories and relations, while our algorithm dynamically computes such links, gives Rosario et al.'s algorithm a clear efficiency advantage. Improving the efficiency of the PRO algorithm, perhaps by automatically compiling a tree of associations between word senses and semantic relations and using that tree in compound disambiguation, is an important aim for future research.

Our second point of comparison concerns the procedures used to assess the two algorithms. In Rosario et al.'s assessment of their rule-based algorithm, an evaluator checked the relations returned by the algorithm for a set of compounds, and found that those relations were acceptable in a large proportion of cases (up to 100%). A problem with this procedure is that many compounds can fall equally under a number of different acceptable semantic relations. The compound *storm damage*, for example, is best defined by the relation *causes* ('damage caused by a storm'), but also falls under the relations *makes* ('damage made by a storm') and *derived from* ('damage derived from a storm'): most people would agree that these paraphrases all acceptably describe the meaning of the compound (Devereux & Costello, 2005). This means that, while the relations returned for compounds by Rosario et al.'s algorithm may have been judged acceptable for those compounds by the evaluator, they were not necessarily the most appropriate relations for those compounds: the algorithm could have returned other relations that would have been equally acceptable. In other words, Rosario et al.'s assessment procedure is somewhat weaker than the assessment procedure we used to test the PRO algorithm, in which there was one correct relation identified for each compound and the algorithm was taken to have performed correctly only if it re-

turned that relation. One aim for future work is to apply the assessment procedure used by Rosario et al. to the PRO algorithm's output, asking an evaluator to assess the acceptability of the relations returned rather than simply counting the cases where the best relation was returned. This would provide a clearer basis for comparison between the algorithms.

6.3 Conclusions

In this paper we've described an algorithm for noun-noun compound disambiguation which automatically identifies the semantic relations and relation senses used in such compounds. We've given evidence showing that, coupled with a corpus of noun-noun compounds annotated with WordNet senses and semantic relations, this algorithm can identify the correct semantic relations for compounds with high precision. Unlike other approaches to automatic compound disambiguation which typically apply to particular specific domains, our algorithm is not domain specific and can identify relations for a random sample of noun-noun compounds drawn from the WordNet dictionary. Further, our algorithm is fully automatic: unlike other approaches, our algorithm does not require the manual construction of relation rules to produce successful compound disambiguation. In future work we hope to extend this algorithm to provide a more efficient algorithmic implementation, and also to apply the algorithm in areas such as the machine translation of noun-noun compounds, where the identification of semantic relations in compounds is a crucial step in the translation process.

References

- B. Devereux & F. J. Costello. 2005. Investigating the Relations used in Conceptual Combination. *Artificial Intelligence Review*, 24(3-4): 489-515.
- C. L. Gagné, & E. J. Shoben, E. 1997. Influence of thematic relations on the comprehension of modifier-noun combinations. *Journal of Experimental Psychology: Learning, Memory and Cognition*, 23: 71-87.
- J. Justeson & S. Katz. 1995. Technical Terminology: Some linguistic properties and an algorithm for identification in text. *Natural Language Engineering*, 1-1: 9-27.
- J. Levi. 1978. *The Syntax and Semantics of Complex Nouns*. New York: Academic Press.
- G. Miller. 1995. *WordNet: A lexical database*. *Communication of the ACM*, 38(11), 39-41.
- B. Rosario, M. Hearst, & C. Fillmore. 2002. The Descent of Hierarchy, and Selection in Relational Semantics. *Proceedings of ACL-02*: 247-254.