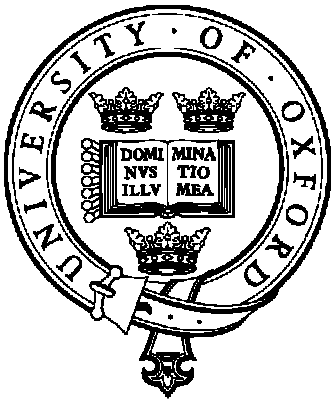
Interpreting Noun Compounds



András Dobó

Linacre College

University of Oxford

A dissertation submitted for the degree of

Master of Science in Computer Science

September 2010

Abstract

Noun compounds, which are sequences of nouns functioning as a single noun, are abundant in both written and spoken English and their interpretation is crucial for many natural language processing tasks, such as machine translation or information retrieval. Therefore there is significant ongoing interest in their interpretation. Although it is an easy task for humans, it is rather challenging for computers.

The interpretation of a noun compound can be given with a list of suitable paraphrases that are ranked according to their aptness, where the paraphrases can be verbs and prepositions. The aim of this dissertation is to develop methods that can automatically interpret two-noun noun compounds by paraphrases using large corpora. A general paraphrasing method is proposed that searches for paraphrases in static corpora, and uses Web search engine queries to validate results. Furthermore, a method for the SemEval-2 Task #9 is developed from the validation part of the general paraphrasing method.

The results of the general paraphrasing method were evaluated by human judges; based on their aptness for the noun compound, the first three paraphrases returned for each noun compound were given a score between 1 and 5 by each judge. The paraphrases ranked first, second and third by the method proposed here received average scores of 3.1842, 2.7687 and 2.5583, respectively. Further, when comparing the returned paraphrase distribution for each noun compound with the judges' distributions, it achieved an average Spearman's rank correlation coefficient of 0.3108, an average Pearson's correlation coefficient of 0.2738 and an average Kullback-Leibler divergence of 0.1589. The method for the SemEval-2 Task #9 was evaluated with the scorer provided for the task, on the test data set, by calculating the similarity of the returned paraphrase distribution for each noun compound with a gold standard. It achieved an average Spearman's rank correlation coefficient of 0.3387, an average Pearson's correlation coefficient of 0.3196 and an average Kullback-Leibler divergence of 4.1520.

Acknowledgements

First of all, I would like to thank my supervisor, Professor Stephen Pulman, for raising my interest in Natural Language Processing through his exciting Computational Linguistics lectures. Moreover, I am really thankful to him for supervising my work; his advice has always been of great value to me.

Furthermore, I would like to say thank you to my family for their constant support and encouragement, without which this wonderful year in Oxford would not have been possible.

Finally, I am deeply grateful to my girlfriend, Marianna Bicskei, for motivating me to study abroad. Without her I would not have decided to apply for Oxford, and I would have missed all the wonderful things I was able to experience during this year abroad. Furthermore, I would like to thank her for being at my side all the time, and providing me with great help in everything.

Contents

List of tables

1 Introduction

1.1 Motivation

Written and spoken English is full of noun compounds, which are, following the definition of Downing (1977), sequences of nouns functioning as a single noun. According to Baldwin and Tanaka (2004), 3.9% of the tokens in the Reuters Corpus, and 2.6% of the tokens in the British National Corpus are part of a noun compound. Their interpretation, especially given their abundance, is crucial for many natural language processing tasks, such as machine translation, question answering, information retrieval and information extraction. For example, when a machine translation system tries to translate the noun compound *bread knife* from English to Hungarian, it does not suffice for it to know the separate translation of each noun. The meaning of the noun compound is also needed, since in Hungarian it is *kenyérvágó kés*, which means *bread cutting knife*, when literally translated. Similarly, an information retrieval system, when searching for information on *plastic bottles*, needs to know whether information found on *bottles that are made of plastic* is relevant or not. Therefore there is significant interest in interpreting noun compounds in the NLP community.

Although humans can interpret noun compounds easily, it is a rather challenging task for computers, because even very similar noun compounds can have completely different meaning. Consider for example the noun compounds *plastic bottle* and *water bottle*. Although both comprise a noun denoting a material and the noun *bottle*, they are interpreted very differently ; a *plastic bottle* is a bottle made of plastic whereas a *water bottle* is a bottle used for storing water.

At first, using dictionaries for interpreting noun compounds seems to be a feasible idea. However, even for relatively frequent noun compounds, static English dictionaries give low coverage (Butnariu et al., 2009), and according to Séaghdha (2008), the frequency spectrum of noun compounds shows a Zipfian distribution, meaning that most noun compounds display a very low frequency. For example, in the British National Corpus more than half of the two-noun noun compounds occur just once (Lapata and Lascarides, 2003). Furthermore, creating a dictionary that contains all the noun compounds is impossible, since there are countless of

them and new noun compounds are constantly created. Therefore they should be interpreted automatically, at the time when their meaning is needed.

1.2 Objectives

This dissertation investigates the automatic interpretation of noun compounds using large corpora. Leaning on as Nakov and Hearst's (2006) proposal, I believe that interpreting noun compounds with paraphrases is better than using a limited number of abstract relational categories, since there exists an unlimited number of them and they can capture even subtle differences in meaning. Using paraphrases for noun compound interpretation has already proven to be useful for many applications, such as machine translation (Nakov, 2008) and solving relational similarity problems (Nakov and Hearst, 2008). I further assume that using a handful of paraphrases is more suitable than using just one paraphrase, as one is often not enough to capture the full meaning of a noun compound; therefore each noun compound is interpreted with a distribution of a small number of suitable paraphrases.

The general interpretation method described in this work aims to find those verbs and prepositions in the used corpora which are suitable for paraphrasing the noun compounds. The basic idea is to search for those sentences that paraphrase the noun compound in focus, count how many times each paraphrase is found with that noun compound, and then create a ranked list of paraphrases based on these frequencies. The search for paraphrases is intended to be done with two static corpora, namely the British National Corpus and the Web 1T 5-gram Corpus. Web search engine queries shall be used to validate the results then. This basic concept described above shall then be further extended to improve the results obtained.

To conclude, the focus of this work lies on the interpretation of two-noun noun compounds, although the interpretation of noun compounds with more than two nouns is also discussed briefly.

1.3 Outline of the dissertation

Chapter 2 describes the material applied during the work. This includes the corpora, software, dataset and Web search engines.

Next, Chapter 3 gives information on previous research conducted in the area of noun compound interpretation, discussing some proposed inventory-based and paraphrasing methods.

Thereafter, Chapter 4 proposes a general paraphrasing algorithm for two-noun noun compound interpretation that uses static corpora to find suitable paraphrases and employs Web search engine queries to validate the results. Then, it is showed how a method can be developed from its Web validation part for the SemEvel-2 Task #9 (Butnariu et al., 2009). At the end of the chapter, light is shed upon how the general paraphrasing algorithm can be employed to interpret noun compounds consisting of more than two nouns.

Chapter 5 is dedicated to explaining two methods, which automatically create categories of nouns that are semantically similar. These categories of semantically similar words are applied in the general paraphrasing method to improve its recall.

Chapter 6 discusses the results of the two paraphrasing methods proposed in this dissertation. The results of the general paraphrasing method are evaluated by human judges, while the evaluation of the program for the SemEval-2 Task #9 is done using the scorer provided for the task.

Finally, Chapter 7 gives a summary of the dissertation and considers potential future directions.

2 Materials used

This chapter gives information on the materials used for this project. Section 2.1 describes the corpora used for paraphrasing noun compounds. Section 2.2 gives information on the software and dataset applied throughout the work. Section 2.3 explains the Web search engines employed in paraphrase validation and in the method for the SemEval-2 Task #9.

2.1 Corpora

2.1.1 Static corpora

2.1.1.1 The British National Corpus (BNC, World Edition)

The British National Corpus (BNC_Consortium, 2001) is a corpus of about a100 million words collected by the BNC Consortium. It has both a written and a spoken part. The first part, constituting about 90% of it, was accumulated from a wide range of sources including newspapers, journals, academic and non-academic books among others. The second part, constituting the remaining 10%, consists of orthographic transcriptions of informal conversations, government meetings and radio shows among others. An instance of the British National Corpus, which was previously parsed with the C&C CCG parser (Clark and Curran, 2004), was used for this dissertation under the license given on the attached CD. It is available on the Curlew server[1] of the Computing Laboratory. The grammatical relations utilised by the parser are fully described by Briscoe (2006), and a comprehensive description of the parser is available in Clark and Curran (2007).

2.1.1.2 The Web 1T 5-gram Corpus

The Web 1T 5-gram Corpus (Brants and Franz, 2006), also commonly known as the Google N-grams Corpus, is generated from more than 1 trillion words from Web pages by Google Inc. It contains English n-grams (up to 5-grams), which appeared at least 40 times in the

[1] curlew.comlab.ox.ac.uk

considered Web pages, with their observed frequencies. The instance used for this project is located on the Curlew server, and it was used under the license given on the attached CD.

2.1.2 The Web

As analysed by Nakov (2007) among others, the Web can be viewed as a huge corpus with Web search engines as interfaces. As opposed to the previous two corpora, it is in constant change. In the course of this project, the Google and Yahoo! search engines were employed to access the corpus provided by the Web.

2.2 Software and dataset

2.2.1 WordNet (version 3.0)

WordNet (Fellbaum, 1998) is a huge lexical database of the English language that comprises sets of cognitive synonyms called "synset" for nouns, verbs, adjectives and adverbs. These synsets form a network of conceptual-semantic and lexical relations. It was selected to find base forms of nouns as well as verbs and to get the synonyms, hypernyms and sister words of nouns. Its latest version (3.0) contains 117798 unique nouns in 82115 synsets, and 11529 unique verbs in 13767 synsets[2]. It was used under the license given on the attached CD.

2.2.2 The Java WordNet Library (JWNL, version 1.4 rc2)

The JWNL[3] is a Java API that enables access to the WordNet relational dictionary and can be consulted under BSD License[4]. It was used to access WordNet from the Java programs written for this dissertation. To improve the performance of the programs that use the WordNet relational dictionary extensively, the dictionary is converted into a map format using the net.didion.jwnl.utilities.DictionaryToMap class of JWNL. Then it can be fully loaded into memory, which results in a significantly better performance than when drawing on the original dictionary files.

---

[2] http://wordnet.princeton.edu/wordnet/man/wnstats.7WN.html
[3] http://sourceforge.net/projects/jwordnet/
[4] http://www.opensource.org/licenses/bsd-license.php

2.2.3 Morphg

Morphg (Minnen et al., 2001) is a tool capable of generating morphological inflections of English nouns and verbs, and is available under the license given on the attached CD. It was employed to generate plural forms of nouns, third person singular, past, past participle and progressive forms of verbs in this research.

2.2.4 C&C POS tagger (version 1.0, with models version 1.02)

The C&C POS tagger, a part of the C&C tools (Clark and Curran, 2004), is an efficient Maximum Entropy part-of-speech tagger (Curran and Clark, 2003), which uses the Penn Treebank POS tagset (Santorini, 1995). It was applied for the part-of-speech tagging of the Web 1T 5-gram Corpus under the license given on the attached CD.

2.2.5 Macro for SPSS computing Krippendorff's alpha

Krippendorff's alpha is a standard reliability measure proposed by Krippendorff (2004). Here it was used to check the reliability of data provided by human judges for evaluation, and it was calculated using a macro written for SPSS[5], by Hayes and Krippendorff (2007).

2.2.6 SemEval-2 Task #9 dataset

The SemEval-2 Task #9 dataset[6] contains the training data, the testing data and the official scorer software for the SemEval-2 Task #9 (Butnariu et al., 2009), and is available under the Creative Commons Attribution 3.0 Unported license[7]. It was utilised for the development and evaluation of the methods described in Chapter 4.

[5] http://www.spss.com/
[6] https://docs.google.com/View?docid=dfvxd49s_35hkprbcpt
[7] http://creativecommons.org/licenses/by/3.0/

2.3 Web search engines

2.3.1 Google

According to its Terms of service[8], the Google search engine can only be used through the interface provided by Google and cannot be accessed through any automated method, unless given explicit permission by Google. Therefore this dissertation was registered for the University Research Program for Google Search[9], and Web search with Google was conducted through it.

2.3.2 Yahoo!

Search with Yahoo! was done using the Yahoo! Search Web Services SDK[10] (version 2.12), which provides easy access to the Yahoo! search engine from various languages, under BSD License[11].

[8] http://www.google.com/accounts/TOS
[9] http://research.google.com/university/search/
[10] http://developer.yahoo.com/search/web/webSearch.html
[11] http://www.opensource.org/licenses/bsd-license.php

3 Interpretation of noun compounds

There is much ongoing interest in the interpretation of noun compounds, since their interpretation is crucial for many NLP tasks. This is also shown by the fact that in the SemEval workshop in 2007, organized by the Special Interest Group on the Lexicon (SIXLEX) of the Association for Computational Linguistics (ACL), one of the tasks proposed dealt with the Classification of Semantic Relations between Nominals (Girju et al., 2007), and in the SemEval-2 workshop in 2010 one of the tasks was on The Interpretation of Noun Compounds Using Paraphrasing Verbs and Prepositions (Butnariu et al., 2009).

3.1 Inventory-based approaches

There are some linguistic theories, which suggest that noun compounds can be divided into a small number of categories based on the semantic relations between their nouns. Levi (1978), for example, claims that noun compounds can only be created through the processes of recoverable predicate deletion and nominalization, and the relation between the nouns of a noun compound can only come from a small set of semantic relations. She proposes nine recoverably deleteable predicates for noun compounds created by recoverable predicate deletion and four verb roles for noun compounds created by nominalization, and claims that these are the only relations that can hold between the nouns of a noun compound (see Table 1).

Warren (1978), through a large-scale study of the Brown Corpus (Francis, 1979), proposes a hierarchical classification of non-verbal-nexus noun compounds based on their semantics with four hierarchical levels. Six major semantic classes are defined at the top level, and then further divided into minor semantic classes, main groups and subgroups (see the major semantic classes in Table 1).

Many previous approaches are based on these theories and aim to interpret noun compounds using a small number of abstract relational categories. Most of these methods propose general relational categories for dealing with a wide range of texts, but there are also methods that propose categories for dealing with texts of a specific domain, such as Rosario and Hearst

(2001) for the biomedical domain. The number of the defined relational categories is typically between 5 and 40.

For example, Rosario and Hearst (2001) propose 18 abstract classes (see Table 1) and apply a standard machine learning algorithm with a domain-specific lexical hierarchy to classify noun compounds from biomedical texts.

Nastase and Szpakowicz (2003) propose a method employing machine learning tools to place noun compounds into clusters. This is based on features extracted from WordNet and Roget's Thesaurus and uses 30 clusters, which are grouped into 5 super-categories (see Table 1).

Moreover, Girju et al. (2004) present an approach to automatically classify nominalized noun phrases, which include nominalized noun compounds, using Support Vector Machines (SVM). They use a set of 35 semantic relations (see Table 1), which was originally presented in Moldovan et al. (2004).

Girju et al. (2005) present supervised models based on linguistic features using WordNet and word sense disambiguation, and test them both on the eight prepositional paraphrases described by Lauer (1995), and the 35 semantic relations used by Girju et al. (2004).

The methods in this category, however, have been criticised for numerous reasons. Although they have the advantage of capturing the generalization of relations in noun compounds, they are constrained by the small number of categories they define (Butnariu et al., 2009). One of the most influential critiques is Downing (1977) who argues that there are so many possible noun compound relations that it is impossible to list all, and that there are many relations that do not fit into any of the standard relationship categories. She also claims that with a limited number of categories, the categories can be ambiguous and noun compounds with different relationships can be assigned to the same category. Furthermore, it is hard to determine which set of relational categories would be best for classifying the relations between noun compounds, since linguists specialized in noun compounds disagree even on the main categories (Lauer, 1995).

3.2 Paraphrasing approaches

A solution for the above mentioned problems is to employ paraphrases for the interpretation of noun compounds instead of predefined abstract semantic categories, with verbs and prepositions as possible paraphrases. By using paraphrases, the number of possible categories is only limited by the vocabulary of the used language, even subtle differences in meaning can be identified and there are no noun compounds that do not fit into any category (Butnariu et al., 2009). Therefore, paraphrasing methods have become popular in recent years.

One of the early automatic noun compound interpretation methods that involves paraphrases is proposed by Lauer (1995). Although using paraphrases, he only uses a small set of eight prepositional paraphrases (see Table 1). Therefore this method is actually inventory based, and has the same problems as the other such methods. A further problem is that there are also noun compounds that cannot be paraphrased by a preposition (Butnariu et al., 2009). On the other hand, its advantage is that it is easy to identify noun-preposition co-occurrences in a corpus with unsupervised methods. This model was then further developed by Lapata and Keller (2005), using Web queries to estimate frequencies of (noun1, preposition, noun2) trigrams.

Nakov and Hearst (2006) and Nakov (2007) propose a method of noun compound interpretation by issuing exact Web search engine queries and extracting a list of paraphrases with their frequencies from the resulting snippets for each noun compound. They argue that using a single paraphrase is not enough to obtain the fine-grained semantic relation between the nouns of a noun compound. Hence, they use paraphrase distributions.

There have also been numerous methods proposed to solve The Interpretation of Noun Compounds Using Paraphrasing Verbs and Prepositions Task of the SemEval-2 workshop (Butnariu et al., 2009). Given a list of suitable paraphrases for each noun compound, this is a task to return a ranked list of paraphrases for each noun compound based on their aptness.

Nulty and Costello (2010) argue that people tend to use general, frequent paraphrases for noun compounds instead of more detailed ones. Therefore they proposed a method based on paraphrase co-occurrence statistics obtained from the training data favouring general paraphrases over less general ones. They managed to achieve an average Spearman's rank

correlation coefficient (this was the official measure) of 0.441, which is only slightly lower than the score of the best performing system.

The best result was obtained by the system proposed by Wubben (2010), which employs a machine learning classifier based on features that were taken from WordNet, the provided training data and the Web 1T 5-gram Corpus. It achieves an average Spearman's rank correlation coefficient of 0.45.

Due to the above listed advantages of paraphrasing methods over inventory-based ones, I believe that paraphrasing methods are more suitable for noun compound interpretation than inventory-based methods. Therefore the methods discussed in this dissertation use paraphrases to interpret noun compounds instead of defining a small set of possible relational categories.

