Semantic Classification of Noun Phrases Using Web Counts and Learning Algorithms

Paul Nulty
School of Computer Science and Informatics
University College Dublin
Belfield, Dublin 4, Ireland
paul.nulty@ucd.ie

Abstract

This paper investigates the use of machine learning algorithms to label modifier-noun compounds with a semantic relation. The attributes used as input to the learning algorithms are the web frequencies for phrases containing the modifier, noun, and a prepositional joining term. We compare and evaluate different algorithms and different joining phrases on Nastase and Szpakowicz's (2003) dataset of 600 modifier-noun compounds. We find that by using a Support Vector Machine classifier we can obtain better performance on this dataset than a current state-of-the-art system; even with a relatively small set of prepositional joining terms.

1 Introduction

Noun-modifier word pairs occur frequently in many languages, and the problem of semantic disambiguation of these phrases has many potential applications in areas such as question-answering and machine translation. One very common approach to this problem is to define a set of semantic relations which capture the interaction between the modifier and the head noun, and then attempt to assign one of these semantic relations to each noun-modifier pair. For example, the phrase "*flu virus*" could be assigned the semantic relation "*causal*" (the virus causes the flu); the relation for "*desert storm*" could be "*location*" (the storm is located in the desert).

There is no consensus as to which set of semantic relations best captures the differences in meaning of various noun phrases. Work in theoretical linguistics has suggested that noun-noun compounds may be formed by the deletion of a predicate verb or preposition (Levi 1978). However, whether the set of possible predicates numbers 5 or 50, there are likely to be some examples of noun phrases that fit into none of the categories and some that fit in multiple categories.

Modifier-noun phrases are often used interchangeably with paraphrases which contain the modifier and the noun joined by a preposition or simple verb. For example, the query "*morning exercise*" returns 133,000 results from the Yahoo search engine, and a query for the phrase "*exercise in the morning*" returns 47,500 results. Sometimes people choose to use a modifier-noun compound phrase to describe a concept, and sometimes they choose to use a paraphrase which includes a preposition or simple verb joining head noun and the modifier. One method for deducing semantic relations between words in compounds involves gathering n-gram frequencies of these paraphrases, containing a noun, a modifier and a "joining term" that links them. Some algorithm can then be used to map from joining term frequencies to semantic relations and so find the correct relation for the compound in question. This is the approach we use in our experiments. We choose two sets of joining terms, based on the frequency with which they occur in between nouns in the British National Cor-

Proceedings of the ACL 2007 Student Research Workshop, pages 79–84,
Prague, June 2007.

pus (BNC). We experiment with three different learning algorithms; Nearest Neighbor, Multi-Layer Perceptron and Support Vector Machines (SVM).

2 Motivation

The motivation for this paper is to discover which joining terms are good predictors of a semantic relation, and which learning algorithms perform best at the task of mapping from joining terms to semantic relations for modifier-noun compounds.

2.1 Joining Terms

Choosing a set of joining terms in a principled manner in the hope of capturing the semantic relation between constituents in the noun phrase is difficult, but there is certainly some correlation between a prepositional term or short linking verb and a semantic relation. For example, the preposition "*during*" indicates a temporal relation, while the preposition "*in*" indicates a locative relation, either temporal or spatial.

In this paper, we are interested in whether the frequency with which a joining term occurs between two nouns is related to how it indicates a semantic interaction. This is in part motivated by Zipf's theory which states that the more frequently a word occurs in a corpus the more meanings or senses it is likely to have (Zipf 1929). If this is true, we would expect that very frequent prepositions, such as "*of*", would have many possible meanings and therefore not reliably predict a semantic relation. However, less frequent prepositions, such as *"while"* would have a more limited set of senses and therefore accurately predict a semantic relation.

