Using Lexical Patterns in the Google Web 1T Corpus to Deduce Semantic Relations Between Nouns

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

Fintan Costello
School of Computer Science and Informatics
University College Dublin, Belfield
Dublin 4, Ireland
fintan.costello@ucd.ie

Abstract

This paper investigates methods for using lexical patterns in a corpus to deduce the semantic relation that holds between two nouns in a noun-noun compound phrase such as “flu virus” or “morning exercise”. Much of the previous work in this area has used automated queries to commercial web search engines. In our experiments we use the Google Web 1T corpus. This corpus contains every 2,3, 4 and 5 gram occurring more than 40 times in Google's index of the web, but has the advantage of being available to researchers directly rather than through a web interface. This paper evaluates the performance of the Web 1T corpus on the task compared to similar systems in the literature, and also investigates what kind of lexical patterns are most informative when trying to identify a semantic relation between two nouns.

1 Introduction

Noun-noun combinations occur frequently in many languages, and the problem of semantic disambiguation of these phrases has many potential applications in natural language processing and other areas. Search engines which can identify the relations between nouns may be able to return more accurate results. Hand-built ontologies such as WordNet at present only contain a few basic semantic relations between nouns, such as hypernymy and meronymy.

If the process of discovering semantic relations from text were automated, more links could quickly be built up. Machine translation and question-answering are other potential applications. Noun compounds are very common in English, especially in technical documentation and neologisms. Latin languages tend to favour prepositional paraphrases instead of direct compound translation, and to select the correct preposition it is often necessary to know the semantic relation. 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 wind* 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.

2 Related Work

The idea of searching a large corpus for specific lexicosyntactic phrases to indicate a semantic relation of interest was first described by Hearst (1992). Lauer (1995) tackled 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 to represent the predicate

Proceedings of the NAACL HLT Workshop on Semantic Evaluations: Recent Achievements and Future Directions, pages 58–63,
Boulder, Colorado, June 2009. ©2009 Association for Computational Linguistics

of the nominal, which poses the same problem of vagueness as Levi's approach. Lapata and Keller (2005) improved on Lauer's results on the same task by using the web as a corpus.

Turney and Littman (2005) used queries to the AltaVista search engine as the basis for their learning algorithm. Using the dataset of Nastase and Szpakowicz (2003), they experimented with 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 neighbour algorithm to assign the noun phrases semantic relations.

Nakov and Hearst (2006) present a system that discovers verbs that characterize the relation between two nouns in a compound. By writing structured queries to a web search engine and syntactically parsing the returned 'snippet', they were able to identify verbs that were suitable predicates. For example, for the compound *neck vein*, they retrieved verbs and verb-preposition such as predicates *emerge from, pass through, terminate in*, and others. However, their evaluation is qualitative; they do not attempt to use the verbs directly to categorize a compound as a particular semantic relation.

Turney (2006) examines similarity measures for semantic relations. He notes that there are at least two kinds of similarity: attributional similarity, which applies between words, and relational similarity, which holds between pairs of words.

Words that have a high attributional similarity are known as synonyms; e.g. chair and stool. When the relations in each of two pairs of words are similar, it is said that there is an analogy between the two pairs of words, e.g. stone:mason, carpenter:wood.

Turney points out that word pairs with high relational similarity do not necessarily contain words with high attributional similarity. For example, although the relations are similar in traffic:street and water:riverbed, water is not similar to traffic, nor street similar to riverbed.

Therefore, a measure of similarity of semantic relations allows a more reliable judgment of analogy than the first-order similarity of the nouns

3 Motivation

When looking for lexical patterns between two nouns, as is required with vector-space approaches, data sparseness is a common problem. To overcome this, many of the best-performing systems in this area rely on automated queries to web search-engines (Lapata and Keller (2005), Turney and Littman (2005), Nakov and Hearst (2006)). The most apparent advantage of using search-engine queries is simply the greater volume of data available.

Keller and Lapata (2003) demonstrated the usefulness of this extra data on a type of word-sense disambiguation test and also found that web frequencies of bigrams correlated well with
frequencies in a standard corpus.

Kilgarriff (2007) argues against the use of commercial search engines for research, and outlines some of the major drawbacks. Search engine crawlers do not lemmatize or part-of-speech tag their text. This means that to obtain frequencies for may different inflectional forms, researchers must perform a separate query for each possible form and sum the results.

If part-of-speech tagging is required, the 'snippet' of text that is returned with each result may be tagged after the query has been executed, however the APIs for the major search engines have limitations on how many snippets may be retrieved for a given query (100 -1000).

Another problem is that search engine query syntax is limited, and sometimes mysterious. In the case of Google, only basic boolean operators are supported (AND, OR, NOT), and the function of the wildcard symbol (*) is limited, difficult to decipher and may have changed over time.

Kilgarriff also points out that the search API services to the major search engines have constraints on the number of searches that are allowed per user per day. Because of the multiple searches that are needed to cover inflectional variants and recover snippets for tagging, a limit of 1000 queries per day, as with the Google API, makes experimentation slow. This paper will describe the use of the Web 1T corpus, made available by Google in 2006 (Brants and Franz 2006). This corpus consists of n-grams collected from web data, and is available to researchers in its entirety, rather than through a web search interface. This means that there is no limit to the amount of searches that may be performed, and an arbitrarily complex query syntax is possible.