Author(s)	**Proposed relational categories**
Levi (1978)	recoverably deleteable predicates: CAUSE, HAVE, MAKE, USE, BE, IN, FOR, FROM, ABOUT verb roles: ACT, PRODUCT, AGENT, PATIENT
Warren (1978)	POSSESSION, LOCATION, PURPOSE, ACTIVITY-ACTOR, RESEMBLANCE, CONSTITUTE (only the major semantic classes)
Lauer (1995)	OF, FOR, IN, AT, ON, FROM, WITH, ABOUT
Rosario and Hearst (2001)	SUBTYPE, ACTIVITY/PHYSICAL PROCESS, PRODUCES (ON A GENETIC LEVEL), CAUSE (1-2), CHARACTERISTIC, DEFECT,PERSON AFFLICTED, ATTRIBUTE OF CLINICAL STUDY, PROCEDURE, FREQUENCY/TIME OF (2-1), MEASURE OF, INSTRUMENT, OBJECT, PURPOSE, TOPIC, LOCATION, MATERIAL, DEFECT IN LOCATION
Nastase and Szpakowicz (2003)	CAUSALITY (cause, effect, detraction, purpose), PARTICIPANT (agent, beneficiary, instrument, object property, object, part, possessor, property, product, source, whole, stative), QUALITY (container, content, equative, material, measure, topic, type), SPATIAL (direction, location at, location from, location), TEMPORALITY (frequency, time at, time through)
Moldovan et al. (2004), Girju et al. (2004) and Girju et al. (2005)	POSSESSION, KINSHIP, ATTRIBUTE-HOLDER, AGENT, TEMPORAL, DEPICTION-DEPICTED, PART-WHOLE, IS-A (HYPERNYMY), ENTAIL, CAUSE, MAKE/PRODUCE, INSTRUMENT, LOCATION/SPACE, PURPOSE, SOURCE, TOPIC, MANNER, MEANS, ACCOMPANIMENT, EXPERIENCER, RECIPIENT, FREQUENCY , INFLUENCE, ASSOCIATED WITH, MEASURE, SYNONYMY, ANTONYMY, PROBABILITY, POSSIBILITY, CERTAINTY , THEME, RESULT, STIMULUS , EXTENT, PREDICATE, MEASURE, RESULT

Table 1: Previously proposed relational categories for noun compound interpretation.

4 Methods for noun compound interpretation

This chapter presents the noun compound interpretation algorithms in detail. The first section gives information on methods for interpreting two-noun noun compounds, and the second describes how the methods displayed in the first section can be applied to interpret noun compounds with more than two nouns.

4.1 Interpreting two-noun noun compounds

The main aim of this dissertation was to write a general paraphrasing method for noun compound interpretation, that, given a list of noun compounds as its input, returns a ranked list of paraphrases for each of them; this is described in Section 4.1.1. Based on the Web validation part of the general paraphrasing method, another method suitable to solve the SemEval-2 Task #9 (Butnariu et al., 2009) can easily be created; this is described in Section 4.1.2. The evaluation of the results for both methods is given in Chapter 6. The Java source code of the implementation of these programs can be found on the attached CD; the main function for the general paraphrasing method in GeneralParaphrasingMethod.java, and the main function for the method for the SemEval-2 Task #9 in SemEval2Task9Method.java

4.1.1 A general paraphrasing method for two-noun noun compounds

The main idea is to try to find paraphrases of noun compounds in the corpora and return a ranked list of paraphrases for each based on some scoring measure. Possible paraphrases are verbs, prepositions, and combinations of the two. However, when a preposition without a verb is found as a paraphrase, it is extracted as if it were connected to the verb *be* instead of standing by itself. Therefore, all the paraphrases extracted are verbs with or without prepositions. It is done so, because if the paraphrase is just a preposition, it essentially has the same meaning as if it were with the verb *be*, therefore their frequencies should be counted together. Thus, if a paraphrase *for* is found for *water bottle*, it is extracted as if it was *be for*.

In almost all noun compounds, the second noun is the head and the first the dependent, defining a property for the head. and the compound of the two nouns syntactically behave as the head would (Nakov and Hearst, 2006, Lauer, 1995)[12]. It will be assumed throughout this work that this holds for the noun compounds to be interpreted. Therefore only such paraphrases are searched for, whose subject is the second noun of the noun compound and whose object is the first noun of the noun compound. When *be for* is said to be a good paraphrase for *water bottle*, it means that *bottle that is for water* captures the meaning of a *water bottle*.

Technically, the first noun of a noun compound is not the actual direct object of the paraphrase if it also includes a preposition; it is the direct object of the preposition in the paraphrase. Still, to make things easier, it will be referred to as the object of the paraphrase, in the rest of this dissertation. (The second noun is always the subject of the paraphrase, since all the prepositions without a verb are extracted with the verb *be*.)

There are two main versions given the different types of paraphrase extraction and search; a description of these will follow.

4.1.1.1 The two main versions

The first version searches for actual paraphrases for the input noun compounds in the corpora. For this, it reads through the corpora and counts the frequency of all occurring (subject, paraphrase, object) triples, where:

- *paraphrase* is a verb, *subject* is its subject, and *object* is its direct object
- *paraphrase* is a verb with a preposition, *subject* is its subject, the preposition is the indirect object of the verb, and *object* is the direct object of the preposition
- *paraphrase* is a single preposition, which is a non-clausal modifier of *subject*, and *object* is the direct object of the preposition

This is very similar to the extraction method used by Nakov (2007), when extracting features from the parsed snippets of Web search query results for paraphrasing noun compounds.

[12] An exception, for example, is *attorney general* and *fetucinne arabiata* (Lauer, 1995).

After this paraphrase extraction, for each noun compound it searches for those extracted (subject, paraphrase, object) triples where the second noun of the noun compound is the subject and the first noun of the noun compound is the object. This results in a list of paraphrases for each noun compound including their frequency with that noun compound, which is counted as their score.

The logic behind the second version is that if there is a paraphrase that frequently has the second noun of the noun compound as subject, and it frequently has the first noun of the noun compound as object, then it is likely that this paraphrase is a suitable one for the noun compound. Therefore, when reading through the corpora, this version counts the frequency of all occurring (subject, paraphrase) pairs, where:

- *paraphrase* is a verb, and *subject* is its subject
- *paraphrase* is a verb with a preposition, preposition is the indirect object of the verb and *subject* is the subject of the verb
- *paraphrase* is a preposition, which is the non-clausal modifier of *subject*

It also counts the frequency of all occurring (paraphrase, object) pairs, where:

- *paraphrase* is a verb, and *object* is its direct object
- *paraphrase* is a verb with a preposition, the preposition is the indirect object of the verb and *object* is the direct object of the preposition
- *paraphrase* is a preposition and *object* is its direct object

Then, for each noun compound, this version searches for such extracted (subject, paraphrase) pairs and (paraphrase, object) pairs, where the second noun of the noun compound is the subject, and the first noun of the noun compound is the object. This results in two lists of paraphrases for each noun compound, one for the second noun (subject), and one for the first noun (object). To compile the list of suitable paraphrases for the noun compound from these two lists, those paraphrases are searched for that appear in both of them; these are then included in the paraphrase list for the noun compound and their score is calculated from the (subject, paraphrase) and (paraphrase, object) frequencies (see Section 4.1.1.4).

This version has the advantage that paraphrases can be found much more easily, because finding a (subject, paraphrase, object) triple for a noun compound is rare compared to finding just one of the nouns at a time with a subject or an object. Therefore, this version is likely to return more possible paraphrases than the other. On the other hand, it is also more likely that it returns incorrect results. Consequentially, this version has better recall, but worse precision.

In order to make these methods more efficient, all words are lemmatized when extracting the triples and the pairs from the corpora. This means that the singular form of the nouns and the infinitive form of the verbs are extracted, irrespective of the form they have in the sentence. Hence, from the sentence “These bottles are for water”, (bottle, be for, water) is extracted with the first version. From the same sentence, the (subject, paraphrase) pair (bottle, be for) and the (paraphrase, object) pair (be for, water) are identified with the other version. In addition, when searching for possible paraphrases for noun compounds, the search is conducted with the lemmatized nouns of the noun compound. The lemma for each word is obtained from WordNet, through JWNL.

With both methods, verb particles are also identified and extracted with the verbs, as described in Section 4.1.1.5. Verb particles are those words wherewith verbs form phrasal verbs. For example, *take off* is a phrasal verb, with *off* being its particle. In the rest of the dissertation, the term verb will be used for a verb with or without particle, and prepositions will be used to denote those prepositions that are not verb particles.

These two versions in the following will be referred to as the subject-paraphrase-object-triples version, and the subject-paraphrase-and-paraphrase-object-pairs version, respectively.

4.1.1.2 The corpora used

In order to search for paraphrases, three corpora were taken into consideration; two static corpora, namely the British National Corpus and the Web 1T 5-gram Corpus together with the Web. All of them have their advantages and disadvantages, and have been used for many natural language processing tasks, including noun compound interpretation.

The power of using the Web lies in the vast amount of data available, as with more data, better results can be obtained (Kilgarriff, 2007). This data can be most easily accessed

through commercial search engines, like Google or Yahoo!. However, as is pointed out by Nakov et al. (2007) and Kilgarriff (2007), using commercial search engines for natural language processing tasks has several drawbacks. First, queries cannot have any linguistic restrictions. Second, the results are instable over time (even running the same query twice, immediately one after the other, can result in different results). Third, search engines can be inconsistent with Boolean logic. Fourth, they ignore punctuation. Fifth, they usually have limitations on the number of queries that can be issued given a period of time[13]. Sixth, they usually have a constraint on the number of pages returned per queries[14]. Further, if page hit counts are used, two additional detrimental effects arise. First, an exact page hit count for an n-gram is not equal to the frequency of that n-gram on the Web. Second, page hit counts returned by these search engines are actually estimates rather than exact counts. Still, Nakov (2007) and Nakov and Hearst (2005) suggest that models using Web search engine statistics are relatively stable, and that resorting to such models is not "bad science". Furthermore, the results of this dissertation indicate that Web search engine statistics can prove useful for noun compound interpretation.

As opposed to using the Web through commercial search engines, static corpora do not have these disadvantages. On the other hand, their drawback is that they are much smaller than the Web. There are several major differences between the British National Corpus and the Web 1T 5-gram Corpus, as well. A major advantage of the latter is that it is much larger than the former; it was generated from more than 1 trillion words, whereas the former consists of approximately 100 million words. On the other hand, the British National Corpus consists of full sentences rather than simply n-grams as the Web 1T 5-gram Corpus, which results in much higher accuracy when automatically parsed or tagged (as is illustrated in the next section, their parsing or tagging is needed to extract the grammatical relations inside them). Furthermore, the British National Corpus is balanced and representative of current English, which is neither true for the Web 1T 5-gram Corpus nor for the Web.

The best solution would be to utilising the corpus provided by the Web, but without having to use commercial search engines. Unfortunately, it is not feasible to do in this dissertation.

[13] At the time of writing the limits were 5000 queries per day per IP address for Yahoo! using Yahoo! Web Search Web Services (see http://developer.yahoo.com/search/web/webSearch.html), and 1 query per second for Google using the University Research Program for Google Search (see http://research.google.com/university/search/docs.html).
[14] The constraint on the number of pages returned was 1000 for both Google and Yahoo! at the time.

Nevertheless, the Web 1T 5-gram Corpus provides a good solution for handling huge amounts of data from the Web without having to rely on commercial search engines, since it was generated from more than 1 trillion words of Web page texts (although it is just a proportion of the whole Web).

All the three abovementioned corpora have benefits, and they are all useful in solving Natural Language Processing tasks. Therefore, it was decided to combine the usage of static corpora and the Web during this work. While the British National Corpus and the Web 1T 5-gram Corpus are employed in the search for paraphrases, the results are validated through Web search engine queries (as described in Section 4.1.1.9). Two search engines were chosen for validation; Google and Yahoo!. It may seem strange that the Google search engine was selected for the validation of the results when one of the used corpora, the Web 1T 5-gram Corpus, was created by Google itself from Web pages. However, the Web 1T 5-gram Corpus was not generated from the entire Web, just from a part of it. Moreover, the data for it was collected more than 4 years ago (in January 2006), and since then, the Web has grown a lot. Furthermore, it only contains n-grams with frequencies above 40. Therefore, very infrequent noun compounds not occurring 40 times with their paraphrases do not show up in this corpus, but can be found on the Web (however, these infrequent (noun compound, paraphrase) pairs can be extracted from the Web 1T 5-gram Corpus even if their actual frequency is lower than 40 if, for example, synonyms of the nouns are also searched for. Therefore it is important that such (noun compound, paraphrase) pairs can be validated with Web search engines). This legitimises the Google search engine as a validation method.

4.1.1.3 Pre-processing of the corpora

In order to be able to extract (subject, paraphrase), (object, paraphrase) pairs and (subject, paraphrase, object) triples, the grammatical relations among the words in the corpus need to be identified. With such large corpora as the ones analysed here, manual parsing is infeasible. Therefore automated parsing methods were applied.

As already described in Chapter 2, the British National Corpus available on the Curlew server had already been parsed with the C&C CCG parser before, so further pre-processing was not needed. But, as it is stated in (Clark and Curran, 2007) and experienced during the work, the C&C CCG parser often makes mistakes in differentiating the indirect object (iobj) and non-

clausal modifier (ncmod) relations for verbs and the recall for the indirect object relation is only 65.63%. Therefore, when using the British National Corpus, instead of just using those prepositions in the paraphrases that are parsed as indirect object of verbs, prepositions parsed as non-clausal modifiers of verbs are also tested for. This modification resulted in an improved outcome. Moreover, when extracting (subject, paraphrase, object) triples, or (subject, paraphrase) and (paraphrase, object) pairs, the part-of-speech tags of the words in them are checked; subjects and objects should be common nouns, and paraphrases should be verbs with or without preposition. This is done using the tags returned by the parser.

The Web 1T 5-gram Corpus available on the Curlew server was not previously parsed though. Automatically parsing this corpus encounters some problems. First, the n-grams are not complete sentences, and in many cases even for humans it is hard to determine what grammatical relations hold between the words of an n-gram without context. Therefore, automatically parsing them would result in many errors, especially when shorter ones are concerned. Because of this, only 4-grams and 5-grams should be considered for parsing. The second problem is that the C&C CCG parser is able to process about 150 n-grams in one second when it is run on the Curlew server. Since there are almost 2.5 billion 4-grams and 5-grams altogether, parsing all of them would take around half a year (in CPU time on a computer like the Curlew server).

Given the lack of that much time, an alternative approach was chosen, namely tagging the corpus since it “only” took about 2 days. As tagging also involves many errors on short n-grams, only the 4-grams and the 5-grams were used. Although the grammatical relations cannot be directly obtained from a tagged text, the relations between the tagged words can be assumed from part-of-speech patterns. The part-of-speech patterns selected with the 4-grams were:

```
(1)         noun vbn prep noun
(2)         noun verbNotVbn det|adj noun
(3)         noun verbNotVbn prep noun
(4)         noun be1 vbg noun
(5)         noun have1 vbn noun
(6)         noun that|modal verbNotVbnNotVbg noun
(7)         noun prep det|adj noun
```

, whereas the ones with 5-grams were:

```
(8)         noun vbn prep det|adj noun
(9)         noun vbn prep prep noun
(10)        noun verbNotVbn det|adj adj noun
(11)        noun verbNotVbn prep det|adj noun
(12)        noun verbNotVbn prep prep noun
(13)        noun have2 been vbg noun
(14)        noun be1 vbg det|adj noun
(15)        noun be1 vbg prep noun
(16)        noun have1 vbn det|adj noun
(17)        noun have1 vbn prep noun
(18)        noun be2 vbn prep noun
(19)        noun that|modal verbNotVbnNotVbg det|adj noun
(20)        noun that|modal verbNotVbnNotVbg prep noun
(21)        noun that|modal be3 vbg noun
(22)        noun that|modal have2 vbn noun
(23)        noun prep det|adj adj noun
```

Table 2 contains the meaning of the expressions included in the patterns. With each pattern it was assumed that the verb with the possible following prepositions forms a paraphrase, the first word of the n-gram being its subject and the last word of the n-gram being its object. The according (subject, paraphrase, object) triple, (subject, paraphrase) and (paraphrase, object) pairs were extracted.

Present continuous and past continuous paraphrases are identified by patterns (4), (14), (15) and (21). Patterns (5), (16), (17) and (22) are designed to match the paraphrases in present perfect and past perfect tense. Patterns (6), (19) and (20) identify future simple, present simple paraphrases with modals, and present simple or past simple paraphrases. Pattern (21) is also used to identify future continuous, present continuous with modals, and present or past continuous. Pattern (22) also identifies future perfect, present perfect with modals, and present or past perfect. Pattern (13) identifies present perfect continuous and past perfect continuous verbs. Passive paraphrases are identified by patterns (1), (8), (9) and (18). There are no patterns identifying passive paraphrases without a preposition, as passive paraphrases cannot have direct objects. Thus, the second noun of the pattern can only be connected to them by a

preposition. The last pattern with both the 4-grams and the 5-grams identifies paraphrases consisting of a single preposition and without a verb. In these cases the paraphrase extracted is the verb *be* with the prepositions, instead of only the preposition. All the other patterns are meant to identify paraphrases in simple present and simple past, in active voice.

Words matching the expressions `det`, `adj`, `that` or `modal` do not form a part of the verb; therefore they are ignored. Words matching the patterns be1, be2, be3, been, have1 and have2 are words that are part of the verb, but only determine the tense and the voice. Hence, they are not included in the paraphrase, either.