2.2 Machine Learning Algorithms

We are also interested in comparing the performance of machine learning algorithms on the task of mapping from n-gram frequencies of joining terms to semantic relations. For the experiments we use Weka, (Witten and Frank, 1999) a machine learning toolkit which allows for fast experimentation with many standard learning algorithms. In Section 5 we present the results obtained using the nearest-neighbor, neural network (i.e. multi-layer perceptron) and SVM. The mechanisms of these different learning approaches will be discussed briefly in Section 4.

3 Related Work

3.1 Web Mining

Much of the recent work conducted on the problem of assigning semantic relations to noun phrases has used the web as a corpus. The use of hit counts from web search engines to obtain lexical information was introduced by Turney (2001). The idea of searching a large corpus for specific lexico-syntactic phrases to indicate a semantic relation of interest was first described by Hearst (1992).

A lexical pattern specific enough to indicate a particular semantic relation is usually not very frequent, and using the web as a corpus alleviates the data sparseness problem. However, it also introduces some problems.

- The query language permitted by the large search engines is somewhat limited.
- Two of the major search engines (Google and Yahoo) do not provide exact frequencies, but give rounded estimates instead.
- The number of results returned is unstable as new pages are created and deleted all the time.

Nakov and Hearst (2005) examined the use of web-based n-gram frequencies for an NLP task and concluded that these issues do not greatly impact the interpretation of the results. Keller and Lapata (2003) showed that web frequencies correlate reliably with standard corpus frequencies.

Lauer (1995) tackles the problem of semantically disambiguating noun phrases by trying to find the preposition which best describes the relation between the modifier and head noun. His method involves searching a corpus for occurrences paraphrases of the form "*noun preposition modifier*". Whichever preposition is most frequent in this context is chosen. Lapata and Keller (2005) improved on Lauer's results at the same task by using the web as a corpus. Nakov and Hearst (2006) use queries of the form "*noun that * modifier*" where '*' is a wildcard operator. By retrieving the words that most commonly occurred in the place of the wildcard they were able to identify very specific predicates that are likely to represent the relation between noun and modifier.

3.2 Machine Learning Approaches

There have been two main approaches used when applying machine learning algorithms to the semantic disambiguation of modifier-noun phrases.

The first approach is to use semantic properties of the noun and modifier words as attributes, using a lexical hierarchy to extract these properties. This approach was used by Rosario and Hearst (2001) within a specific domain – medical texts. Using an ontology of medical terms they train a neural network to semantically classify nominal phrases, achieving 60% accuracy over 16 classes.

Nastase and Szpakowicz (2003) use the position of the noun and modifier words within general semantic hierarchies (Roget's Thesaurus and WordNet) as attributes for their learning algorithms. They experiment with various algorithms and conclude that a rule induction system is capable of generalizing to characterize the noun phrases.

Moldovan et al (2004) also use WordNet. They experiment with a Bayesian algorithm, decision trees, and their own algorithm; semantic scattering. There are some drawbacks to the technique of using semantic properties extracted from a lexical hierarchy. Firstly, it has been noted that the distinctions between word senses in WordNet are very fine-grained, making the task of word-sense disambiguation tricky. Secondly, it is usual to use a rule-based learning algorithm when the attributes are properties of the words rather than n-gram frequency counts. As Nastase and Szpakowicz (2003) point out, a large amount of labeled data is required to allow these rule-based learners to effectively generalize, and manually labeling thousands of modifier-noun compounds would be a time-consuming task.

causal	flu virus, onion tear
temporal	summer travel, morning class
spatial	west coast, home remedy
participant	mail sorter, blood donor
quality	rice paper, picture book

Table 1**:** Examples for each of the five relations

The second approach is to use statistical information about the occurrence of the noun and modifier in a corpus to generate attributes for a machine learning algorithm. This is the method we will describe in this paper. Turney and Littman (2005) use a set of 64 short prepositional and conjunctive phrases they call "joining terms" to generate exact queries for AltaVista of the form "*noun joining term modifier*", and "*modifier joining term noun*".

These hit counts were used with a nearest neighbor algorithm to assign the noun phrases semantic relations. Over the set of 5 semantic relations defined by Nastase and Szpakowicz (2003), they achieve an accuracy of 45.7% for the task of assigning one of 5 semantic relations to each of the 600 modifier-noun phrases.