Despite being available since 2006, few researchers have made use of the Web 1T corpus. Hawker (2006) provides an example of using the corpus for word sense documentation, and describes a method for efficient searching. We will outline the performance of the corpus on the task of identifying the semantic relation between two nouns. Another motivation behind this paper is to examine the usefulness of different lexical patterns for the task of deducing semantic relations.

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 "during" would have a more limited set of senses and therefore accurately predict a semantic relation. Zipf also showed that the frequency of a term is related to its length. We will investigate whether longer lexical patterns are more useful at identifying semantic relations than shorter patterns, and whether less frequent patterns perform better than more frequent ones.

4 Web 1T Corpus

The Web1T corpus consists of n-grams taken from approximately one trillion words of English text taken from web pages in Google's index of web pages. The data includes all 2,3,4 and 5-grams that occur more than 40 times in these pages. The data comes in the form of approximately 110 compressed files for each of the window sizes. Each of these files consists of exactly 10 million n-grams, with their frequency counts. Below is an example of the 3-gram data:

```
ceramics collection and 43
ceramics collection at 52
ceramics collection is 68
ceramics collection | 59
ceramics collections , 66
ceramics collections . 60
```

The uncompressed 3-grams, 4-grams 5-grams together take up 80GB on disk. In order to make it possible to index and search this data, we excluded n-grams that contained any punctuation or non-alphanumeric characters. We also excluded n-grams that contained any uppercase letters, although we did allow for the first letter of the first word to be uppercase.

We indexed the data using Ferret, a Ruby port of the Java search engine package Lucene. We were able to index all of the data in under 48 hours, using 32GB of hard disk space. The resulting index was searchable by first word, last word, and intervening pattern. Only n-grams with a frequency of 40 or higher are included in the dataset, which obviously means that an average query returns fewer results than a web search. However, with the data available on local disk it is stable, reliable, and open to any kind of query syntax or lemmatization.

5 Lexical Patterns for Disambiguation

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 noun-phrase "morning exercise" may be paraphrased as "exercise in the morning" or "exercise during the morning". In a very large corpus, it is possible to find many reasonable paraphrases of noun phrases. These paraphrases contain information about the relationship between the modifier and the head noun that is not present in the bare modifier-noun phrase. By analyzing these paraphrases, we can deduce what semantic relation is most likely. For example, the paraphrases "exercise during the morning" and "exercise in the morning" are likely to occur more frequently than "exercise about the morning" or "exercise at the morning".

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 lexical pattern that links them. Some algorithm can then be used to map from lexical patterns to frequencies to semantic relations and so find the correct relation for the compound in question. This is the approach we use in our experiments.

In order to describe the semantic relation between two nouns in a compound *“noun1 noun2”* we search for ngrams that begin with *noun2* and end with *noun1*, since in English the head of the noun compound is the second word. For example, for the compound 'flu virus', we look at n-grams that begin with 'virus' and end with 'flu'. We extract the words that occur between the two nouns (a string of 1-3 words) and use these lexical patterns as features for the machine learning algorithm.

For each compound we also include n-grams which have the plural form of *noun1* or *noun2*. We assign a score to each of these lexical patterns, as the log of the frequency of the n-gram. We used the 400 most frequent lexical patterns extracted as the features for the model. Below are examples of some of the lexical patterns that were extracted:

and	or
of the	on the
of	from the
in the	the
for	to
and the	of a
for the	with the
to the	on
with	that the
in	from

Figure 1: The 20 most frequent patterns

The simplest way to use this vector space model to classify noun-noun combinations is to use a distance metric to compare a novel pair of nouns to ones previously annotated with semantic relations. Nulty (2007) compares these nearest neighbor models with other machine learning techniques and finds that using a support vector machine leads to improved classification.

In our experiments we used the support vector machine and k-nearest-neighbor algorithms from the WEKA machine learning toolkit. All experiments were conducted using leave-one-out cross validation: each example in the dataset is in turn tested alone, with all the other examples used for training. The first dataset used in these experiments was created by Nastase and Szpackowicz (2003) and used in experiments by Turney and Littmann (2005) and Turney (2006). The data consists of 600 noun-modifier compounds. Of the 600 examples, four contained hyphenated modifiers, for example “test-tube baby”. These were excluded from our dataset, leaving 596 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 relations. In these experiments we use only the set of 5 relations. The reason for this is that splitting a set of 600 examples into 30 classes results in few training examples per class. This problem is compounded by the fact that the dataset is uneven, with far more examples in some classes than in others. Below are the five relations and some examples.

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

Figure 2: Example phrases and their semantic relations

For our research we are particularly interested in noun-noun combinations. Of the 596 examples in the dataset, we found that 325 were clearly noun-noun combinations, e.g.“picture book”, rice paper”, while in the remainder the modifier was an adjective, for example “warm air”, “heavy storm”. We used only the noun-noun combinations in our experiments, as this is the focus of our research. We experimented with both lemmatization of the data and excluding semantically empty stop words (determiners and conjunctions) from the lexical patterns, however neither of these methods improved performance. Below are the results obtained with the k-nearest neighbor algorithm. The optimum value of k was 3.