In a substantial part of the n-grams, all the words start with a capital letter, or all the words are entirely written in capital letters. This is probably due to the fact that many Web page titles and article titles are written in one of these forms. The problem with these n-grams is that the C&C CCG parser simply tags each word as a proper noun, which is very rarely correct. Therefore, in order for these n-grams to be properly tagged as well, all the n-grams were converted to lowercase before tagging. This, however, results in not recognizing proper nouns correctly. This problem does not apply to those proper nouns that cannot be common nouns (such as most human names among many others) or those which are not part of any of the input noun compounds. Those noun occurrences, where a common noun is used as (part of) a proper noun, are rare in most cases compared to their occurrence as common nouns (an exception for example is the noun *apple*, as described later). It is even less likely that these proper noun occurrences are found in such n-grams from which a paraphrase for a noun compound is extracted. For example, consider the noun compound *apple cake*, where the noun *apple* can also be understood as a proper noun referring to Apple Inc. (it probably appears more often as a proper noun in many texts). The noun *cake* can also represent a proper name, although much less frequently (for example a company called Cake Inc.). But, when searching for paraphrases for *apple cake*, it is rather unlikely that *apple* or *cake* should function as proper nouns in the n-gram that provides a paraphrase for it, since Apple Inc. and cakes, Cake Inc. and apples, Apple Inc. and Cake Inc. are very probably not associated with each other[15]. Nevertheless, most noun compounds contain nouns appearing far less frequently as proper nouns, than *apple* occurs as a proper noun. Therefore, this should not represent a

[15] It is possible that the Cake Inc. actually does something with apples or Apple Inc. started to bake cakes or there is a connection between the two incorporations, but it is unlikely, and even if it is the case, it is likely that the correct paraphrases for *apple cake* are with higher frequency.

problem in most cases. Hence, it is assumed that the advantage of a substantial gain in extra information by converting all n-grams to lowercase weighs much higher than the disadvantage represented by this problem.

To reduce the amount of false information extracted, only those n-grams were considered in which each word consisted of only alphanumerical characters. From the n-grams matching one of the above defined patterns, only those were used in which the word tagged as verb was found in the WordNet database as verb. When utilising this corpus, the (subject, paraphrase) and (paraphrase, object) pairs are extracted when a (subject, paraphrase, object) triple is found, rather than searching for separate occurrences (the difference between the two is that the latter also extracts (subject, paraphrase) pairs, where no object is specified, and it also extracts (paraphrase, object) pairs, where no subject is specified). The reason for this is that, as stated before, the tagger makes many errors on short n-grams. Moreover, identifying part-of-speech patterns of length 2 or 3 in the 4-grams and 5-grams also entails too many errors. Because of this, those (subject, paraphrase) and (object, paraphrase) pairs are not found, when there is no object or subject connecting to the verb.

Expression	**Meaning (based on tags, except where a list of words is given)**
noun	a common noun
verbNotVbn	a verb that is not past participle
verbNotVbnNotVbg	a verb that is neither past participle nor in progressive form
vbn	a verb that is a past participle
vbg	a verb in progressive form
be1	one of the following words: *am, is, are, was, were*
be2	one of the following words: *am, is, are, was, were, being*
be3	one of the following words: *am, is, are, was, were, be*
been	the word *been*
have1	one of the following words: *has, have, had, having*
have2	one of the following words: *has, have, had*
that	one of the following relative pronouns: *that, which, who*
modal	a modal verb
prep	a preposition

Table 2: Expressions used in the part-of-speech patterns

4.1.1.4 Scoring methods

The score for the subject-paraphrase-object-triples version is rather simple; when searching for paraphrases for a noun compound, it returns a list of paraphrases from the (subject, paraphrase, object) triples with frequencies. These frequencies are then displayed as scores.

Nevertheless, applying simply frequencies with the subject-paraphrase-and-paraphrase-object-pairs version carries problems; whether the noun is the subject or the object, the most frequent verbs going along with all nouns are very common ones, such as *be*, *do* or *make*. When the (subject, paraphrase) and (paraphrase, object) frequencies are combined, the highest scores are achieved by the paraphrases with those verbs not typical of the noun compounds and usually not suitable for paraphrasing them[16]. To avoid this, a possible solution is to use a word association measure for the extracted pairs instead of frequencies, in order to find the most typical verbs for nouns (both in (paraphrase, object) and (subject, paraphrase) relation).

One such word association measure is mutual information[17] described by Church and Hanks (1989) and Church et al. (1991), which compares the probability of observing two words together to the probabilities of observing them independently. Church et al. (1991) also relate how it can be used for parsed texts. Here, however, a slightly changed version is used, since (subject, paraphrase) and (paraphrase, object) pairs are extracted separately, not as (subject, paraphrase, object) triples. The mutual information of x being the paraphrase and y being its subject can be observed here:

$$I(\text{paraphrase(x)}, \text{subject(y)}) = log_2 \frac{P\big(\text{paraphrase(x)}, \text{subject(y)}\big)}{P\big(\text{paraphrase(x)}\big) * P\big(\text{subject(y)}\big)}$$

$$\approx log_2 \frac{\frac{f\big(\text{paraphrase(x)}, \text{subject(y)}\big)}{N}}{\frac{f\big(\text{paraphrase(x)}\big)}{N} * \frac{f\big(\text{subject(y)}\big)}{N}}$$

$$= log_2 \frac{f\big(\text{paraphrase(x)}, \text{subject(y)}\big) * N}{f\big(\text{paraphrase(x)}\big) * f\big(\text{subject(y)}\big)}$$

[16] Although for many noun compounds the verb *be* is a good paraphrase, it makes no sense to classify it as the best paraphrase for every noun compound, since, usually, there are more suitable ones.

[17] First, the authors called it association ratio, since due to its lack of symmetry it is actually different from mutual information, Further, f(x,y) is not necessarily smaller than f(x) and f(y). In many other publications, such as Manning and Schütze, (2000), it is called pointwise mutual information. Here, it shall be referred to as mutual information, as the authors used the same denomination in later articles.

where $P(\text{paraphrase(x)}, \text{subject(y)})$ is the probability that the paraphrase is x and the subject is y in a (subject, paraphrase) pair; $P(\text{paraphrase(x)})$ and $P(\text{subject(y)})$ are the probabilities that the paraphrase is x in a (subject, paraphrase) relation and the subject is y in a (subject, paraphrase) relation, respectively. The probabilities of the abovementioned events are estimated in this application through their frequencies in the corpus; these frequencies are denoted as $f(\text{paraphrase(x)}, \text{subject(y)})$, $f(\text{paraphrase(x)})$ and $f(\text{subject(y)})$ respectively, while N is the number of all the (subject, paraphrase) relations found. The mutual information of x as the paraphrase and y as its object is calculated very similarly.

In the case of a genuine association between the paraphrase x and its subject y, their mutual information is much larger than 0. When there is no interesting relationship between them, then it is around 0; considering a genuine dissociation between them (they occur very rarely together, much more rarely than by chance), it is much smaller than 0. The mutual information of a (subject, paraphrase) pair and the mutual information of a (paraphrase, object) pair is then multiplied or added together to form a single score for the (subject, paraphrase, object) triple. However, since a mutual information below 0 is equivalent to a genuine dissociation between the words, only those paraphrases, with a mutual information of the (subject, paraphrase) pair and the (paraphrase, object) pair both above 0 are considered for a noun compound. Furthermore, Church and Hanks (1989) note that mutual information is unstable for very small counts, therefore paraphrases with a (subject, paraphrase) or (paraphrase, object) frequency of at most 5 are also discarded (this is the same cut-off as described by Church and Hanks (1989)).

4.1.1.5 Prepositions and verb particles

As stated before, the prepositions that are simultaneously verb particles are treated differently than those that are not verb particles. The term preposition here denotes those prepositions that are not verb particles.

In order to speed up the search for paraphrases and without suffering from tagging errors of prepositions, a predefined list of prepositions is used (this is possible for prepositions, as there are only a manageable number of them, but for example with verbs it could not be done). A list of prepositions was compiled automatically by extracting those words from the British National Corpus tagged as *IN* or *TO*, with their frequencies. Due to tagging errors, only those

words occurring at least 100 times in the corpus are considered, which only requires 1 appearance in 1 million words. Because subordinating conjunctions are also tagged with *IN*, those were then manually deleted from the list. This resulted in a list of 67 prepositions which can be found with frequency counts in Appendix A.

If a paraphrase with or without preposition is encountered, a (subject, paraphrase) pair is then extracted, for which the preposition is not part of the paraphrase. Moreover, if the paraphrase does contain a preposition, a (subject, paraphrase) pair, with a preposition-including paraphrase is saved too. The (subject, paraphrase) pairs without the preposition are extracted, since from the sentence “The professor teaches anatomy at a university”, it seems reasonable to extract the (subject, paraphrase) pair (professor, teach); it can then be paired with the (paraphrase, object) pair (teach, anatomy), for example, to form the paraphrase *teach* for *anatomy professor*. It is necessary to save each (subject, paraphrases) pair with all its prepositions too, because without this the subject-paraphrase-and-paraphrase-object-pairs version would not find paraphrases including prepositions. For example, if only the (subject, paraphrase) pair (professor, teach) is extracted from the above mentioned sentence, there will be no matching (subject, paraphrase) pair for the (object, paraphrase) pair (teach at, university). Obviously, this is not correct, since the example sentence provides the paraphrase *teach at* with the noun compound *university professor*. The (paraphrase, object) pairs and the (subject, paraphrase, object) triples are not specially treated as (subject, paraphrase) pairs, regardless of whether the paraphrase contains a preposition or not. For example, from the sentence “The girl drew a picture with a pencil”, the subject-paraphrase-and-paraphrase-object-pairs version extracts the (subject, paraphrase) pairs (girl, draw), (girl, draw with) and the (paraphrase, object) pairs (draw, picture), (draw with, pencil). In the case of the same sentence, the other version extracts the (subject, paraphrase, object) triples (girl, draw, picture) and (girl, draw with, pencil).

The paraphrase list is used to identify verb particles, too. A word is considered a particle of a verb if it is included in the preposition list, it is headed by the verb, and it has no dependents.

4.1.1.6 Passive paraphrases

Passive paraphrases are different from other paraphrases, because their surface subject is actually their underlying object. Therefore a (subject, paraphrase) pair with a passive

paraphrase and without a preposition (as noted before, the term preposition does not contain verb particles here), in fact has the same meaning (at least from the point of view of this dissertation) as the (paraphrase2, object) pair, where *paraphrase2* is the same as *paraphrase* except that it is active. For example, the (subject, paraphrase) pair (pizza, be eaten) has the same meaning as the (paraphrase, object) pair (eat, pizza). It makes sense to count their frequency together, as (paraphrase, object) pairs, rather than separately because of the following two reasons: First, they only differ in their structure, but have the same meaning. Second, since passive verbs cannot have direct objects[18], there will be no (paraphrase, object) pairs with a passive paraphrase and no preposition. Therefore there is no value in storing such (subject, paraphrase) pairs, since no matching (paraphrase, object) pairs can be found. Thus, whenever a (subject, paraphrase) pair is extracted with a passive paraphrase and no preposition, it is saved as a (paraphrase2, object) pair instead, with *object* as the original *subject* and *paraphrase2* as the active version of *paraphrase*. Since passive verbs cannot have direct objects, there can be no (paraphrase, object) pairs or (subject, paraphrase, object) triples with a passive paraphrase and no preposition. For example, from the sentence "The pizza was eaten", the subject-paraphrase-and-paraphrase-object-pairs version extracts the (paraphrase, object) pair (eat, pizza). The other version cannot extract any (subject, paraphrase, object) triples from this sentence.

Passive paraphrases with a preposition other than *by* are also a different case. As stated in the previous section, when a paraphrase has a preposition, a (subject, paraphrase) pair without the preposition is extracted too, not just the one with the preposition. Thus, in the event of a passive paraphrase with a preposition other than *by* (and with a subject), a (subject, paraphrase) pair both with and without the preposition is extracted. From these two pairs the one without the preposition is treated as described in the previous paragraph, whereas the other is treated normally. With such paraphrases, (paraphrase, object) pairs and (paraphrase, object, subject) triples are given normal treatment. For example, from the sentence "This house was built from stone", the subject-paraphrase-and-paraphrase-object-pairs version extracts the (subject, paraphrase) pair (house, be built from) and the (paraphrase, object) pairs

[18] They can have second objects though, which are easily confused with direct objects. A typical verb that has a second object is *give*, for example. The C&C CCG parser usually recognizes them correctly as second objects. In case the parser made a mistake, and there is an occurrence of a passive paraphrase with a direct object in the sentence, than that paraphrase is discarded.

(build, house) and (be built from, stone). Contrariwise, the other version extracts the (subject, paraphrase, object) triple (house, be built from, stone) from the same sentence.

It is also different when a passive paraphrase includes a preposition *by* that refers to a direct object, in which case the direct object of *by* is the underlying subject of the paraphrase. Therefore, a (paraphrase, object) pair with a passive paraphrase and a preposition *by* in effect has the same meaning as the (subject, paraphrase2) pair, where *paraphrase2* is the same as *paraphrase* except that it is active and does not have the preposition *by*. For example, the (paraphrase, object) pair (be built by, architect) has the same meaning as the (subject, paraphrase) pair (architect, build). Thus, it makes sense to count their frequencies together. So, if a (paraphrase, object) pair is encountered where the paraphrase is as described, it is instead extracted as a (subject, paraphrase2) pair where *subject* is the same as *object* and *paraphrase2* is the active version of *paraphrase* without the preposition *by*. In the case of a passive paraphrase with the preposition *by*, only a (subject, paraphrase) pair without the preposition is extracted whereas the one including the preposition is not. This is because there would be no matching (paraphrase, object) pairs, since no such pairs are saved with a passive paraphrase going with the preposition *by*. The one extracted without the preposition is treated as described in the first paragraph of this section. In the case of such paraphrases, a (subject, paraphrase, object) triple has the same meaning as the (subject2, paraphrase2, object2) triple, where *subject2* is equal to *object*, *object2* is equal to *subject*, and *paraphrase2* is the active version of *paraphrase* without the preposition *by*. Hence, their frequency should be counted together. Therefore, whenever a (subject, paraphrase, object) triple is identified with a passive paraphrase and a preposition *by*, it is saved as a (subject2, paraphrase2, object2) triple instead, where *subject2* is equal to *object*, *object2* is equal to *subject*, and *paraphrase2* is the active version of *paraphrase* without the preposition *by*. For example, from the sentence “This house was built by an architect”, the subject-paraphrase-and-paraphrase-object-pairs version extracts the (subject, paraphrase) pair (architect, build) and the (paraphrase, object) pair (build, house). Still speaking about the same sentence, the other version extracts the (subject, paraphrase, object) triple (architect, build, house).

Because of these conversions, the frequency counts for such (subject, paraphrase, object) triples (subject, paraphrase) and (paraphrase, object) pairs with passive paraphrases and the preposition *by*, are stored with their converted version. Therefore, in order to find paraphrases like this for noun compounds, both methods search for such paraphrases for the reverse noun

compound (the noun compound where the order of the nouns is changed; it might not be an actual noun compound, but this is not problematic) that are active and have no preposition. If such a paraphrase is found for the reversed noun compound, its passive version with the preposition *by* is then saved for the (not reversed) noun compound, with its score. That is, in order to find paraphrases for the noun compound *band concert* that are passive and have the preposition *by*, the subject-paraphrase-object-triples version searches for such extracted (subject, paraphrase, object) triples where the subject is *band*, the object is *concert* and the paraphrase is active and has no preposition. For example, if there is a triple (band, give, concert), the paraphrase *be given by* is then saved for *band concert* with the score of the (band, give, concert) triple. This works very similarly with the subject-paraphrase-and-paraphrase-object-pairs version too.

Passive verbs are identified in two ways; if the subtype slot of a non-clausal subject relation (ncsubj) is filled with the value *obj*, then the verb in that relation is passive. Furthermore, the C&C CCG parser fails to identify this many times. Consequentially, in case a verb is not identified as passive by the parser, it is determined whether the verb is in past participle form and has an auxiliary *be*; if so, then it is passive.

4.1.1.7 Ambitransitive verbs

Each English verb is either strictly intransitive, strictly transitive, or ambitransitive (sometimes also called labile) (Dixon and Aikhenvald, 2000), whereby the latter means that it functions both transitively and intransitively. Good examples for strictly transitives are *like* and *recognise*, for strictly intransitives *arrive* and *chat*, and for ambitransitive *break* and *read*. The Unaccusative Hypothesis, proposed by Perlmutter (1978) and then elaborated by Burzio (1986), proposes two subclasses of intransitive verbs; the unaccusative verbs being those with a surface subject acting as their underlying object, and the unergative verbs being those with a surface subject acting as their underlying subject. Although the claim that these two subcategories of intransitive verbs exist is widely accepted, since the introduction of the hypothesis there has been much debate about whether, as argued by Perlmutter (1978) as well as Perlmutter and Postal (1984), the distinction between the two classes is represented syntactically and can be fully determined semantically. Some researchers like Rosen (1984) do not agree that it can be fully determined semantically, whereas others, like Van Valin Jr

(1990), deny that it is syntactically encoded, while some, including Levin and Hovav (1995), support both.

The two categories cannot only be applied to intransitive verbs, but also to ambitransitive verbs; the patientive ambitransitive verbs are unaccusative in their intransitive use, so the semantic role of their subject in their intransitive use is equal to the semantic role of their object in their transitive use. Further, agentive ambitransitive verbs are unergative in their intransitive use, so the semantic role of their subject in their intransitive use is the same as is in their transitive use (Mithun, 2000) (they are called as S=O and S=A ambitransitives, respectively, by Dixon and Aikhenvald (2000), but here the names proposed by Mithun (2000) will be used). A typical patientive ambitransitive is *break*; the sentence "The window broke" actually means that someone or something broke the window, so the window is actually the object of the action and has the same semantic role as in the sentence "Someone broke the window". A typical agentive ambitransitive is *read*; in the sentence "She reads" *she* is truly the subject of the action and it has the same semantic role as in the sentence "She reads a book".

Since both in transitive and intransitive use the grammatical subject of agentive ambitransitives is their underlying subject, they do not represent a problem for this paraphrasing method; in both cases their underlying subject is correctly extracted as their subject. On the other hand, patientive ambitransitives represent a problem. If one such verb is used in its transitive form, its underlying subject is correctly extracted as its subject. However, in case of their intransitive use, their underlying object (which is their surface subject) is, contrariwise, incorrectly extracted as its subject. This can result in paraphrasing errors. For example, when using the subject-paraphrase-and-paraphrase-object-pairs version of the program and examining the sentence "The screen broke", screen is falsely extracted as the subject of broke, because it is actually its underlying object. In addition, in case the sentence "I broke my computer" is also encountered, and a paraphrase for *computer monitor* is searched for, this extraction mistake would result in finding the paraphrase *break* for *computer monitor*, which is obviously not correct. The subject-paraphrase-object-triples version has to deal with a very similar problem, too, if there is a patientive ambitransitive verb in intransitive use with an object connected to it by a preposition.