4 Method

The method described in this paper is similar to the work presented in Turney and Littman (2005). We collect web frequencies for queries of the form "*head joining term modifier*". We did not collect queries of the form "*modifier joining term head*"; in the majority of paraphrases of noun phrases the head noun occurs before the modifying word. As well as trying to achieve reasonable accuracy, we were interested in discovering what kinds of joining phrases are most useful when trying to predict the semantic relation, and which machine learning algorithms perform best at the task of using vectors of web-based n-gram frequencies to predict the semantic relation.

For our experiments we used the set of 600 labeled noun-modifier pairs of Nastase and Szpakowicz (2003). This data was also used by Turney and Littman (2005). Of the 600 modifier-noun phrases, three contained hyphenated or two-word modifier terms, for example "*test-tube baby*". We omitted these three examples from our experiments, leaving a dataset of 597 examples.

The data is labeled with two different sets of semantic relations: one set of 30 relations with fairly specific meanings, and another set of 5 relations with more abstract meanings. For our experiments we focused on the set of 5 relations. One reason for this is that dividing a set of 600 instances into 30 classes results in a fairly sparse and uneven dataset. Table 1 is a list of the relations used and examples of compounds that are labeled with each relation.

4.1 Collecting Web Frequencies

In order to collect the n-gram frequencies, we used the Yahoo Search API. Collecting frequencies for

600 noun-modifier pairs, using 28 different joining terms required 16,800 calls to the search engine. We will discuss our choice of the joining terms in the next section.

When collecting web frequencies we took advantage of the OR operator provided by the search engine. For each joining term, we wanted to sum the number of hits for the term on its own, the term followed by '*a*' and the term followed by '*the*'. Instead of conducting separate queries for each of these forms, we were able to sum the results with just one search. For example, if the noun phrase was "student invention" and the joining phrase was "by"; one of the queries would be:

"invention by student" OR "invention by a student" OR "invention by the student"

This returns the sum of the number of pages matched by each of these three exact queries. The idea is that these sensible paraphrases will return more hits than nonsense ones, such as:

"invention has student" OR "invention has a student" OR "invention has the student"

It would be possible to construct a set of hand-coded rules to map from joining terms to semantic relations; for example "during" maps to temporal, "by" maps to causal and so on. However, we hope that the classifiers will be able to identify combinations of prepositions that indicate a relation.

4.2 Choosing a Set of Joining Terms

Possibly the most difficult problem with this method is deciding on a set of joining terms which is likely to provide enough information about the noun-modifier pairs to allow a learning algorithm to predict the semantic relation. Turney and Littman (2005) use a large and varied set of joining terms. They include the most common prepositions, conjunctions and simple verbs like "*has*", "*goes*" and "*is*". Also, they include the wildcard operator '*' in many of their queries; for example "*not*", "* *not*" and "*but not*" are all separate queries. In addition, they include prepositions both with and without the definite article as separate queries, for example "*for*" and "*for the*".

The joining terms used for the experiments in this paper were chosen by examining which phrases most commonly occurred between two nouns in the BNC. We counted the frequencies with which phrases occurred between two nouns and chose the 28 most frequent of these phrases as our joining terms. We excluded conjunctions and determiners from the list of the most frequent joining terms. We excluded conjunctions on the basis that in most contexts a conjunction merely links the two nouns together for syntactic purposes; there is no real sense in which one of the nouns modifies another semantically in this context. We excluded determiners on the basis that the presence of a determiner does not affect the semantic properties of the interaction between the head and modifier.

