# UCD-PN: Classification of Semantic Relations Between Nominals using WordNet and Web Counts

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

## Abstract

For our system we use the SMO implementation of a support vector machine provided with the WEKA machine learning toolkit. As with all machine learning approaches, the most important step is to choose a set of features which reliably help to predict the label of the example. We used 76 features drawn from two very different knowledge sources. The first 48 features are boolean values indicating whether or not each of the nominals in the sentence are linked to certain other words in the WordNet hypernym and meronym networks. The remaining 28 features are web frequency counts for the two nominals joined by certain common prepositions and verbs. Our system performed well on all but two of the relations; theme-tool and origin entity.

## 1 Introduction and Related Work

This paper describes a system for participating in SemEval 2007 task 4; “Classification of Semantic Relations Between Nominals”. This SemEval task required systems to establish whether or not a particular semantic relation held between two nominals in a sentence. There were 7 semantic relations, with approximately 70 positive and 70 negative example sentences for each relation. There were approximately 70 examples in the test sets for each relation.

This task is similar to the problem of determining what semantic relation holds between the constituents of a noun-noun compound. Work in this area has used both statistical information about the frequencies of lexical patterns and hand-built knowledge databases such as WordNet and thesaura. In our system we combine these two knowledge sources and build a set of features to use as input to a Support Vector Machine learning algorithm.

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 number of results returned is unstable as pages are created and deleted all the time, and the major search engines return only rounded frequency estimates and do not allow a very sophisticated query interface. 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.

Turney and Littman (2005) use web queries to the AltaVista search engine as the basis for their system to assign semantic relations to modifier-noun phrases. They use a set of 64 short prepositional and conjunctive phrases (joining terms) to generate exact queries of the form “*noun joining term modifier*”, and “*modifier joining term noun*”. Using 64 joining terms and trying the noun and modifier in either order resulted in a vector of 128

*Proceedings of the 4th International Workshop on Semantic Evaluations (SemEval-2007)*, pages 374–377,
Prague, June 2007.

hit counts for each noun-modifier pair. These hit counts were used with a supervised (nearest neighbor) algorithm to label the modifier-noun phrases.

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.

There have also been several approaches which used hand built knowledge sources. Rosario and Hearst (2001) used MeSH, a lexical hierarchy of medical terms. They use this hierarchy to assign semantic properties to head and modifier words in the medical domain. They use a neural network trained on these attributes to assign the noun phrases a semantic relation.

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 decision trees, a rule induction system, a relational learner and memory based learning. They conclude that the 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.

As far as we are aware ours is the first system to combine features derived from a hand-built lexical database with corpus frequencies of lexical patterns.

## 2 System Description

### 2.1 WordNet Features

Our system uses both features derived from WordNet and features obtained by collecting web frequencies for lexical patterns. We did not use any information from the sentence in which the two nominals appeared, nor did we use the query used to retrieve the examples. We did make use of the WordNet sense for the features we obtained from WordNet.

There are 48 features derived from WordNet. Most of these are boolean values indicating whether or not each of the nominals in the sentence appear below certain other high-level concepts in the hypernym hierarchy. We chose 22 high level concepts we believed may be good predictors of whether or not a nominal could be an argument of the semantic relations used in this task. These concepts are listed below in table 1.

| | |
|---|---|
| physical_entity | physical_object |
| grouping | substance |
| attribute | matter |
| psychological_feature | process |
| quantity | causal_agent |
| container | tool |
| act | device |
| work | content |
| being | event |
| natural_object | unit |
| instrumentation | state |

Table 1. Concepts in the WordNet hierarchy used to generate features.

For each of these WordNet entries we checked whether or not each of the nominals in the example sentence appeared below the entry in the WordNet hypernym tree. This gave us 44 features. We also checked whether the first nominal was a hypernym of the second; and vice-versa; and whether the first nominal was a meronym of the second; and vice versa. This gives us in total 48 boolean features derived from WordNet.

### 2.2 Web Frequencies

The remaining features were numerical values obtained by retrieving the frequencies of web searches for the two nominals joined by certain common prepositions and verbs. These joining terms are listed below in table 2.

| | |
|---|---|
| of | produces |
| for | used for |
| in | has |
| on | contains |
| at | from |
| with | causes |
| about | made from |

Table 2. Joining terms used to generate features.

To obtain the frequencies we used the API to the “MSN Live” search engine.

Choosing a set of joining terms in a principled manner is not an easy task, but there is certainly some correlation between a prepositional term or short linking verb and a semantic relation. For example, “*contains*” tends to indicate a spatial relation, while the preposition “*in*” indicates a locative relation, either temporal or spatial.

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 two nominals in the sentence were “*battery*” and “*phone*”; one of the queries would be:

*“battery in phone” OR “battery* in *a phone” OR “battery in the phone”*

These features were numeric values; the raw number of documents returned by the query.