There is, however, a solution to this problem. Patientive ambitransitives in their intransitive use behave in the same way as passive verbs; their surface subject is their underlying object. Therefore, patientive ambitransitives in their intransitive form should be treated as if they were passive, which would solve the problem. A comprehensive list of these verbs is given by Levin (1993) in Section 1.1, which is used in this method to identify them. Those verbs that can also have Unexpressed Object Alternation (Section 1.2 in Levin (1993)) were excluded from the list though[19], because they can behave as both unaccusative and unergative in their intransitive use. The employed list of patientive ambitransitive verbs can be found in Appendix B.

Although the abovementioned problem was diminished, treating the intransitive version of patientive ambitransitives as passives brings up another, yet minor, problem. Some verbs, for example most verbs listed in Section 1.1.2.2 and Section 1.1.2.3 in Levin (1993), are verbs mostly encountered in their intransitive use, implying that they describe an internally controlled action. On the other hand, they can also be understood in a transitive way, involving the same action, but caused externally. Treating those verbs in their intransitive use as passives can sometimes result in extracting strange paraphrases, especially if their subject is an animate entity. In the case of the sentence "The bell rang", for example, it is not a problem; extracting the (subject, paraphrase) pair (bell, be rung) is correct. But, in case of the sentence "The girl ran", extracting (girl, be run) is rather strange, because run, with animate subjects, is usually controlled internally rather than externally. Still, in order to identify the difference between the transitive and the intransitive function of these verbs, this special treatment of patientive ambitransitives is done, and rarely does this result in strange paraphrases, in particular since most noun compounds comprise inanimate objects.

4.1.1.8 Using synonyms, hypernyms, sister words and semantically similar words

Although the two corpora used seem large enough, no paraphrases for several noun compounds are found in them, especially when using the subject-paraphrase-object-triples version. This is because exact paraphrases in the corpora are searched for, which are rare. This is applicable to the other version due to a minimum number of co-occurrence frequencies

[19] Such verbs are the ones listed as Amuse-Type Psych-Verbs, and the verb *cut*.

employed to make mutual information stable. Moreover, there are several nouns in noun compounds that occur only very rarely or not at all in the corpora.

Kim and Baldwin (2007) demonstrate that using synonyms, hypernyms and sister words from WordNet can be useful in generating noun compounds that have the same semantic relation between their nouns. A word can be denominated a hypernym of another word if the sense of the latter is more specific than the former, thus turning the second word into a representation of a subclass of the former (Jurafsky and Martin, 2009). Two words are sister words if they share the same immediate hypernym (Fellbaum, 1998). Based on the work of Kim and Baldwin (2007), it is hypothesised here too, that noun compounds comprising semantically similar words are interpreted in the same way. Therefore, instead of just using the nouns in the noun compound when searching for paraphrases, the interpretation method is also tested by using their synonyms, hypernyms, sister words and words that are semantically similar, in order to improve the recall for noun compound interpretation. The synonyms, hypernyms and sister words for each noun are obtained from WordNet via JWNL, and the semantically similar words for a noun are retrieved through the two methods described in Chapter 5, which automatically extract sets of semantically similar words from corpora.

This means that, when drawing on sister words from WordNet and a (subject, paraphrase, object) triple (bottle, be made of, glass) is extracted with the first method, the paraphrase *be made of* will also be found for the noun compound *plastic bottle* besides *glass bottle*, since *plastic* and *glass* are sister words in WordNet. Although this helps to improve recall (since it extracts more paraphrases from the corpus), it lowers precision because unsuitable paraphrases are also extracted more frequently; the more words are used, the higher the recall, but the lower the precision.

4.1.1.9 Validation of results with Web search engines

When searching for paraphrases, even if with the subject-paraphrase-object-triples version, and only using the nouns in the noun compounds (not using synonyms, hypernyms, sister words or semantically similar words), some of the extracted paraphrases are not correct. This can be due to a rare usage of one of the nouns in the noun compounds, or due to parsing or tagging errors among others. And, as was noted before, applying the subject-paraphrase-and-paraphrase-object-pairs version, or using synonyms, hypernyms, sister words or semantically

similar words further reduces precision. Therefore, especially if one of these above mentioned versions is employed, the results should be validated by some means.

It was decided to use the Web through Web search engines to validate the paraphrases extracted from static corpora. It is assumed that if a paraphrase is suitable for a noun compound, at least some Web pages containing the noun compound paraphrased by that suitable paraphrase in some form should show up. This can easily be checked by issuing Web search engine queries in the form defined below and then verifying the number of returned pages. If no hit is returned for one of the (noun compound, paraphrase) pairs, it is discarded, and if at least some hits show up, its score is recalculated from its original score and the number of returned pages. Many types of search engine queries were tried; a description of these shall follow. First, very simple queries were tried; for a noun compound *n1 n2* and a paraphrase *p*, all the possible exact queries in the form

"n2Infl THAT p n1Infl"

were issued, where n1Infl and n2Infl are any of the inflections of n1 and n2, respectively, and THAT can be one of the following relative pronouns: that, which or who. The returned page hit counts of all these queries for a (noun compound, paraphrase) pair were then added together to form the Web validation score for that pair. Queries without these relative pronouns were also tested for. Here, the assumption that noun2 is usually the head of the noun compound also applies. The problem with this simple validation method, however, was that sometimes, Web search engines do not return a single result even for suitable paraphrases. In order to improve coverage, an extension of the simple method was undertaken by searching for other verb tenses of the paraphrase rather than simple present. Queries searching for the verb tenses present continuous, present perfect, simple past and simple future beside present simple were also employed. Thereafter, the page hits returned by them were added together to form the Web validation score (noun compound, paraphrase) pair. Then, another alternative version was tried out, namely queries with wildcards, similarly to Nakov and Hearst (2006) and Nakov (2007). The wildcard characters were placed between the paraphrase (p), and the first noun of the noun compound (n1Infl). Queries with up to 9 wildcards were issued. The wildcard character is * in both search engines, which can substitute any word, although in the case of Google sometimes it substitutes more than one word.

In order to reduce the number of queries, the Boolean operator OR was employed, both inside the exact queries and between them. Although, as stated before, Web search engines can be inconsistent with Boolean logic, when the number of the pages found is small (as with exact queries like the ones proposed here), using a small number of Boolean operators usually returns correct results. Thus, instead of issuing all the possible queries for a given (noun compound, paraphrase) pair that covers all inflections of both nouns and all possible substitutes for THAT, the following query is performed (in the case of the basic approach):

"noun2Sing that OR which OR who pSing noun1Sing OR noun1Plur" OR
"noun2Plur that OR which OR who pPlur noun1Sing OR noun1Plur"

where noun1Sing and noun2Sing mean the singular form of the nouns, noun1Plur and noun2Plur mean the plural form of the nouns, pSing means the third person singular form of the paraphrase and pPlur means the plural form of the paraphrase. The queries are very similar in case of the extended versions too. For example, with the version that uses several verb tenses, the only difference is that a query for each verb tense is issued where the paraphrases are in the appropriate tense, and the resulting page hits are added together to form the Web validation score of the (noun compound, paraphrase) pair. These multiple queries could also be concatenated by the OR operator. However, very long queries and many Boolean operators result in false results with both search engines, so these multiple queries are carried out separately instead.

After searching for a (noun compound, paraphrase) pair on the Web with one of the above described queries, the score of the (noun compound, paraphrase) pair is recalculated from its original score and its Web validation score[20]. This is done as follows:

$$score_{new} = \ln\left(score_{original} + 1\right) * \ln\left(WebValidationScore + 1\right)$$

where $score_{original}$ is the original score of the (noun compound, paraphrase) pair, and $WebValidationScore$ is its Web validation score. The logarithms of the scores are used in

[20] Actually, the method defined in GeneralParaphrasingMethod.java returns the two scores separately for each (noun compound, paraphrase) pair, rather than just returning the combined score for each. Therefore, these scores should be combined with a different program (for example, this is possible with the program defined in ListMerge.java, when one of the input lists is empty). This is done so that the results of two different versions of the general paraphrasing method can be combined, as is done to obtain the best results (see Section 6.2).

order to promote those paraphrases that are frequent with the noun compound in both the static corpora and on the Web, and plus one is added to the original score and the page hit count, so that their logarithm is a valid number even if they are equal to 0.

The queries were issued to search for Web pages only in English. Furthermore, in order to improve the page hit estimates, the largest possible number of returned pages was requested for each query[21]. All the inflections of nouns and verbs generated for this validation method were obtained with the Morphg morphological generator tool.

4.1.1.10 Improving the performance of the programs

The problem with applying the two mentioned corpora is that they are huge, it takes a long time to read them through, and only a small part of the information is utilised. Therefore, in order to avoid having to read through the whole corpus each time, two programs were created that are able to read through the given corpus (the British National Corpus or the Web 1T 5-gram Corpus) and extract all the (subject, paraphrase, object) triples, (subject, paraphrase) and (paraphrase, object) pairs, together with all the needed frequencies to compute mutual information. In the end, they write all the information collected into a simple text file. Furthermore, they also extract all the information needed for the methods described in Chapter 5 in order to improve their performance, too. This results in a file the size of 410 MB for the British National Corpus, and a file the size of 800 MB for the Web 1T 5-gram Corpus, minor sizes compared to the approximately 5 GB size of the parsed British National Corpus, and around 100 GB size of tagged 4-grams and 5-grams of the Web 1T 5-gram Corpus (as described in Section 4.1.1.3, only the 4-grams and 5-grams were used). Further and more importantly, running the paraphrasing program on these files takes only a fragment of the time needed to run the program on the corpora, especially in the case of the Web 1T 5-gram Corpus. The main function of these programs can be found in ExtractInformationFromBNC.java and ExtractInformationFromWeb1T5gramCorpus.java, respectively.

[21] At the time of writing this was 1000 for both search engines. Although Nakov (2007) reported that using page hit estimates impacts accuracy negatively after having requested 1000 pages, in the noun compound bracketing task, with this noun compound interpretation task it was found that requesting the maximum number of pages improves accuracy.

4.1.2 A method to solve the SemEval-2 Task #9

The method described in the previous section to validate the results returned by the general paraphrasing method issues Web search engine queries for a list of possible paraphrases for each input noun compound and returns a Web validation score for each (noun compound, paraphrase) pair. Based on that, a method for the SemEval-2 Task #9 can be very easily developed, since the task consists in ranking a predefined list of paraphrases for noun compounds. The method proposed here returns a ranked list of the input paraphrases for each noun compound based on the returned Web validation score for that paraphrase. The same type of Web search engine queries can be attempted with this method too; a comprehensive illustration of the types of these queries can be found in Section 4.1.1.9. This method requires an input file containing the noun compounds with a list of possible paraphrases for each in the format of a gold standard file defined for the SemEval-2 Task #9. It returns a file containing the ranked list of paraphrases for each noun compound in the output format defined for the task, so that the results can then be evaluated by means of the scorer provided.

4.2 Interpreting noun compounds with more than two nouns

To interpret noun compounds that consist of more than two nouns, first, their syntactic structure needs to be determined. For example, to interpret *plastic water bottle*, it is required to know whether the noun *plastic* connects to the noun compound *water bottle* (meaning a water bottle made of plastic) or whether the noun *bottle* connects to the noun compound *plastic water* (meaning a bottle for plastic water); here, obviously the first is correct. This syntactic structure of a noun compound can be represented by brackets (hence the name of this task is noun compound bracketing); three-noun noun compounds can either be left bracketed ([[noun1 noun2] noun3]) or right bracketed ([noun1 [noun2 noun3]]). Regarding *plastic water bottle,* this structure can be represented as [plastic [water bottle]].

There have already been numerous methods proposed for noun compound bracketing. Lauer (1995), for example, proposed an unsupervised model that uses conceptual associations between the words of the noun compounds based on n-gram statistics from Grolier's encyclopedia[22] to determine left or right bracketing. Girju et al. (2005), on the other hand,

[22] http://go.grolier.com

presented a supervised model containing 15 features for three-noun noun compound bracketing (5 feature for each noun), based on the WordNet sense of the nouns. Moreover, Nakov (2007) use a lightly supervised approach based on surface features and paraphrases that are extracted from the Web by Web search engine queries.

After the structure of these longer noun compounds is determined, their interpretation can be obtained by defining the relationship between their parts. For a noun compound of *n* nouns, there are *n-1* such relations to be determined, so its interpretation can be given with *n-1* paraphrases (or paraphrase distributions). For example, to interpret the noun compound [plastic [water bottle]], the relationship between *bottle* and *water*, and the relationship between *water bottle* and *plastic* needs to be determined; a possible interpretation for it is given by the paraphrases *be for* and *be made of*, where the first defines the relationship between *bottle* and *water*, and the second defines the relationship between *water bottle* and *plastic*.

As stated before, in almost all noun compounds, the second noun is the head and the first is the dependent, defining a property for the head, and the compound of the two nouns syntactically behaves as the head would. Thus, the head of the noun compound can substitute the whole noun compound to form a more general concept. Therefore, the relation between a noun compound and a noun is the same as the relation between the head of the noun compound and the noun, a relation which is actually the interpretation of the noun compound formed by the head of the original noun compound and the noun. Hence, the task of interpreting a noun compound with *n* nouns can be substituted with the task of interpreting *n-1* two-noun noun compounds, by replacing smaller noun compounds in the whole noun compound with their heads. For example, the interpretation of the noun compound [plastic [water bottle]] can be performed by analysing the noun compounds *water bottle* and *plastic bottle* separately, since the noun compound *water bottle* can be replaced with *bottle* to form a more general concept. Then, the two separate interpretations form the overall interpretation of the whole noun compound.

Because interpreting longer noun compounds can be substituted with the task of interpreting two-noun noun compounds separately, the general paraphrasing method described in Section 4.1.1 can also be employed to interpret noun compounds of more than two nouns, given their syntactic structure has been previously determined. Although identifying the syntactic

structure of noun compounds is beyond the scope of this dissertation, numerous methods are already solving this task (for example the methods mentioned above). Thus, for a previously bracketed noun compound [[noun1 noun2] noun3]], for example, the general paraphrasing method can be used to interpret the noun compounds *noun1 noun2* and *noun2 noun3* separately, which two interpretations together form the interpretation of the whole noun compound. This can be applied very similarly to noun compounds comprising any number of nouns, with any structure.

5 Methods for automatically creating sets of semantically similar words

As explained in Section 4.1.1.8, it is hypothesised, similarly to Kim and Baldwin (2007), that noun compounds comprising semantically similar words share the same semantic relations. Therefore, in order to improve coverage, instead of just using the inflections of nouns in the noun compounds to search for paraphrases, their synonyms, hypernyms, sister words and semantically similar words are also tested for. In this chapter, two methods are presented by which a set of semantically similar nouns for each input noun can be automatically created. The underlying idea assumes that nouns that behave similarly, i.e. occur in relation with similar words, are semantically similar. Therefore, in order to identify semantically similar nouns, the words that they are frequently in relation with should be identified. This can be achieved by automatically analyzing relations extracted from a large corpus, where one of the words in the relation is a noun.

Both methods presented here resort to (subject, paraphrase), (object, paraphrase) and (noun, non-clausal modifier) relations extracted from a corpus, and define (noun, noun) similarity based on these relations. The same two static corpora were selected for this task as well as for the general paraphrasing method, namely the British National Corpus and the Web 1T 5-gram Corpus. However, as described in Section 4.1.1.3, tagging short n-grams results in too many errors, therefore no (noun, non-clausal modifier) relations, only (subject, paraphrase) and (object, paraphrase) relations were extracted from the latter corpus. The (subject, paraphrase) and (paraphrase, object) relations are extracted from the corpus in the same way, as described in Section 4.1.1.1, and the (noun, non-clausal modifier) relations are simply the (noun, non-clausal modifier) relations identified by the parser.

Both methods take a list of nouns as input, compare all of these nouns to all of the common nouns found in the corpus, and return a ranked list of nouns for all input nouns based on their similarity. However, only nouns with frequency of more than 5 are considered as possible similar nouns. The reason for this was three-fold; first, many of the words discarded this way are in fact not common nouns and were tagged that way because of a tagging error. Second, feature distribution is not really representative with such low frequencies. Third, as stated

before, these methods need a significant amount of computation, and by ignoring these words the amount of computation needed is proportional compared to if they were also used.

Section 5.1 describes a method that was originally proposed by Lin (1998) to define word similarity; this has been implemented with some changes here. Section 5.2 exposes a method that determines the similarity between nouns by means of numerical feature vectors. The main function of these methods can be found in ExtractSemanticallySimilarWordsWithLinMethod.java and ExtractSemanticallySimilarWordsWithNumericalFeatureVectors.java, respectively.

5.1 The method originally proposed by Lin (1998)

The method to measure word similarity as proposed by Lin (1998) was implemented for this dissertation with some changes. As stated above, (subject, paraphrase), (paraphrase, object) and (noun, non-clausal modifier) relations are used as the features of the nouns (with the Web 1T 5-gram Corpus, the latter type of relations are not used). Each noun is described by a set of these features, irrespective of the number of times a relation occurred. These feature sets are then compared with the following similarity measure:

$$sim(w_1, w_2) = \frac{2 * I\big(F(w_1) \cap F(w_2)\big)}{I\big(F(w_1)\big) + I\big(F(w_2)\big)}$$

where for any set S, $I(S)$ is the amount of information contained in it. In order to make computation easier, it is assumed that all the features are independent of each other (although this is not completely true). In this case the amount of information contained in a set can be calculated as:

$$I(S) = -\sum_{f \in S} \log P(f)$$

where $P(f)$ is the probability of feature f. Although the probability of a feature cannot be exactly computed, it can be estimated with the proportion of common nouns that have that feature among all the extracted common nouns. This similarity measure returns a value for each word pair between 0 and 1 (inclusive), 1 being when the two words have the same set of features, and 0 being when they do not have a common feature.