4.3 Learning Algorithms

There were three conditions experimented with using three different algorithms. For the first condition, the attributes used by the learning algorithms consisted of vectors of web hits obtained using the 14 most frequent joining terms found in the BNC. The next condition used a vector of web hits obtained using the joining terms that occurred

1-14	**15-28**
of	against
in	within
to	during
for	through
on	over
with	towards
at	without
is	across
from	because
as	behind
by	after
between	before
about	while
has	under

Table 2: Joining terms ordered by the frequency with which they occurred between two nouns in the BNC.

from position 14 to 28 in the list of the most frequent terms found in the BNC. The third condition used all 28 joining terms. The joining terms are listed in Table 2. We used the log of the web counts returned, as recommended in previous work (Keller and Lapata, 2003).

The first learning algorithm we experimented with was the nearest neighbor algorithm 'IB1', as implemented in Weka. This algorithm considers the vector of n-gram frequencies as a multi-dimensional space, and chooses the label of the nearest example in this space as the label for each new example. Testing for this algorithm was done using leave-one-out cross validation.

The next learning algorithm we used was the multi-layer perceptron, or neural network. The network was trained using the backpropagation of error technique implemented in Weka. For the first two sets of data we used a network with 14 input nodes, one hidden layer with 28 nodes, and 5 output nodes. For the final condition, which uses the frequencies for all 28 joining terms, we used 28 input nodes, one hidden layer with 56 nodes, and again 5 outputs, one for each class. We used 20-fold cross validation with this algorithm.

The final algorithm we tested was an SVM trained with the Sequential Minimal Optimization method provided by Weka. A support vector machine is a method for creating a classification function which works by trying to find a hypersurface in the space of possible inputs that splits the positive examples from the negative examples for each class. For this test we again used 20-fold cross validation.

5. Results

The accuracy of the algorithms on each of the conditions is illustrated below in Table 3. Since the largest class in the dataset accounts for 43% of the examples, the baseline accuracy for the task (guessing "participant" all the time) is 43%.

The condition containing the counts for the less frequent joining terms performed slightly better than that containing the more frequent ones, but the best accuracy resulted from using all 28 frequencies. The Multi-Layer Perceptron performed better than the nearest neighbor algorithm on all three conditions. There was almost no difference in accuracy between the first two conditions, and again using all of the joining terms produced the best results.

The SVM algorithm produced the best accuracy of all, achieving 50.1% accuracy using the combined set of joining terms. The less frequent joining terms achieve slightly better accuracy using the Nearest Neighbor and SVM algorithms, and very slightly worse accuracy using the neural network. Using all of the joining terms resulted in a significant improvement in accuracy for all algorithms. The SVM consistently outperformed the baseline; neither of the other algorithms did so.

6. Discussion and Future Work

Our motivation in this paper was twofold. Firstly, we wanted to compare the performance of different machine learning algorithms on the task of mapping from a vector of web frequencies of paraphrases containing joining terms to semantic relations. Secondly, we wanted to discover whether the frequency of joining terms was related to their effectiveness at predicting a semantic relation.

6.1 Learning Algorithms

The results suggest that the nearest neighbor approach is not the most effective algorithm for the classification task. Turney and Littman (2005) achieve an accuracy of 45.7%, where we achieve a maximum accuracy of 38.1% on this dataset using a nearest neighbor algorithm. However, their technique uses the cosine of the angle between the vectors of web counts as the similarity metric, while the nearest neighbor implementation in Weka uses the Euclidean distance.

Also, they use 64 joining terms and gather counts for both the forms "noun joining term modifier" and "modifier joining term noun" (128 frequencies in total); while we use only the former construction with 28 joining terms. By using the SVM classifier, we were able to achieve a higher accuracy than Turney and Littman (50.1% versus 45.7%) with significantly fewer joining terms (28 versus 128). However, one issue with the SVM is

	Joining Terms 1-14	**Joining terms 15-28**	**All 28 Joining terms**
Nearest Neighbor	32.6	34.7	38.1
Multi Layer Perceptron	37.6	37.4	42.2
Support Vector Machine	44.2	45.9	50.1

Table 3**:** Accuracy for each algorithm using each set of joining terms on the Nastase and Szpakowicz test set of modifier-noun compounds.

that it never predicted the class "causal" for any of the examples. The largest class in our dataset is "participant", which is the label for 43% of the examples; the smallest is "temporal", which labels 9% of the examples. "Causal" labels 14% of the data. It is difficult to explain why the algorithm fails to account for the "causal" class; a useful task for future work would be to conduct a similar experiment with a more balanced dataset.