### 2.3 Learning Algorithm

All of the features were used as input to our learning algorithm, which was a Support Vector Machine (SVM). An SVM 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. We did not normalize these values as normalization is handled by the WEKA implementation which we used.

WEKA is a machine learning toolkit written in Java (Witten and Frank, 1999). The algorithm we used was an SVM trained with the Sequential Minimal Optimization method provided by Weka.

## 3. Results

The average f-value obtained by our system using all of the training data was 65.4. There was a significant difference in performance across different relations. The results for each relation are below.

| **Relation** | **Pre** | **Rec** | **F** | **Acc** |
|---|---|---|---|---|
| cause-effect | 61.7 | 90.2 | 73.3 | 66.2 |
| instrument-agency | 59.3 | 84.2 | 69.6 | 64.1 |
| product-producer | 70.9 | 98.4 | 82.4 | 72.0 |
| origin-entity | 51.4 | 50.0 | 50.7 | 56.8 |
| theme-tool | 52.9 | 31.0 | 39.1 | 60.6 |
| part-whole | 66.7 | 69.2 | 67.9 | 76.4 |
| content-container | 71.4 | 78.9 | 75.0 | 73.0 |
| Average | 62.0 | 71.7 | 65.4 | 67. |

The standard deviation of the f-values is 13.9. The average of the f-values is brought down by two of the relations; origin-entity and theme-tool. The poor performance of these relations was noted during early experimentation with the training data; and the list of WordNet concepts and joining terms was amended to try to improve classification, but no improvement was achieved. If the results for these relations are omitted the average f-score rises to 73.6

### 3.1 Information Gain

In order to evaluate which features were the most useful for each relation, we used the Information Gain feature ranking tool in WEKA. This tool measures the change in entropy attributed to each feature and ranks them accordingly. In some cases we found that the high ranking features for a relation were ones which were intuitively relevant to predicting that relation; however some features still had high Information Gain despite seeming unlikely to be predictive of the relation.

The eight most informative features for the Cause-Effect and Content-Container relations are shown below. WordNet features are in normal

| **Cause-Effect** | **Content-Container** |
|---|---|
| quantity | Instrumentation2 |
| *at* | Container2 |
| *used for2* | *contains* |
| grouping | physical_object2 |
| object2 | physical_entity2 |
| substance | psychological_feature |
| substance2 | substance2 |
| instrumentation2 | device2 |

Table 3. The features with the highest information gain for cause-effect and content-container.

font; the joining terms for web searches in italics. The '2' after a feature indicates that the web search was of the form "N2 joining term N1"; or that the WordNet property holds for N2; where the relation is *relation*(N1,N2).

Most of these features make sense. For example, the search query “contains” and the Wordnet entry “Container” linked to the second noun are the second and third most informative for the content container class, and the query “N2 used for N1” ranks highly in the cause-effect relation. However, it is unclear why being a hyponym of “quantity” would provide information about the cause-effect relation.

## 4 Conclusion and Future Work

This paper describes a system for participating in SemEval 2007 task 4; “Classification of Semantic Relations Between Nominals”. Our system combines features generated by analyzing the WordNet hypernym tree with features which indicate the frequencies of certain lexical patterns involving the nominals and common prepositions, using the web as a corpus.

The performance of the system was above the average score of other systems which used the WordNet sense of the training examples but not the query used to obtain them. The system was held back particularly by two relations, theme-tool and origin-entity.

There are many potential avenues for future work in this area. We chose 48 features based on WordNet and 28 lexical patterns to search the web for. These were chosen arbitrarily on the basis that they looked like they would be informative in general, over all seven relations. A more principled approach would be to begin with a much larger number of features and use information gain to select the most informative features for *each relation individually*. This should improve performance by ensuring that only the most relevant features for a specific relation are used to train the classifier for that relation.

Also, there is room for more investigation into how short prepositional joining phrases map onto underlying semantic relations (Girjiu 2006).

## References

Roxana Girju. 2006. Out-of-context noun phrase semantic interpretation with cross-linguisitic evidence. In *Proceedings of the 15th ACM international conference on Information and knowledge management*

Marti A. Hearst: 1992. Automatic Acquisition of Hyponyms from Large Text Corpora. *COLING:539-545*

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. Boston , MA.*

Preslav Nakov and Marti Hearst. 2006. Using Verbs to Characterize Noun-Noun Relations, *in the Proceedings of AIMSA 2006,*

Preslav Nakov and Marti Hearst. 2005. Using the Web as an Implicit Training Set: Application to Structural Ambiguity Resolution, *in HLT/EMNLP'05,*

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

Barbara Rosario and Marti A. Hearst. 2001. Classifying the semantic relations in noun compounds via a domain-specific lexical hierarchy. *In Proceedings of the 2001 Conference on Empirical Methods in Natural Language Processing. ACL*

Peter D. Turney. 2001. Mining the web for synonyms: PM-IR vs LSA on TOEFL, *Proceedings of the Twelth European Conference on machine learning,*

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 (1999)