Some changes have been made to the originally proposed method, however. First, Lin proposed to use (subject, verb), and (verb, object) relations directly available from the dependency triples returned by the parser. Here, instead of verbs, paraphrases are adopted which can consist of a single verb and of a verb and preposition(s), as well. In addition, they can also contain verb particles. It is supposed that by testing for paraphrases instead of single verbs, the results should improve, since phrasal verbs and verbs with prepositions should characterize a noun better than single verbs. For example, take the sentence "The plane took off"; extracting the (subject, paraphrase) pair (plane, took off) is much more descriptive of the noun *plane* than the pair (plane, took). Moreover, extracting just the verb without the particle can bias the results negatively; it makes the noun *plane* more similar to nouns that also occur with the verb *take* with or without any particle, and not only to nouns that occur with *take off*. Prepositions and verb particles are extracted the in same way as detailed in Section 4.1.1.5. Second, as demonstrated in Section 4.1.1.6, passive paraphrases are different from other paraphrases since their grammatical subject is actually their underlying object. Therefore, they should be treated differently than active paraphrases; this is done exactly the same way as depicted in Section 4.1.1.6. Third, as was detailed in Section 4.1.1.7, patientive ambitransitive paraphrases in their intransitive use behave the same way as passive paraphrases do, namely their surface subject being their underlying object. Therefore, under this method as well, patientive ambitransitive paraphrases encountered in their intransitive use are treated as if they were passive. Fourth, only three types of relations are considered in the feature sets, namely (subject, paraphrase), (paraphrase, object) and (noun, non-clausal modifier) relations, since only these were deemed relevant for nouns. The (determiner, noun) relations were not taken into account, since these relations are not relevant when only comparing nouns. Besides, no other types of relations were considered, since they are believed to originate in parsing errors or being irrelevant to noun-noun comparison.

5.2 Automatically creating semantic categories using numerical feature vectors

The method described in the previous section, however, does not take the frequencies of relations in the corpus into account, which is also useful information. To make use of this information too, in this method each noun is described with a vector containing its dependency relations as features with weights. Noun similarity is based on the similarity of

these vectors. The weight for a feature is based on its frequency, and the two types of weights considered are described in Section 5.2.1. Similar to the method outlined in the previous section, paraphrases are used with possible prepositions and/or verb particles instead of simple verbs, and passive paraphrases and patientive ambitransitives in intransitive use are also treated the same way as in the other approach.

5.2.1 The measures used as weights in the feature vectors

Two measures are applied as weights in the feature vectors. Whereas the first is simply the frequency of that particular feature, the other is mutual information (a description of mutual information can be found in Section 4.1.1.4). When frequencies are concerned, nouns displaying similar frequency distribution in their features are considered similar. In the case of mutual information, those nouns are considered similar, for which the same features are most descriptive. As also mentioned in 4.1.1.4, mutual information is unstable with very small frequencies, therefore only relations with a frequency of greater than 5 were included in the feature vectors in this case.

5.2.2 Vector similarity measures

Three vector similarity measures were considered. The first was cosine similarity, which measures the similarity of two vectors by accounting for the cosine of the angle between them (Crook, 2010, Manning and Schütze, 2000). It can be calculated as:

$$\cos(\vec{x},\vec{y}) = \frac{\vec{x}\cdot\vec{y}}{|\vec{x}||\vec{y}|} = \frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} x_i^2}\sqrt{\sum_{i=1}^{n} y_i^2}}$$

It was decided to test for another vector similarity measure as well, namely the Dice coefficient (Manning and Schütze, 2000), as it also has been used by many researchers for comparing vectors. In its original version, it can only be used for Boolean vectors though, as:

$$\text{Dice}(\vec{x},\vec{y}) = \frac{2|\vec{x}\cap\vec{y}|}{|\vec{x}|+|\vec{y}|}$$

Therefore, it has to be generalized to numerical vectors in order to be able to apply it for this purpose. However, more than one way of generalizing has been proposed. Lin (1998) suggested the following calculation:

$$\mathrm{Dice}(\vec{x}, \vec{y}) = \frac{2 \sum_{i=1}^{n} x_i y_i}{\sum_{i=1}^{n} x_i^2 + \sum_{i=1}^{n} y_i^2}$$

On the other hand, Nakov (2007) proposed the following:

$$\mathrm{Dice}(\vec{x}, \vec{y}) = \frac{2 \sum_{i=1}^{n} \min(x_i, y_i)}{\sum_{i=1}^{n} x_i + \sum_{i=1}^{n} y_i}$$

As both generalizations were tested for here, the three similarity measures calculated were the cosine similarity as well as the two versions of the Dice coefficient.

6 Evaluation of the results

This chapter is dedicated to the evaluation of the two paraphrasing approaches proposed in this dissertation. First, the method, which was written for the SemEval-2 Task #9 was tested, in order to find out which type of Web searches provide the best results for paraphrasing. These results are presented in Section 6.1. Then, the general paraphrasing approach was assessed using the type of Web search that proved to be best in the case of the other method. Section 6.2 provides the results for this general paraphrasing method. As pointed out in Section 4.1.1.2. Web search engine results are unstable over time (even running the same query twice immediately one after the other can result in different results), therefore the results for both methods are variable in time.

6.1 Evaluation of the method for the SemEval-2 Task #9

All the different versions of queries described in Section 4.1.1.9 were tried in this method. First, the versions were tested on the first 100 noun compounds of the test data provided for the SemEval-2 Task #9[23], and just the best performing one was tested on the whole dataset. This can be attributed to the fact that some of them require a significant number of Web searches and that the number of queries that can be issued with commercial search engines in a given time period is limited (as described in Section 4.1.1.2). The evaluation was done by means of the scorer provided for the task, which compares the distribution of returned paraphrases for each noun compound against a gold standard provided for the task. Since the Kullback-Leibler divergence is only defined for positive values, all the 0 scores were replaced by 0.001. Furthermore, if for one of the noun compounds the score of all the (noun compound, paraphrase) pairs are the same, the rank of one of those pairs (this pair is randomly selected) is changed from 1 to the next value (if there are x (noun compound, paraphrase) pairs for the noun compound, then to x), so that the Spearman's rank correlation coefficient can be calculated (otherwise the calculations would require a division by 0). The results of some of the versions tried can be found in Table 3.

[23] https://docs.google.com/View?docid=dfvxd49s_35hkprbcpt

Method No.	Search Engine	Max. number of wildcards	Number of verb tenses	Relative pronouns	Average Spearman's correlation	Average Pearson's correlation	Average Kullback-Leibler divergence
1	Yahoo	0	1	no	0.2577	0.2191	6.6105
2	Yahoo	0	5	no	0.2628	0.2266	6.5531
3	Yahoo	0	1	yes	0.2260	0.2645	6.0842
4	Yahoo	7	1	no	0.2936	0.2006	6.5193
5	Google	0	1	no	0.2313	0.2685	5.1769
6	Google	0	5	no	0.2472	0.2828	5.0271
7	Google	0	1	yes	0.2183	0.2961	5.2899
8	Google	1	1	no	**0.3159**	**0.3140**	**4.2370**

Table 3: Results of some of the versions of the method for the SemEval-2 Task #9

If the maximum number of wildcards is x for a method, it means that queries with 0, 1, ... x number of wildcards were issued and that their results were added up. Queries without the relative pronouns *that*, *which* or *who* proved to return better results than the ones including them. Moreover, as can be observed from the results, testing for several verb tenses instead of just simple present affected the results positively. In the case of both search engines, inserting wildcards proved to be the best; optimal results were achieved with up to one wildcard for Google and with up to seven wildcards for Yahoo!.

As method No. 8 came off best on the first 100 noun compounds of the test data set (according to all three measures), that method was tested on the whole test data set. It achieved an average Spearman's rank correlation coefficient of 0.3387, an average Pearson's correlation coefficient of 0.3196 and an average Kullback-Leibler divergence of 4.1520 (the first being the official measure). Those 15 noun compounds of the test data set, on which this method performed best and worst (according to the Spearman's rank correlation coefficient), can be found in Table 4 and Table 5, respectively. Its separate performance on all the noun compound of the test data set can be found in Appendix C.

Noun compound	Spearman's rank correlation coefficient	Pearson's correlation coefficient	Kullback-Leibler divergence
colour printer	0.7050	0.6434	2.5173
warbler family	0.6762	0.4604	4.4521
photo album	0.6577	0.7501	1.0420
furniture company	0.6423	0.6736	0.8902
soup pot	0.6355	0.1015	2.5343
fabric house	0.6248	0.5740	3.1687
family business	0.5993	0.3232	2.2401
gun boat	0.5988	0.2997	3.9465
light bulb	0.5978	0.7003	1.5152
protein source	0.5825	0.7029	2.1611
metal body	0.5697	0.6111	2.1404
vehicle industry	0.5604	0.2262	2.7620
margin note	0.5552	0.5602	3.9474
carbon deposit	0.5543	0.8221	2.6708
paper tray	0.5538	0.6876	2.3574

Table 4: Those noun compounds of the test data set, on which the method for the SemEval-2 Task #9 performed best

Noun compound	Spearman's rank correlation coefficient	Pearson's correlation coefficient	Kullback-Leibler divergence
absorption hygrometers	0.0351	0.3631	2.8540
monkey pox	0.0170	-0.0259	6.8030
commonwealth status	0.0168	-0.0178	8.5196
summer comfort	0.0104	0.5012	6.9117
war secretary	0.0103	-0.0456	10.1940
dominion status	0.0076	0.0044	7.0167
eaves troughs	-0.0055	0.0104	4.9515
excavation skills	-0.0082	-0.0305	4.2094
altitude reconnaissance	-0.0096	-0.0513	6.0788
newspaper subscriptions	-0.0099	0.1932	4.8037
catalog illustrations	-0.0400	0.2526	5.1265
ballet genres	-0.0601	0.0126	5.0827
broadway youngster	-0.0693	-0.0637	6.9027
rococo spirit	-0.0778	-0.0569	4.0635
phonograph pickups	-0.1707	-0.2062	5.4311

Table 5: Those noun compounds of the test data set, on which the method for the SemEval-2 Task #9 performed worst

6.2 Evaluation of the general paraphrasing method

The evaluation of the general paraphrasing method was much trickier than the evaluation of the method for the SemEval-2 Task #9. Although it was also tested on the noun compounds in the test set of the SemEval-2 Task #9, the scorer provided to compare the results of the general paraphrasing method with the gold standard is not suitable for two main reasons. First, this general paraphrasing method does not use a predefined list of paraphrases for each noun compound to be ranked, therefore it also returns paraphrases that are not in the gold standard, and does not return some that are in it. Second, the paraphrase list for each noun compound in the provided gold standard is usually rather long, sometimes more than 100 paraphrases are proposed for a noun compound. That high number is absolutely not necessary for the interpretation of a noun compound; I believe that a handful of them are perfectly enough to define a noun compound's full meaning. The problem with these long lists of noun compounds is that the method proposed here often only returns a small number of possible paraphrases. Although these may be sufficient to define the full meaning of the noun compound, comparing these against the long lists of paraphrases in the gold standard is not meaningful. Therefore, English native speakers were recruited who were given the returned set of paraphrases for each noun compound without their scores and ranking, and were asked to score each. Because of the limited amount of available human resources, the different versions of the general paraphrasing algorithm were tested manually first. In the end, only the method considered to return the best results was evaluated by the judges. Furthermore, for the evaluation only the first 50 nouns of the SemEval-2 test data set were taken into account. As I believe that a handful of paraphrases are perfectly suitable to interpret a noun compound, only the best 3 returned paraphrases were evaluated for each noun compound. After all, 5 native speakers were employed and each was given the task of evaluating these 50 times 3 (noun compound, paraphrase) pairs. They were required to deliver a score between 1 and 5 (both inclusive) for each paraphrase, 1 meaning that it is completely unsuitable and 5 meaning that it is perfectly suitable.

So, before the evaluation, the best version of this method was determined by manually testing the results. The subject-paraphrase-object-triples version performed with a significantly higher precision, yet with lower recall. The higher recall for the other method is due to the fact that identifying a suitable (subject, paraphrase, object) triple is much more improbable than just finding suitable (subject, paraphrase) and (object, paraphrase) pairs. However, the

former provides a suitable paraphrase with a much higher probability since it recognises an actual paraphrase of the noun compound. Furthermore, many more paraphrases were returned when applying the Web 1T 5-gram Corpus than when using the British National Corpus; this is due to the fact that the former is many times larger than the second. Nevertheless, as stated before, even with these large corpora, there are no paraphrases found for many noun compounds in them. Therefore, the inclusion of synonyms, hypernyms, sister words and semantically similar words for the nouns in the noun compounds was proposed.

The synonyms, hypernyms or sister words for a noun can be obtained by WordNet, while semantically similar words for a noun can be gained through one of the methods proposed in Chapter 5. Both methods proposed in Chapter 5 proved to be successful in returning semantically similar words. They both delivered a better performance when tested on the Web 1T 5-gram Corpus. The method employing numerical feature vectors performed significantly better when using mutual information as weights for features instead of simply frequencies, as was expected. Applying the three different proposed vector similarities did not result in much change in the results, however; still, the cosine similarity was considered best. Comparing the results of the two methods resulted in a decision in favour of the method originally proposed by Lin (1998). The three most similar nouns for all the nouns in the first 50 noun compounds of the test data set returned by this approach can be found in Appendix D[24]. However, these methods to obtain semantically similar nouns for a noun have a significant disadvantage over applying WordNet to obtain synonyms, hypernyms or sister words: it is hard to define a proper limit, above which a noun is considered to be similar to another noun. If a limit seems reasonable for a given noun it usually includes too many nouns for another, and too few for yet another. The general paraphrasing method was tested using the best automatic extraction method to obtain semantically similar words, and using WordNet to obtain synonyms, hypernyms or sister words. After many experiments, the sister word obtained from WordNet proved to be the best. Although initially it seemed logical to only resort to the sister words of one of the nouns in a noun compound at a time, as Kim and Baldwin (2007) also suggest, in order to avoid over-generation, actually better results were obtained when using the sister words for both nouns at a time.

[24] The most similar noun for any noun is actually itself, with a similarity score of 1, therefore in Appendix D those 3 most similar nouns are listed for each noun, which are different from that noun in focus.

Although including sister words improves recall considerably, it lowers precision significantly. Besides, when only using the nouns in the noun compounds (not using synonyms, hypernyms, sister words or semantically similar words), some of the extracted paraphrases are not correct. Therefore Web search engine queries are performed to validate the results; the new score for each paraphrase is calculated from its original score and its Web validation score, as described more comprehensively in Section 4.1.1.9. Probably the Web validation method is the reason why including sister words for both nouns returned better results; using sister words for both words results in more results (therefore higher recall). Although the precision is lower, most of the incorrect hits will be filtered out by the Web validation method anyway.

The final version of the general paraphrasing method, which returned the best overall results, is the subject-paraphrase-object-triples version, using the Web 1T 5-gram corpus. Obviously, the best precision is obtained when no substitute words are included, but then the recall is low. On the other hand, when many substitute words are used, the recall is high and the precision is low. As neither of these cases is optimal, a combination of the two methods is proposed. First, both the method with no substitute words and the method containing sister words is run; this results in two ranked list of paraphrases for each noun compound (although for many noun compounds only the version using the sister words returns paraphrases). These need to be somehow compiled into one common list. Since no substitute words involved results in a much higher precision, paraphrases returned by that method should be more likely to rank higher. Nonetheless, the other method returns significantly higher scores, since some nouns have more than 200 sister words, which can result in more than 40000 possible noun-noun combinations for a noun compound. Therefore, the scores returned by the method using the sister words need to be rescaled; as is done by the measure:

$$score_{new} = \frac{score_{original} * lowest\_of\_nosubst}{highest\_of\_withsisterwords}$$

where $score_{original}$ is the original score of the (noun compound, paraphrase) pair, $lowest\_of\_nosubst$ is the score of the lowest scoring paraphrase for the given noun compound returned by the version not testing for substitute words, and $highest\_of\_withsisterwords$ is the score of the highest scoring paraphrase for the given noun compound returned by the version applying sister words. Obviously, this formula is only

valid for those cases where both methods returned at least one paraphrase for the given noun compound. Furthermore, in case a paraphrase occurs in both lists, it is included in the common list with the score it achieved when the method with no substitute words was applied. With this rescaling, the best paraphrase for a noun compound returned by the method using sister words has the same score as the worst paraphrase for that noun compound returned by the method not using any substitute words. The ratio between the scores of the paraphrases returned by the same method remains the same. The two lists are combined before the Web validation. The main function of the program that does this rescaling and compiles a single list from the two paraphrase lists can be found in ListMerge.java. This rescaling proved to be useful, since it promotes paraphrases that are returned by the method with no substitute words. However, since a further validation with Web search engine queries is performed, it is possible that paraphrases returned by the other method rank higher, if they are highly suitable according to the Web validation. The Method No. 8 described in the previous section is used for the Web validation, since it returned the best results for the SemEval-2 Task #9. This validation is detailed in Section 4.1.1.9. The results of this method were evaluated then by human judges. For one of the noun compounds in the test set, namely *altitude reconnaissance*, no paraphrases were returned at all. This can be attributed to the fact that this noun compound is very rare, and even the second noun by itself occurs very infrequently. Therefore the human judges were actually requested to score 49 times 3 (noun compound, paraphrase) pairs. Subsequent calculations were realised as if 3 paraphrases had been returned for *altitude reconnaissance* too, all of which were given a score of 1 by each judge.

Before the human judges' evaluation can be used, a certain agreement between the individual judges needs to be corroborated. In the case of significant disagreement, neither is data provided by them reliable nor can conclusions be deduced from it. The reliability of the data was checked using Krippendorff's alpha measure, which is a standard reliability measure proposed by Krippendorff (2004). Krippendorff's alpha was calculated using a macro written for SPSS[25], by Hayes and Krippendorff (2007). The alpha returned was 0.435 for the evaluation provided by the 5 judges, which means that there was significant disagreement between them. This could result from the fact that many of the noun compounds given in the test data set are hard to interpret even for humans; for example, 2 of the judges were not able

[25] http://www.spss.com/

to evaluate the paraphrases for *activity spectrum*. Furthermore, sometimes the meaning of noun compounds can be ambiguous, which can lead to disagreement. Those 39 (noun compound, paraphrase) pairs with a standard deviation of at least 1.5 were discarded. Then the alpha measure became 0.696, which was considered acceptable for this task.