6.2 Joining Terms

The difference in accuracy achieved by the two sets of joining terms is quite small, although for two of the algorithms the less frequent terms did achieve slightly better results. The difficulty is that the task of deducing a semantic relation from a paraphrase such as "storm in the desert" requires many different types of information. It requires knowledge about the preposition "in"; i.e. that it indicates a location. It requires knowledge about the noun "desert", i.e. that it is a location in space rather than time, and it requires the knowledge that a "storm" may refer both to an event in time and an entity in space. It may be that a combination of semantic information from an ontology and statistical information about paraphrases could be used together to achieve better performance on this task.

Another interesting avenue for future work in this area is investigation into exactly how "joining terms" relate to semantic relations. Given Zipf's observation that high frequency words are more ambiguous than low frequency words, it is possible that there is a relationship between the frequency of the preposition in a paraphrase such as *"storm in the desert"* and the ease of understanding that phrase. For example, the preposition *'of'* is very frequent and could be interpreted in many ways. Therefore, the *'of'* may be used in phrases where the semantic relation can be easily deduced from the nominals in the phrase alone. Less common (and therefore more informative) prepositions such as '*after'* or '*because*' may be used more often in phrases where the nominals alone do not contain enough information to deduce the relation, or the relation intended is not the most obvious one given the two nouns.

References

Marti A. Hearst. 1992. Automatic Acquisition of Hyponyms from Large Text Corpora. *COLING 92:* (2) pp. 539-545*,* Nantes, France,

Frank Keller and Mirella Lapata. 2003. Using the Web to Obtain Frequencies for Unseen Bigrams. *Computational Linguistics,* 29: pp 459-484.

Mirella Lapata and Frank Keller. 2005. Web-Based Models for Natural Language Processing. *ACM Transactions on Speech and Language Processing* 2:1, pp 1-31.

Mark Lauer. 1995. *Designing Statistical Language Learners: Experiments on Noun Compounds.* PhD thesis, Macquarie University, NSW 2109, Australia.

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

Dan Moldovan, Adriana Badulescu, Marta Tatu, Daniel Antohe and Roxana Girju. 2004. Models for the Semantic Classification of Noun Phrases. In *Proceedings of the HLT/NAACL Workshop on Computational Lexical Semantics.* pp 60-67 Boston , MA.

Preslav Nakov and Marti Hearst. 2006. Using Verbs to Characterize Noun-Noun Relations. In *Proceedings of AIMSA 2006,* pp 233-244, Varne, Bulgaria.

Preslav Nakov and Marti Hearst. 2005. Using the Web as an Implicit Training Set: Application to Structural Ambiguity Resolution. *In Proceedings of HLT/EMNLP'05.* pp 835-842, Vancouver, Canada.

Vivi Nastase and Stan Szpakowicz. 2003. *Exploring Noun-Modifier Semantic Relations.* In *Fifth International Workshop on Computational Semantics*, pp 285-301. Tillburg, Netherlands.

Barbara Rosario and Marti A. Hearst. 2001. Classifying the semantic relations in noun compounds via a domain-specific lexical hierarchy. In *Proceedings of EMNLP 2001*, pp 82-90, Pittsburgh, PA, USA.

Peter D. Turney. 2001. Mining the web for synonyms: PM-IR vs LSA on TOEFL. *Proceedings of ECML'01.* pp 491-502. Freiburg, Germany.

Peter D. Turney and Michael L. Littman. 2005. Corpus-based learning of analogies and semantic relations. *Machine Learning,* 60(1–3):251–278.

Ian H. Witten and Eibe Frank. 1999. *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*. Morgan Kaufmann.

George K. Zipf. 1932. *Selected Studies of the Principle of Relative Frequency in Language*. Cambridge, MA.