Precision	**Recall**	**f-score**	**class**
.442	.452	.447	Quality
.75	.444	.558	Temporal
.243	.167	.198	Causal
.447	.611	.516	Participant
.571	.138	.222	Spatial

Figure 3: Results using the K-NN algorithm

The overall accuracy was 44% and the macro-averaged f-value was .39.

Below are the results obtained using the support-vector machine algorithm:

Precision	**Recall**	**f-score**	**class**
.725	.345	.468	Quality
.733	.407	.524	Temporal
.545	.111	.185	Causal
.472	.885	.615	Participant
.462	.207	.268	Spatial

Figure 4: Results using the Support Vector Machine

The overall accuracy was 51.7% and the macroaveraged f-value was .42. A majority class baseline (always predicting the largest class) would achieve an accuracy of 43.7%.

6 Which Lexical Patterns are Most Useful?

In addition to evaluating the Google Web 1T corpus, a motivation for this paper is to investigate what kind of lexical patterns are most useful for deducing semantic relations. In order to investigate this, we repeated the experiment one using the 3-grams, 4-grams and 5-grams separately, which gave lexical patterns of length 1, 2 and 3 respectively. Accuracy obtained using the support vector machine and k-nearest-neighbor algorithms are below:

	3-grams	4grams	5-grams	All
KNN	36	42.5	42.4	44
SVM	44.3	49.2	43.4	51.7

Figure 5: Results for different sizes of lexical patterns

Again, in each case the support vector machine performs better than the nearest neighbor algorithm. The 4- grams (two-word lexical patterns) give the best performance. One possible explanation for this is that the single word lexical patterns don't convey a very specific relation, while the 3 word patterns are relatively rare in the corpus, leading to many missing values in the training data.

We were also interested in how the frequency of the lexical patterns related to their ability to predict the correct semantic relation. To evaluate this, we ordered the 400 lexical patterns retrieved by frequency and then split them into three groups. We took the 64 most frequent patterns, the patterns ranked 100-164 in frequency, and those ranked 300-364. We chose to include 64 patterns in each group to allow for comparison with Turney and Littman (2001), who use 64 hand-generated patterns. Examples of the most frequent patterns are shown in Fig 1. Below are examples of patterns from the other two groups.

as well as	my
out of the	on Friday
of one	without
of fresh	which the
into	with my
for all	and their
was	around the
with your	when
related to the	whose
in the early	during

Figure 6: Frequency Ranks 100-120

to produce	one
but	provides
that cause	from your
of social	of edible
while the	levels and
or any other	comes from
such as the	chosen by the
are in the	producing
to provide	does not
if a	than the
from one	belonging to the

Figure 7: Frequency Ranks 300-320

The accuracies obtained using patterns in the different frequency groups are shown below.

	1-64	100-164	300-364
KNN	40.9	43.5	41.9
SVM	47.6	45.2	41.5

Figure 8: Results for different frequency bands of patterns

Although there is no large effect to the accuracy of the KNN algorithm, the Support Vector Machine seems to perform better with the most frequent patterns. One possible explanation for this is that although the less frequent patterns seem more informative, they more often result in zero matches in the corpus, which simply leaves a missing value in the training data.

7 Conclusion

This paper reports several experiments on the semantic disambiguation of noun-noun phrases using the Google Web 1T corpus, and shows that the results are comparable to previous work which has relied on a web interface to search engines. Having a useful corpus based on web data that can be stored and searched locally means that results will be stable across time and can be subject to complex queries. Experiments designed to evaluate the usefulness of different lexical patterns did not yield strong results and further work is required in this area.

References

Thorsten Brants and Alex Franz. 2006. Web 1T 5-gram Corpus Version 1.1. *Technical report, Google Research*

Tobias Hawker. 2006. Using Contexts of One Trillion Words for WSD. *Proceedings of the 10th Conference of the Pacific Association for Computational Linguistics, pages 85–93.*

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

Keller, Frank and Mirella Lapata. 2003. Using the Web to Obtain Frequencies for Unseen Bigrams Computational Linguistics 29:3, 459-484.

Adam Kilgarriff, 2007. Googleology is Bad Science. *Comput. Linguist.* 33, 1 147-151.

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

Mark Lauer. *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.*

Phil Maguire *(2007) A cognitive model of conceptual combination* Unpublished PhD Thesis, UCD Dublin

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'0*

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

Paul Nulty and Fintan Costello, 2007. Semantic Classification of Noun Phrases Using Web Counts and Learning Algorithms. *Proceedings of ACL 2007 Student Reseach Workshop.*

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

Peter D. Turney. 2005. Measuring semantic similarity by latent relational analysis. In *Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence (IJCAI-05)*, pages 1136-1141.

Peter D. Turney., and Michael L. Littman,. 2006. 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 Kaufman (1999)

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