The evaluation was performed in two different ways. First, by calculating the average score given for those paraphrases ranked best by the method described here; they obtained a 3.1842 average on the scale from 1 to 5. The average score of the second and third ranked paraphrases were calculates similarly, resulting in 2.7687 and 2.5583, respectively. Then, the similarity of the paraphrase distribution returned for each noun compound by this method and the distributions given by the human judges were calculated, similarly as done with the method for the SemEval-2 Task #9. To do this, the evaluation returned by the judges was converted into the gold standard format required for the SemEval-2 Task #9 scorer, which provided the means for the comparison. This method achieved an average Spearman's rank correlation coefficient of 0.3108, an average Pearson's correlation coefficient of 0.2738, and an average Kullback-Leibler divergence of 0.1589[26]. Those 10 noun compounds of the test data set, for which the judges' average score of all the returned (and not omitted) paraphrases are the best and the worst, can be found in Table 6 and Table 7, respectively. The best 3 paraphrases returned for the 50 test noun compounds, together with the returned score, as well as the average and divergence of the scores given by the judges can be found in Appendix E.

[26] This calculation was performed on those noun compounds, for which at least 2 paraphrases with different score remained after omitting the (noun compound, paraphrase) pairs with a standard deviation of at least 1.5. On the other noun compounds the employed measures cannot be calculated.

Noun Compound	Average score given by the judges
broadway youngster	4.7500
cell membrane	4.6000
cattle population	4.4000
arts museum	4.3333
business sector	4.2000
arts colleges	4.0000
backwoods protagonist	3.8750
antibiotic regimen	3.8667
census population	3.8667
business applications	3.7000

Table 6: Those noun compounds of the test data set, on which the general paraphrasing method performed best (considering the judges' average score of all the returned (and not omitted) paraphrases)

Noun Compound	Average score given by the judges
championship bout	2.0000
buddhist philosophy	1.8000
cell block	1.7500
banana industry	1.7333
ancestor spirits	1.6000
anode loss	1.5000
bird droppings	1.2667
bow scrape	1.2500
activity spectrum	1.0000
altitude reconnaissance	1.0000

Table 7: Those noun compounds of the test data set, on which the general paraphrasing method performed worst (considering the judges' average score of all the returned (and not omitted) paraphrases)

7 Conclusion and future work

The interpretation of noun compounds is required for many NLP tasks, including machine translation, question answering, information retrieval and information extraction. The aim of this dissertation has been to develop a method that interprets noun compounds using paraphrases. Utilising paraphrasing methods for noun compound interpretation has already proven to be useful for many tasks, including machine translation (Nakov, 2008) and solving relational similarity problems (Nakov and Hearst, 2008) among others. Two such methods have been developed; one for general noun compound interpretation and one for the SemEval-2 Task #9.

The SemEval-2 Task #9 requires programs to return a ranked list of a given sample of suitable paraphrases for noun compounds according to their aptness. A method for this task was generated from the Web validation part of the general paraphrasing approach. This method was evaluated on the test data set of the SemEval-2 Task #9; the distribution of returned paraphrases for each noun compound was compared to the gold standard provided by the organizers with the scorer that was made available. It achieved an average Spearman's rank correlation coefficient of 0.3387, an average Pearson's correlation coefficient of 0.3196 and an average Kullback-Leibler divergence of 4.1520 (the first being the official measure). Although these scores are significantly lower than the scores of the best methods presented for this task (the best achieved an average Spearman's rank correlation coefficient of 0.45, and an average Pearson's correlation coefficient of 0.411 (Wubben, 2010); its average Kullback-Leibler divergence was not published), the results are promising, especially given the simplicity of the method (it is much simpler than most methods submitted for the SemEval-2 Task #9). In addition, it performed better than two of the submitted programs. Moreover, extending the initiated queries could lead to improvements in the future. For example, it could test for all the possible verb tenses, or employ the synonyms, hypernyms, sister words or semantically similar words of the nouns as well, similar to how it is done with the general paraphrasing method. Besides, the different extensions of queries could also be combined; for example searching for multiple verb tenses with wildcard characters or synonyms with wildcard characters, respectively.

It should be noted that all the methods proposed for the SemEval-2 Task #9, including the method discussed here, have the weakness of requiring a set of suitable paraphrases for each noun compound as input to be able to produce a ranked list of paraphrases. This limits their suitability for real-world applications, since in most applications requiring noun compound interpretation, no such input is available. Therefore, these methods need to be generalized in order to be implemented in most other applications. This method could be generalised easily. With all types of the Web queries proposed, instead of issuing exact queries with all the paraphrases for the noun compounds provided, queries with wildcard character(s) in the place of the verb could be performed. Then, there remains nothing else than to analyse the returned snippets. This generalization is very similar to the noun compound interpretation method described by Nakov and Hearst (2006) and Nakov (2007). I believe that this method, especially after generalization, could contribute to the development of future methods; it could be integrated into them to improve their results. Furthermore, these results indicate that Web search engine statistics can be useful for noun compound interpretation.

The general paraphrasing method uses large corpora to search for possible paraphrases for noun compounds, and then constructs a ranked list of possible paraphrases for each noun compound based on different scoring measures. This method was tested on the first 50 noun compounds of the test data set provided for the SemEval-2 Task #9, and the resulting first three paraphrases for each noun compound were evaluated by human judges. The average scores received by the paraphrases that were ranked first, second and third by the method proposed here are 3.1842, 2.7687 and 2.5583, respectively, on a scale of 1 to 5. This shows that the returned paraphrases are considered moderately suitable on average. Further, when comparing the paraphrase distribution for each noun compound returned by this method with the distributions given by the judges, it achieved an average Spearman's rank correlation coefficient of 0.3108, an average Pearson's correlation coefficient of 0.2738 and an average Kullback-Leibler divergence of 0.1589. Although the two correlation coefficients are not really high, from the beginning it was assumed that a general paraphrasing method would perform worse than a method written for the SemEval-2 Task #9, since it is much harder to automatically identify suitable paraphrases for a noun compound and return them ordered in terms of aptness than simply ranking a given list of paraphrases for a noun compound. Still, the Kullback-Leibler divergence obtained here is much smaller than with the other method. Given the difficulty of the task, I believe that these are promising results, especially when considering a significant disagreement about the suitability of paraphrases for noun

compounds even among native speakers. Furthermore, as related in Section 4.1.1.3, due to a lack of time, the Web 1T 5-gram Corpus was tagged and not parsed (it would have taken approximately half a year in CPU time), and the grammatical relations inside the n-grams were deduced based on part-of-speech patterns. However, inferring the grammatical relations from part-of-speech patterns embodies a much higher error rate than when the relations are obtained with a parser. Therefore it is suggested that in the future, the Web 1T 5-gram Corpus also be parsed and the general paraphrasing method be tested on that as well. This should improve the results significantly.

After all, the presented evidence allows me to believe that the methods presented here seem suitable to be employed in real-world applications. In any case, they are a valuable contribution to the field of Natural Language Processing.

Appendix A – List of Prepositions

of	2895148	along	18832
to	2445692	above	18667
in	1836647	up	18232
for	830314	down	15005
on	683603	near	13795
with	624486	off	12936
as	549222	outside	12441
at	489915	below	11980
by	486235	throughout	11608
from	401298	beyond	10528
about	166376	except	9619
into	147964	inside	7567
than	136592	onto	5499
like	112057	beneath	4321
over	106547	unlike	4314
after	106020	beside	4298
between	86204	via	4111
through	76334	till	3183
before	73721	alongside	2455
out	58127	amongst	2406
under	57300	past	2373
per	51502	besides	2303
against	51326	toward	1098
since	45859	amid	1012
within	43526	aboard	619
without	42188	underneath	570
during	41330	minus	397
until	37672	notwithstanding	387
around	34010	plus	319
towards	25839	save	210
across	22128	atop	118
upon	21530	amidst	104
among	21267	astride	103
behind	19626		

Appendix B – List of patientive ambitransitive verbs

abate
accelerate
acetify
acidify
advance
age
agglomerate
air
alkalify
alter
ameliorate
americanize
asphyxiate
atrophy
attenuate
awake
awaken
balance
bang
beam
beep
belch
bend
bivouac
blacken
blare
blast
bleed
blink
blunt
blur
board
bounce
break
brighten
broaden
brown
bubble
burn
burp
burst
buzz
calcify
canter
capsize
caramelize
carbonify
carbonize
change
char
cheapen
chill
chip
choke
clack
clang
clash
clatter
clean
clear
click
clog
close
coagulate
coarsen
coil
collapse
collect
compress
condense
contract
cool
corrode
crack
crash
crease
crimson
crinkle
crisp
crumble
crumple
crush
crystallize
dampen
dangle
darken
de-escalate
decelerate
decentralize
decompose
decrease
deepen
deflate
defrost
degenerate
degrade
dehumidify
demagnetize
democratize
depressurize
desiccate
destabilize
deteriorate
detonate
dim
diminish
dirty
disintegrate
dissipate
dissolve
distend
divide
double
drain
dribble
drift
drip
drive
drool
drop
drown
dry
dull
ease
emanate
empty
emulsify
energize
enlarge
equalize
evaporate
even
expand
explode
exude
fade
fatten
federate
fill
firm
flash
flatten
float
flood
fly
fold
fossilize
fracture
fray
freeze
freshen
frost
fructify
fuse
gallop
gasify
gelatinize
gladden
glide
glutenize
granulate
gray
green
grow
gush
halt
hang
harden
harmonize
hasten
heal
heat
heighten
hoot
humidify
hush
hybridize
ignite
improve
increase
incubate
inflate
intensify
iodize
ionize
jangle
jingle
jump
kindle
leak
lean
leap
lengthen
lessen
level
levitate
light
lighten
lignify
liquefy

lodge
loop
loose
loosen
macerate
magnetize
magnify
march
mature
mellow
melt
moisten
move
muddy
multiply
narrow
neaten
neutralize
nitrify
ooze
open
operate
ossify
overturn
oxidize
pale
perch
petrify
polarize
pop
pound
pour
proliferate
propagate
puff
pulverize
purify
purple
putrefy
quadruple
quicken
quiet
quieten
race
radiate
redden
regularize
rekindle
reopen
reproduce
rest
revolve
ring
rip
ripen
roll
rotate
roughen
round
rumple
run
rupture
rustle
scorch
sear
seep
settle
sharpen
shatter
shed
shelter
shine
short
short-circuit
shorten
shrink
shrivel
shut
sicken
silicify
silver
singe
sink
sit
slack
slacken
slide
slim
slow
smarten
smash
smooth
snap
soak
sober
soften
solidify
sour
spew
spin
splay
splinter
split
spout
sprout
spurt
squeak
squeal
squirt
stabilize
stand
steady
steam
steep
steepen
stiffen
stifle
straighten
stratify
stream
strengthen
stretch
submerge
subside
suffocate
sweat
sweeten
swim
swing
tame
tan
taper
tauten
tear
tense
thaw
thicken
thin
tighten
tilt
tinkle
tire
topple
toughen
triple
trot
turn
twang
twirl
twist
ulcerate
unfold
unionize
vaporize
vary
vibrate
vitrify
volatilize
waken
walk
warm
weaken
westernize
whirl
whiten
widen
wind
worsen
wrap
wrinkle
yellow

Appendix C – Results of the method for the SemEval-2 Task #9

Noun compound	Spearman's rank correlation coefficient	Pearson's correlation coefficient	Kullback-Leibler divergence
absorption hygrometers	0.0351	0.3631	2.8540
activity spectrum	0.3225	0.2413	5.3023
afternoon rain	0.3014	0.4808	4.1405
air current	0.2830	0.0243	3.0473
air pocket	0.3525	0.0923	4.5210
altitude reconnaissance	-0.0096	-0.0513	6.0788
anatomy professor	0.3794	0.7871	3.2733
ancestor spirits	0.2141	0.1423	7.6719
anode loss	0.3965	0.2594	5.8170
antelope species	0.3568	0.4778	4.7729
antibiotic regimen	0.5295	0.5311	5.5873
apartment dwellers	0.2579	0.6377	3.1506
application areas	0.4675	0.4917	1.6311
arab world	0.4164	0.0835	5.3588
area basis	0.2630	0.2517	3.2206
arts colleges	0.1874	0.1767	2.6013
arts museum	0.4223	0.4266	3.0470
automobile factory	0.4296	0.7759	2.7935
baccalaureate curriculum	0.1598	0.1288	8.0222
backwoods protagonist	0.1745	0.0365	4.1530
ballet genres	-0.0601	0.0126	5.0827
banana industry	0.2677	0.0858	6.7506
band concert	0.3581	0.2266	4.2111
bathing suit	0.1447	0.0916	7.3140
battery technology	0.4480	0.2610	3.6966
bile duct	0.3307	0.3404	4.1319
bird droppings	0.1430	-0.0165	7.5374
bow scrape	0.1637	0.0942	4.9282
broadway youngster	-0.0693	-0.0637	6.9027
buddhist philosophy	0.2961	0.0699	6.1899
building site	0.5071	0.6519	2.2238
business applications	0.3249	0.2651	1.8650
business economics	0.3619	0.3375	2.2550
business education	0.3216	0.1168	3.6730
business holdings	0.2272	0.1090	2.7102
business investment	0.3668	0.4123	1.8863
business sector	0.2844	0.2960	3.2914

cancer cells	0.4384	0.3713	1.7242
car odor	0.2692	0.3904	5.4695
carbon deposit	0.5543	0.8221	2.6708
carrier system	0.3028	0.3478	3.4558
catalog illustrations	-0.0400	0.2526	5.1265
cattle industry	0.2389	0.0693	4.9911
cattle population	0.5279	0.3858	4.3068
cattle town	0.4103	0.4606	4.9698
cell block	0.4063	0.6027	2.4021
cell membrane	0.4176	0.4843	2.6950
census population	0.2540	0.0618	5.1592
ceramics products	0.3358	0.0540	3.9485
championship bout	0.4618	0.3529	6.4705
chemistry laboratories	0.3680	0.3535	3.0710
chest pain	0.3886	0.4641	4.1406
child custody	0.5212	0.2672	3.8768
child welfare	0.4839	0.1806	3.3632
childhood sexuality	0.1974	0.1285	6.6084
choice species	0.2224	0.6173	4.6155
cirrus cloud	0.3427	0.5547	4.8605
city dwellers	0.4227	0.8589	1.8592
city legislature	0.1916	0.4071	6.0389
city population	0.3990	0.7175	2.3697
climate pattern	0.2454	0.1043	4.7968
coalition cabinet	0.2307	0.0430	7.4851
coalition government	0.4019	0.3928	3.8633
cold virus	0.5512	0.8210	3.0392
colour printer	0.7050	0.6434	2.5173
commonwealth status	0.0168	-0.0178	8.5196
communications industries	0.3065	0.1645	3.3833
communications satellite	0.3668	0.6102	3.7057
communications systems	0.2554	0.3574	2.1955
community education	0.2451	0.1331	3.4127
company car	0.4999	0.4925	1.7792
computation skills	0.2325	-0.0230	3.1452
computer catalog	0.2542	0.0151	6.6311
computer expert	0.3254	0.2854	2.6149
computer memory	0.3217	0.5085	3.7966
computer novices	0.0969	0.5766	4.1269
concert appearances	0.2264	0.0110	5.7627
concert hall	0.3350	0.5323	3.9461
concert music	0.3386	0.3458	2.9317
consonant systems	0.2970	0.0760	4.1435
construction industry	0.3691	0.0519	3.0895

construction materials	0.2967	0.7052	1.9644
construction quality	0.2079	0.1677	5.3792
convenience foods	0.4599	0.3300	2.8529
coronation portal	0.1654	0.0263	6.0594
country estate	0.3374	0.7047	4.1996
country music	0.4111	0.2969	3.0704
county town	0.3525	0.5902	2.6334
crime novelist	0.2768	0.1606	5.0888
crossroads village	0.4009	0.5914	6.1658
cumulus cloud	0.3126	0.7695	5.1867
cupboard doors	0.2847	0.0702	4.3926
customs union	0.3895	0.1583	6.7790
dairy barn	0.4111	0.1356	6.2713
dairy cattle	0.3072	0.0099	5.3395
day evaporation	0.3633	0.2431	4.2364
death penalty	0.4904	0.1251	4.9931
desert storm	0.2959	0.2441	3.6387
diesel engine	0.4192	0.5154	1.5983
disease agent	0.3116	0.5773	3.0733
disease organisms	0.2612	0.7683	2.4348
dog house	0.4236	0.3378	2.8007
dominion status	0.0076	0.0044	7.0167
drainage basins	0.4187	0.0676	2.3962
drainage patterns	0.2759	0.1006	3.1573
eaves troughs	-0.0055	0.0104	4.9515
education journals	0.2551	0.0338	3.3665
education movement	0.4305	0.4113	3.8036
election laws	0.1645	0.0361	7.7644
electron microscope	0.3260	0.7919	3.9184
emergency detention	0.2935	0.1041	7.7805
engine repair	0.1919	0.3402	4.3364
entrance stair	0.3303	0.5821	5.6645
equipment charge	0.2581	0.3887	6.2530
equivalence principle	0.3346	-0.0127	5.3463
estimation methods	0.3330	0.3803	2.3658
evening dance	0.1725	0.2759	5.0269
eviction notice	0.1970	0.0150	7.8954
exam anxiety	0.4342	0.1994	5.8474
excavation skills	-0.0082	-0.0305	4.2094
exit route	0.3742	0.1017	4.7003
expansion turbine	0.2753	0.1304	6.3440
extinction theory	0.3805	0.5824	4.3071
fabric house	0.6248	0.5740	3.1687
faculty members	0.5006	0.2705	2.2167

family business	0.5993	0.3232	2.2401
family connection	0.3384	0.2436	3.5361
family members	0.3681	0.5702	1.7878
family sagas	0.3358	0.1492	3.2676
family tradition	0.3008	0.3279	2.6219
fence post	0.4159	0.3709	4.4616
fertility pill	0.4168	0.4154	4.3810
fiber optics	0.3893	0.1433	3.7190
film music	0.5475	0.4320	2.8738
flood water	0.4619	0.0990	3.4371
flowing solution	0.2311	0.8185	8.0049
food industry	0.4506	-0.0132	3.8899
food products	0.5528	0.5071	1.3200
food shortages	0.2302	-0.0037	3.4709
frontier community	0.4574	0.1265	5.2950
frontier life	0.2745	0.0254	8.6968
frontier problems	0.2402	-0.0510	4.3881
furniture company	0.6423	0.6736	0.8902
fusion devices	0.2814	-0.0135	6.4041
game bus	0.5380	0.0900	5.6982
gestation period	0.2881	0.2574	5.7129
gingerbread man	0.4938	0.1013	4.8965
government agencies	0.3081	0.1687	2.8893
government buildings	0.4779	0.6966	1.7831
government officials	0.3914	0.1492	2.0400
government patronage	0.4812	0.4300	5.5330
government policy	0.3234	0.1667	2.1209
grinding abrasive	0.2497	0.2959	6.1900
group plan	0.3154	0.1386	2.7672
growth centre	0.3422	0.1492	5.0010
guild members	0.3747	0.2113	2.9964
gun boat	0.5988	0.2997	3.9465
hair follicles	0.3188	0.5407	2.8612
hardware business	0.5260	0.1755	2.4582
hardware technology	0.2861	0.1044	3.3200
health problems	0.4291	0.4574	1.7778
health standards	0.3478	0.2143	2.5365
heath family	0.3311	0.3598	6.1297
home town	0.3661	0.6358	2.9908
horror novel	0.4937	0.1007	4.5886
horror tale	0.4232	0.0360	5.3469
household refrigeration	0.0951	0.2585	8.1746
ice crystal	0.4508	0.1149	3.7860
ice water	0.2793	0.0797	2.4309

impeachment trial	0.2901	-0.0125	8.1745
incubation period	0.3671	0.2713	5.6234
industry revenues	0.4013	0.7158	3.2610
information sources	0.4014	0.5054	1.4971
insurance industry	0.3261	0.1877	2.7892
intelligence community	0.3257	0.5734	4.4956
january temperature	0.2473	0.7448	5.0010
jesuit origin	0.2044	0.1028	6.8288
jute products	0.4177	0.2639	3.3060
kerosene lamps	0.3835	0.6502	2.0730
kidney disease	0.4168	0.6325	2.8544
lab periods	0.1666	0.1332	5.2638
lab printer	0.3154	0.2701	5.5883
laboratory applications	0.1456	0.3563	3.2812
laboratory quantities	0.2762	0.4222	4.7396
language family	0.4348	0.5969	2.6833
language literature	0.4406	0.1748	3.5424
laser printer	0.2321	0.4293	4.0082
laser technology	0.4619	0.6957	2.7300
lava fountains	0.2495	0.0684	3.2692
law systems	0.3858	0.1321	2.0516
life imprisonment	0.2357	0.7254	5.2689
life savings	0.2983	0.4202	4.1254
life sciences	0.3736	0.4479	1.5965
life scientists	0.3189	0.3685	2.1574
lifetime achievement	0.2401	0.1140	7.0205
light bulb	0.5978	0.7003	1.5152
lightning stroke	0.3431	0.0362	6.6872
logic unit	0.4078	0.3683	4.8095
luxury hotels	0.4232	0.6306	1.7433
machinery operations	0.2058	0.0326	6.0216
majority leader	0.4717	0.6048	3.4480
management procedures	0.3099	0.1547	3.1820
margin note	0.5552	0.5602	3.9474
marriage customs	0.3112	0.1871	6.0749
meat ball	0.4304	0.2325	4.2089
meat products	0.4881	0.4383	2.1791
memory system	0.3502	0.3832	1.6486
metal airplane	0.5415	0.3763	3.4745
metal body	0.5697	0.6111	2.1404
metal separator	0.5325	0.2940	4.7877
metallurgy industry	0.2310	0.0602	7.2186
midnight train	0.3374	0.1644	6.2123
military assault	0.1442	0.1779	7.1586

minority businesses	0.3494	0.3764	4.7962
monastery buildings	0.3623	0.1207	4.9677
money policy	0.2662	-0.0154	3.6050
monkey pox	0.0170	-0.0259	6.8030
morning class	0.4752	0.3944	2.6439
morning exercise	0.3714	0.3145	3.7584
morning frost	0.4062	0.3603	4.9032
mosquito repellent	0.4763	0.7043	4.5846
moth ball	0.5033	0.7704	4.3131
mountain country	0.5284	0.6303	2.6165
mountain glaciers	0.3109	0.1962	3.8416
mountain valleys	0.3707	0.6032	3.4916
muscle group	0.3960	0.4533	2.6913
music theory	0.3394	0.1835	3.4515
musk deer	0.4999	0.3324	5.0293
mystery novels	0.3939	0.3108	2.9078
needle work	0.4027	0.1804	3.6529
newspaper subscriptions	-0.0099	0.1932	4.8037
north wind	0.3237	0.5272	2.7764
oak tree	0.3120	0.4016	3.1832
ocean basins	0.1769	0.0531	5.9202
ocean side	0.1945	0.5488	6.2803
office buildings	0.4993	0.5803	1.3667
oil pan	0.5493	0.3790	2.4060
opposition coalition	0.4552	0.2046	6.4279
ozone machine	0.3781	0.7618	4.9822
paper tray	0.5538	0.6876	2.3574
passover festival	0.2671	0.3850	6.9291
peach tree	0.4881	0.2740	2.4135
percentage composition	0.3513	0.0995	5.0951
perfume content	0.3040	0.1489	6.0681
period classifications	0.1756	0.4038	2.9298
petroleum industry	0.5099	0.3950	2.8631
petroleum products	0.4776	0.3142	2.0478
petroleum wealth	0.3276	0.7886	5.5281
phonograph pickups	-0.1707	-0.2062	5.4311
photo album	0.6577	0.7501	1.0420
photography movement	0.3118	0.2145	6.6524
piano performance	0.3556	0.1263	5.3112
pigment granules	0.3296	0.4529	3.2585
plasma membrane	0.2817	-0.0451	4.7134
plutonium theft	0.0945	0.0000	0.6207
policy options	0.3158	0.4084	2.9579
pollen basket	0.4377	0.2223	6.1334

population density	0.3646	0.2752	3.6605
population explosion	0.3219	0.3421	7.0786
pottery vessels	0.1711	0.2612	4.2452
poultry pests	0.1518	0.0780	3.3377
poultry products	0.3607	0.1204	3.4769
printer tray	0.3721	0.3987	5.1578
priority areas	0.5315	0.7294	1.9615
prison poems	0.1249	0.3083	3.9028
production facilities	0.3857	0.4466	2.5029
prohibition law	0.2595	0.1823	3.9791
property law	0.5308	0.4259	1.7407
protein source	0.5825	0.7029	2.1611
quadrant elevation	0.1825	0.5939	6.0216
quantum theory	0.3274	0.3855	3.1396
radar observation	0.4087	0.4165	4.7559
railway union	0.0943	-0.0534	7.3354
rain maker	0.1259	0.5743	6.3910
ratings systems	0.3704	0.3155	2.3500
reaction mixture	0.4140	0.1988	2.8600
reckoning season	0.2563	0.2380	6.6645
recreation area	0.4024	0.4591	3.0034
refrigeration storage	0.4159	0.4683	5.0340
relations agency	0.3444	0.1458	3.8241
river valleys	0.4189	0.4452	2.6648
road competitions	0.1096	0.1483	4.6432
rococo spirit	-0.0778	-0.0569	4.0635
room temperature	0.3118	0.3638	5.0071
rotation period	0.3536	0.0668	4.5815
safety standard	0.4020	0.4575	2.2589
sanskrit texts	0.4246	0.4009	4.0841
satellite data	0.3223	0.4659	3.1002
satellite system	0.4104	0.6246	2.2305
saturation point	0.4541	0.4147	4.1142
savanna areas	0.4860	0.6296	3.0755
sea animals	0.4513	0.4619	3.9961
sea lanes	0.1929	0.4561	4.1058
sea lions	0.3117	0.3044	3.3446
sea mammals	0.1932	0.8476	3.7626
sea monster	0.3406	0.4636	4.4892
sea urchins	0.2690	0.3269	4.0322
security pacts	0.1757	0.1081	3.4999
separation negatives	0.3550	-0.0415	3.7307
settlement patterns	0.2910	0.2836	3.3757
shorthand device	0.2687	0.2971	5.5539

silk worm	0.4941	0.8106	2.2103
snow ball	0.4503	0.0302	5.5508
soul music	0.3825	0.3736	2.4752
soup pot	0.6355	0.1015	2.5343
spring semester	0.2900	0.0865	5.3685
state fund	0.3594	0.4254	2.0365
steel frame	0.3707	0.5511	2.7451
storage batteries	0.3807	0.4076	2.6691
storage capacity	0.3411	0.2859	3.9044
storm cloud	0.4240	0.4555	3.6497
story idea	0.3437	0.3228	3.0712
street scenes	0.4035	0.2495	3.9100
strength properties	0.3839	0.2846	2.9516
string playing	0.4286	0.1394	4.8882
student discount	0.4128	0.4492	3.6414
student loan	0.4101	0.4319	2.1421
student price	0.4959	0.3829	3.0422
student protest	0.3390	-0.0091	5.6116
subsistence cultivation	0.3070	0.3729	6.3180
suffrage committee	0.0510	0.0725	5.6347
sugar cane	0.5486	0.4089	2.2856
summer comfort	0.0104	0.5012	6.9117
summer morning	0.3191	0.3379	4.7492
sun block	0.3107	0.3723	4.7369
sunday restrictions	0.1043	0.1698	5.2103
suspension system	0.3709	0.3371	3.5553
symphony orchestra	0.4912	0.7664	4.5641
tea room	0.5370	0.3575	3.6870
teaching professor	0.3831	0.2931	2.9098
telephone wire	0.5126	0.6820	3.6443
television era	0.2777	0.0497	7.8296
television newscaster	0.3639	0.6978	5.3740
television production	0.2012	0.0118	3.5240
television series	0.4122	0.2232	3.0411
television writer	0.4007	0.3952	3.3034
temple portico	0.1081	0.1848	6.2349
terrorist activities	0.3245	-0.0015	5.3474
theater history	0.1407	0.0519	6.4348
theater orchestra	0.3374	0.3684	4.9155
tobacco leaf	0.3470	0.2798	4.3174
town halls	0.4341	0.8403	3.1664
transmission system	0.3771	0.4221	2.1544
transportation equipment	0.4265	0.0303	4.8045
transportation system	0.3579	0.5843	2.1923

treatment systems	0.3361	0.4024	1.4363
treaty relationships	0.1609	0.2057	5.4405
trial lawyers	0.3398	0.1778	2.5462
trio sonata	0.3718	0.3124	7.0781
tuesday night	0.4839	0.3246	4.6499
tv antenna	0.4302	0.3276	3.9593
typewriter mechanisms	0.1756	0.2870	3.4020
union leader	0.4701	0.2404	2.2817
university cabinets	0.2466	0.4257	3.7561
university education	0.3586	0.3463	2.3581
university teachers	0.4129	0.4334	1.5214
valve click	0.4393	0.2455	5.1538
valve systems	0.4596	0.5314	2.1597
vase paintings	0.3377	0.0683	3.7249
vehicle industry	0.5604	0.2262	2.7620
vibration ratio	0.0531	-0.0445	7.8169
violin concerto	0.5164	0.2088	4.2728
war captives	0.0358	0.0921	5.0034
war crimes	0.3870	0.4869	2.3775
war god	0.3700	0.2209	4.1634
war secretary	0.0103	-0.0456	10.1940
warbler family	0.6762	0.4604	4.4521
warfare equipment	0.2642	0.6198	4.8843
warrior caste	0.5379	0.1785	6.3937
water vapour	0.5391	0.0502	3.2320
weapons policy	0.2141	0.0492	5.9932
weather report	0.4251	0.3986	3.3114
welfare agencies	0.3817	0.5504	2.6523
wilderness areas	0.2804	0.4346	3.1575
wind mill	0.3891	0.7597	4.4400
wing tip	0.1998	0.1798	4.7925
winter blooming	0.3724	0.5601	4.5283
winter semester	0.4604	0.2016	5.2793
wool scarf	0.5116	0.4979	3.0814
worker satisfaction	0.3182	0.1259	6.9511
world championships	0.2380	0.3305	3.7366
world community	0.3225	0.3006	2.9725
world economies	0.4320	0.4156	3.0778
world population	0.1974	0.5950	3.1551
world soul	0.2760	0.5901	3.1894
world war	0.4524	0.2101	2.6929
yesterday afternoon	0.1168	0.3870	9.2727
yesterday evening	0.1719	0.3491	8.4529

Appendix D – The most similar nouns returned by the method originally proposed by Lin (1998)

Noun	Similar noun	Score
absorption	accumulation	0.3490
absorption	airflow	0.3427
absorption	progression	0.3306
activity	project	0.3760
activity	development	0.3631
activity	program	0.3585
afternoon	evening	0.3714
afternoon	morning	0.3380
afternoon	tonight	0.2637
air	water	0.3023
air	gas	0.2776
air	food	0.2759
altitude	amplitude	0.2451
altitude	elevation	0.2438
altitude	thickness	0.2436
anatomy	physiology	0.3415
anatomy	epidemiology	0.3365
anatomy	workings	0.3160
ancestor	grandparent	0.2096
ancestor	tribe	0.1959
ancestor	cousin	0.1926
anode	cathode	0.3428
anode	inductor	0.3130
anode	thoroughfare	0.3110
antelope	femur	0.2918
antelope	merseyside	0.2917
antelope	fruitcake	0.2871
antibiotic	antidepressant	0.3119
antibiotic	chemotherapy	0.2906
antibiotic	estrogen	0.2850
apartment	cottage	0.3100
apartment	suite	0.2943
apartment	villa	0.2942
application	software	0.3894
application	program	0.3881
application	system	0.3761
arab	muslim	0.2850
arab	palestinian	0.2848
arab	russian	0.2707
area	country	0.3606
area	city	0.3582
area	region	0.3460
arts	science	0.3152
arts	marketing	0.2968
arts	engineering	0.2868
automobile	motorcycle	0.2908
automobile	bicycle	0.2670
automobile	workplace	0.2568
baccalaureate	doctoral	0.3736
baccalaureate	doctorate	0.3690
baccalaureate	bfa	0.3551
backwoods	rectory	0.3778
backwoods	dreamtime	0.3756
backwoods	banal	0.3613
ballet	shakespeare	0.2375
ballet	sociology	0.2335
ballet	cycling	0.2193
banana	cucumber	0.2330
banana	vibrator	0.2285
banana	vegetable	0.2211
band	artist	0.3131
band	song	0.3048
band	club	0.3043
basis	criterion	0.2577
basis	objective	0.2543
basis	behalf	0.2541
bathing	haircut	0.2701
bathing	urination	0.2628
bathing	whiteboard	0.2607
battery	cable	0.3080
battery	printer	0.3074
battery	drive	0.3017
bile	feces	0.2633
bile	arterial	0.2584
bile	analogue	0.2492
bird	animal	0.2744
bird	fish	0.2698
bird	dog	0.2503
block	component	0.2841

block	domain	0.2778
block	module	0.2778
bout	tribulation	0.2384
bout	nausea	0.2362
bout	ache	0.2338
bow	ribbon	0.2332
bow	tail	0.2226
bow	sink	0.2198
broadway	ne	0.2020
broadway	mtv	0.1993
broadway	smash	0.1927
buddhist	tibetan	0.2177
buddhist	sikh	0.2154
buddhist	hindu	0.2125
building	development	0.3191
building	house	0.3074
building	area	0.3066
business	company	0.3895
business	program	0.3591
business	school	0.3588
cancer	disease	0.3937
cancer	infection	0.3020
cancer	aids	0.2956
car	vehicle	0.3486
car	home	0.2933
car	book	0.2900
carbon	oxygen	0.2413
carbon	nitrogen	0.2191
carbon	mercury	0.2172
carrier	provider	0.3445
carrier	operator	0.3222
carrier	vendor	0.3167
catalog	brochure	0.3382
catalog	catalogue	0.3170
catalog	faq	0.3044
cattle	sheep	0.2311
cattle	livestock	0.2143
cattle	cows	0.1950
cell	plant	0.2925
cell	network	0.2842
cell	type	0.2812
census	evaluation	0.2689
census	questionnaire	0.2578
census	registry	0.2577
ceramics	calculus	0.2312
ceramics	genealogy	0.2303
ceramics	slr	0.2281
championship	tournament	0.2955
championship	celebration	0.2428
championship	olympics	0.2412
college	university	0.4274
college	education	0.3509
college	department	0.3493
concert	festival	0.2866
concert	ceremony	0.2782
concert	dance	0.2635
current	disturbance	0.2456
current	rainfall	0.2359
current	mutation	0.2298
curriculum	infrastructure	0.3193
curriculum	architecture	0.3168
curriculum	programme	0.3152
deposit	investment	0.2836
deposit	contribution	0.2787
deposit	distribution	0.2627
droppings	waypoint	0.3472
droppings	straggler	0.3327
droppings	galbraith	0.3274
duct	artery	0.2561
duct	backbone	0.2374
duct	urethra	0.2363
dweller	depart	0.2422
dweller	interact	0.2206
dweller	organize	0.2199
economics	psychology	0.3805
economics	biology	0.3445
economics	physics	0.3356
education	development	0.3985
education	management	0.3970
education	health	0.3936
factory	lab	0.2556
factory	printer	0.2401
factory	station	0.2399
genre	musical	0.2405
genre	cuisine	0.2396
genre	alphabet	0.2382
holding	recruitment	0.2599
holding	subsidiary	0.2581
holding	migration	0.2487
hygrometer	spirometer	0.4748
hygrometer	hydrometer	0.4183
hygrometer	lifejacket	0.3917

illustration	artwork	0.3231	reconnaissance	fieldwork	0.2872
illustration	photograph	0.3074	reconnaissance	consecration	0.2793
illustration	diagram	0.2996	reconnaissance	interpolation	0.2743
industry	community	0.3656	regimen	aspirin	0.3232
industry	organization	0.3656	regimen	chemotherapy	0.3212
industry	management	0.3465	regimen	supplementation	0.3152
investment	development	0.3314	scrape	swam	0.4483
investment	loan	0.3271	scrape	borne	0.4329
investment	employment	0.3265	scrape	swept	0.4096
loss	losses	0.3109	sector	industry	0.3148
loss	damage	0.3009	sector	organisation	0.3102
loss	increase	0.2901	sector	region	0.3090
membrane	tissue	0.2666	site	page	0.3914
membrane	enzyme	0.2477	site	website	0.3854
membrane	plasma	0.2442	site	program	0.3741
museum	centre	0.3140	species	population	0.2899
museum	institute	0.3051	species	plant	0.2829
museum	studio	0.2994	species	disease	0.2716
odor	smell	0.2677	spectrum	dimension	0.2814
odor	odour	0.2422	spectrum	characteristic	0.2702
odor	distortion	0.2361	spectrum	interaction	0.2581
philosophy	theory	0.3268	spirits	ghost	0.1967
philosophy	religion	0.2938	spirits	wolf	0.1954
philosophy	attitude	0.2921	spirits	creature	0.1881
pocket	bag	0.2437	suit	lawsuit	0.2924
pocket	sleeve	0.2367	suit	complaint	0.2331
pocket	leather	0.2365	suit	petition	0.2252
population	sector	0.2969	system	program	0.4239
population	species	0.2899	system	software	0.3887
population	region	0.2824	system	product	0.3821
product	item	0.3836	technology	software	0.4043
product	system	0.3821	technology	solution	0.3949
product	software	0.3742	technology	tool	0.3749
professor	instructor	0.3243	town	city	0.3441
professor	faculty	0.3108	town	village	0.3377
professor	colleague	0.3025	town	church	0.2945
protagonist	npc	0.2634	world	country	0.3461
protagonist	prius	0.2631	world	city	0.3375
protagonist	heroine	0.2596	world	state	0.3279
rain	snow	0.3003	youngster	salesperson	0.2055
rain	weather	0.2678	youngster	scout	0.2038
rain	storm	0.2479	youngster	classmate	0.2021

Appendix E – Results of the general paraphrasing method

Noun Compound	Paraphrase	Score (by method)	Average score (by judges)	Divergence
absorption hygrometers	be	14.0439	1.0000	0.0000
absorption hygrometers	be on	13.3909	1.2500	0.5000
absorption hygrometers	measure	10.6100	5.0000	0.0000
activity spectrum	be of	26.7131	3.6667	1.5275
activity spectrum	be for	16.2726	2.6667	2.0817
activity spectrum	be in	10.7864	1.0000	0.0000
afternoon rain	be in	56.1612	4.8000	0.4472
afternoon rain	be during	23.0597	4.4000	0.5477
afternoon rain	be for	21.6821	1.2000	0.4472
air current	be in	30.1152	3.4000	1.8166
air current	be of	28.5133	4.4000	1.3416
air current	be to	24.3634	1.0000	0.0000
air pocket	trap	18.4497	3.2000	1.0954
air pocket	be of	16.4381	4.4000	0.8944
air pocket	be with	13.3394	1.6000	0.8944
altitude reconnaissance		0.0000	1.0000	0.0000
altitude reconnaissance		0.0000	1.0000	0.0000
altitude reconnaissance		0.0000	1.0000	0.0000
anatomy professor	be of	15.5992	4.8000	0.4472
anatomy professor	be in	2.7777	2.6000	1.1402
anatomy professor	create	1.3618	1.2000	0.4472
ancestor spirits	be	15.5989	2.6000	1.6733
ancestor spirits	be in	2.6736	1.6000	0.5477
ancestor spirits	be produced by	0.0000	3.8000	1.6432
anode loss	be to	23.5312	3.4000	1.6733
anode loss	be at	22.2881	1.8000	1.3038
anode loss	be	21.9398	1.2000	0.4472
antelope species	be of	8.9923	4.8000	0.4472
antelope species	be	8.2826	1.8000	1.3038
antelope species	be in	3.9306	1.8000	0.8367
antibiotic regimen	include	19.1349	3.2000	1.3038
antibiotic regimen	consist of	8.8430	4.0000	0.7071
antibiotic regimen	be of	4.5163	4.4000	1.3416
apartment dwellers	live in	46.5472	5.0000	0.0000
apartment dwellers	go	39.0400	1.4000	0.5477
apartment dwellers	be	35.6423	3.0000	2.0000
application areas	be of	47.7512	3.4000	1.5166

application areas	be for	43.9145	4.2000	1.0954
application areas	be in	41.8724	1.8000	0.8367
arab world	be of	22.8370	4.8000	0.4472
arab world	be by	21.6905	1.6000	0.8944
arab world	be to	19.8338	1.2000	0.4472
area basis	be in	50.0286	3.2500	1.2583
area basis	be of	33.9275	2.5000	1.2910
area basis	be for	28.6121	3.2500	2.0616
arts colleges	be of	72.1971	3.8000	1.3038
arts colleges	be for	48.1469	4.2000	0.8367
arts colleges	be	40.7589	1.8000	1.7889
arts museum	be of	61.8303	3.8000	1.3038
arts museum	be devoted to	23.2533	5.0000	0.0000
arts museum	be for	23.1931	4.2000	0.4472
automobile factory	produce	20.0413	4.6000	0.5477
automobile factory	be in	18.0759	1.4000	0.5477
automobile factory	build	14.9290	3.4000	1.8166
baccalaureate curriculum	lead to	17.4878	3.4000	1.3416
baccalaureate curriculum	be for	7.7258	4.2000	1.3038
baccalaureate curriculum	be of	7.4517	1.8000	0.8367
backwoods protagonist	arrive at	0.0000	2.2500	1.8930
backwoods protagonist	be from	0.0000	5.0000	0.0000
backwoods protagonist	be in	0.0000	2.7500	1.2583
ballet genres	be	48.3079	1.6000	0.8944
ballet genres	include	25.9521	3.0000	1.5811
ballet genres	be based on	20.5786	4.0000	1.4142
banana industry	be	40.3781	1.8000	1.0954
banana industry	have	17.4962	2.0000	1.0000
banana industry	be by	13.7967	1.4000	0.5477
band concert	be by	38.1481	3.6000	1.5166
band concert	be of	37.0090	2.0000	1.0000
band concert	feature	35.0123	3.8000	1.6432
bathing suit	be	20.8833	1.2000	0.4472
bathing suit	be for	9.6829	4.8000	0.4472
bathing suit	be of	6.6970	1.2000	0.4472
battery technology	extend	36.5688	2.4000	1.3416
battery technology	enable	36.1910	2.0000	1.4142
battery technology	ensure	35.8417	1.6000	0.5477
bile duct	carry	35.2641	4.4000	0.5477
bile duct	join	26.6471	1.6000	0.8944
bile duct	transport	23.9108	4.6000	0.5477
bird droppings	be in	5.6527	1.4000	0.5477
bird droppings	be for	5.5403	1.0000	0.0000
bird droppings	be	5.0592	1.4000	0.5477
bow scrape	be in	6.9084	1.7500	1.5000
bow scrape	be	5.4948	1.2500	0.5000

bow scrape	have	4.8759	2.5000	1.9149
broadway youngster	be in	0.0000	4.7500	0.5000
broadway youngster	be into	0.0000	2.7500	1.7078
broadway youngster	sleep through	0.0000	2.0000	2.0000
buddhist philosophy	be of	5.3374	2.0000	1.2247
buddhist philosophy	be in	4.9098	1.6000	0.8944
buddhist philosophy	be	2.6448	2.8000	2.0494
building site	be	108.2961	1.4000	0.5477
building site	be of	55.0322	1.4000	0.8944
building site	be for	50.8700	4.6000	0.5477
business applications	be for	64.2917	4.8000	0.4472
business applications	be to	53.8009	2.2000	1.6432
business applications	be in	51.0856	2.6000	0.8944
business economics	be of	39.4818	3.6000	1.6733
business economics	be in	33.5548	2.8000	1.3038
business economics	be	32.1465	1.8000	0.8367
business education	be in	56.2773	4.6000	0.5477
business education	be	46.9496	1.4000	0.5477
business education	be for	36.8789	3.6000	0.5477
business holdings	be in	30.7344	4.6000	0.5477
business holdings	be	23.2833	1.6000	0.5477
business holdings	be of	17.8628	2.6000	1.1402
business investment	be in	61.4082	4.8000	0.4472
business investment	be for	44.1176	3.8000	1.3038
business investment	be	40.1514	1.4000	0.5477
business sector	be of	51.9284	4.2000	0.8367
business sector	be in	36.4256	2.6000	1.5166
business sector	be for	34.9956	3.6000	1.5166
cancer cells	be in	37.5728	1.8000	0.8367
cancer cells	be from	37.0726	2.2000	1.3038
cancer cells	lead to	33.6453	4.6000	0.5477
car odor	be in	12.8259	3.0000	1.5811
car odor	be from	7.1802	4.0000	0.7071
car odor	be to	4.5855	1.2000	0.4472
carbon deposit	be	17.1600	2.8000	1.3038
carbon deposit	be of	16.4623	4.6000	0.5477
carbon deposit	be in	11.7220	1.4000	0.8944
carrier system	be	35.5630	2.2500	0.9574
carrier system	be with	31.6172	2.7500	1.7078
carrier system	be for	31.0823	3.7500	0.9574
catalog illustrations	be in	14.2092	4.0000	1.7321
catalog illustrations	be from	10.3936	3.8000	0.4472
catalog illustrations	be	7.8472	1.4000	0.5477
cattle industry	be in	14.3383	2.6000	0.5477
cattle industry	be with	12.1840	2.4000	1.1402
cattle industry	be of	11.1582	3.4000	1.5166

cattle population	be	34.2653	2.6000	1.8166
cattle population	be in	23.8646	2.0000	1.7321
cattle population	be of	15.6345	4.4000	0.8944
cattle town	be of	8.0499	2.2000	1.7889
cattle town	have	6.0044	3.6000	1.6733
cattle town	be for	5.1439	3.2000	2.0494
cell block	be of	30.5239	3.5000	1.7321
cell block	be in	28.7987	1.7500	0.9574
cell block	call on	23.8065	2.0000	2.0000
cell membrane	be of	55.7722	3.2000	1.6432
cell membrane	be in	42.9233	3.0000	1.8708
cell membrane	surround	39.8416	4.6000	0.8944
census population	be in	41.1507	2.6000	0.8944
census population	be from	36.0210	4.0000	0.7071
census population	be estimated by	34.7135	5.0000	0.0000
ceramics products	include	22.5991	2.0000	1.4142
ceramics products	be of	13.2030	4.0000	1.4142
ceramics products	be	12.0003	3.8000	1.3038
championship bout	be	48.9412	2.0000	0.7071
championship bout	be for	45.0480	3.8000	1.7889
championship bout	be in	20.8150	3.0000	1.8708

References

BALDWIN, T. & TANAKA, T. 2004. Translation by machine of complex nominals: Getting it right. *Proceedings of the ACL 2004 Workshop on Multiword Expressions: Integrating Processing*. Barcelona, Spain.

BNC_CONSORTIUM 2001. The British National Corpus, version 2 (BNC World). Distributed by Oxford University Computing Services on behalf of the BNC Consortium.

BRANTS, T. & FRANZ, A. 2006. Web 1T 5-gram Version 1. Linguistic Data Consortium.

BRISCOE, T. 2006. An introduction to tag sequence grammars and the RASP system parser. *Computer Laboratory Technical Report*, 662.

BURZIO, L. 1986. *Italian syntax: A government-binding approach*, Springer.

BUTNARIU, C., KIM, S., NAKOV, P., SEAGHDHA, D. O., SZPAKOWICZ, S. & VEALE, T. 2009. Semeval-2010 Task 9: The interpretation of noun compounds using paraphrasing verbs and prepositions. *Proceedings of the 5th SIGLEX Workshop on Semantic Evaluation.* Uppsala, Sweden.

CHURCH, K., GALE, W., HANKS, P. & KINDLE, D. 1991. Using Statistics in Lexical Analysis. *In:* ZERNIK, U. (ed.) *Lexical acquisition: exploiting on-line resources to build a lexicon.* Hillsdale, NJ: Lawrence Erlbaum.

CHURCH, K. & HANKS, P. 1989. Word association norms, mutual information, and lexicography. *Computational Linguistics,* 16**,** 22-29.

CLARK, S. & CURRAN, J. 2004. Parsing the WSJ using CCG and log-linear models. *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics.* Barcelona, Spain.

CLARK, S. & CURRAN, J. 2007. Wide-coverage efficient statistical parsing with CCG and log-linear models. *Computational Linguistics,* 33**,** 493-552.

CROOK, N. 2010. Information retrieval lecture notes. University of Oxford.

CURRAN, J. & CLARK, S. 2003. Investigating GIS and smoothing for maximum entropy taggers. *Proceedings of the 11th Meeting of the European Chapter of the ACL.* Budapest, Hungary.

DIXON, R. & AIKHENVALD, A. 2000. Introduction. *In:* DIXON, R. & AIKHENVALD, A. (eds.) *Changing valency: Case studies in transitivity.* Cambridge: Cambridge University Press.

DOWNING, P. 1977. On the creation and use of English compound nouns. *Language,* 53**,** 810-842.

FELLBAUM, C. 1998. *WordNet: An electronic lexical database,* MIT press Cambridge, MA.

FRANCIS, W. 1979. *A manual of information to accompany A standard sample of present-day edited American English, for use with digital computers,* Dept. of Linguistics, Brown University.

GIRJU, R., GIUGLEA, A., OLTEANU, M., FORTU, O., BOLOHAN, O. & MOLDOVAN, D. 2004. Support vector machines applied to the classification of semantic relations in nominalized noun phrases. *Proceedings of the HLT-NAACL Workshop on Computational Lexical Semantics.* Boston, Massachusetts.

GIRJU, R., MOLDOVAN, D., TATU, M. & ANTOHE, D. 2005. On the semantics of noun compounds. *Computer speech & language,* 19**,** 479-496.

GIRJU, R., NAKOV, P., NASTASE, V., SZPAKOWICZ, S., TURNEY, P. & YURET, D. 2007. Semeval-2007 task 04: Classification of semantic relations between nominals. *Proceedings of the 4th International Workshop on Semantic Evaluations.* Prague, Czech Republic.

HAYES, A. & KRIPPENDORFF, K. 2007. Answering the call for a standard reliability measure for coding data. *Communication Methods and Measures,* 1**,** 77-89.

JURAFSKY, D. & MARTIN, J. 2009. *Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition,* Upper Saddle River, NJ, Pearson Education.

KILGARRIFF, A. 2007. Googleology is bad science. *Computational Linguistics,* 33**,** 147-151.

KIM, S. & BALDWIN, T. 2007. Interpreting noun compounds using bootstrapping and sense collocation. *Proceedings of the 10th Conference of the Pacific Association for Computational Linguistics.*

KRIPPENDORFF, K. 2004. *Content analysis: An introduction to its methodology*, Sage Publications, Inc.

LAPATA, M. & KELLER, F. 2005. Web-based models for natural language processing. *ACM Transactions on Speech and Language Processing (TSLP).*

LAPATA, M. & LASCARIDES, A. 2003. Detecting novel compounds: The role of distributional evidence. *Proceedings of the 10th conference on European chapter of the Association for Computational Linguistics.* Budapest, Hungary.

LAUER, M. 1995. Designing statistical language learners: Experiments on noun compounds. *Arxiv preprint cmp-lg/9609008*.

LEVI, J. 1978. *The syntax and semantics of complex nominals*, Academic Press.

LEVIN, B. 1993. *English verb classes and alternations: A preliminary investigation,* Chicago, Il.

LEVIN, B. & HOVAV, M. 1995. *Unaccusativity: At the syntax-lexical semantics interface,* The MIT Press.

LIN, D. 1998. An information-theoretic definition of similarity. *ICML '98: Proceedings of the Fifteenth International Conference on Machine Learning.* San Francisco, CA, USA.

MANNING, C. & SCHÜTZE, H. 2000. *Foundations of statistical natural language processing*, MIT Press.

MINNEN, G., CARROLL, J. & PEARCE, D. 2001. Applied morphological processing of English. *Natural Language Engineering,* 7**,** 207-223.

MITHUN, M. 2000. Valency-changing derivation in Central Alaskan Yup'ik. *In:* DIXON, R. & AIKHENVALD, A. (eds.) *Changing valency: case studies in transitivity.* Cambridge: Cambridge University Press.

MOLDOVAN, D., BADULESCU, A., TATU, M., ANTOHE, D. & GIRJU, R. 2004. Semantic Classification of Non-nominalized Noun Phrases. *Proceedings of the HLT-NAACL Workshop on Computational Lexical Semantics.* Boston, Massachusetts.

NAKOV, P. 2007. Using the Web as an Implicit Training Set: Application to Noun Compound Syntax and Semantics. *University of California at Berkeley, Berkeley, CA.*

NAKOV, P. 2008. Improved statistical machine translation using monolingual paraphrases. *Proceeding of the 2008 conference on ECAI 2008: 18th European Conference on Artificial Intelligence.*

NAKOV, P. & HEARST, M. 2005. A study of using search engine page hits as a proxy for n-gram frequencies. *In Proceedings of RANLP'2005*. Borovets, Bulgaria.

NAKOV, P. & HEARST, M. 2006. Using verbs to characterize noun-noun relations. *Artificial Intelligence: Methodology, Systems, and Applications***,** 233-244.

NAKOV, P. & HEARST, M. 2008. Solving relational similarity problems using the web as a corpus. *Proceedings of ACL'08: HLT*. Columbus, OH, USA.

NASTASE, V. & SZPAKOWICZ, S. Year. Exploring noun-modifier semantic relations. *In*, 2003. 285–301.

NULTY, P. & COSTELLO, F. 2010. UCD-PN: Selecting General Paraphrases Using Conditional Probability.

PERLMUTTER, D. 1978. Impersonal passives and the unaccusative hypothesis. *Proceedings of the Fourth Annual Meeting of the Berkeley Linguistics Society.*

PERLMUTTER, D. & POSTAL, P. 1984. The 1-advancement exclusiveness law. *Studies in relational grammar 2,* 2**,** 81-125.

ROSARIO, B. & HEARST, M. 2001. Classifying the semantic relations in noun compounds via a domain-specific lexical hierarchy. *In Proceedings of EMNLP*.

ROSEN, C. 1984. The Interface between Semantic Roles and Initial Grammatical Relations. *In:* PERLMUTTER, D. & ROSEN, C. (eds.) *Studies in relational grammar 2.*

SANTORINI, B. 1995. Part-of-speech tagging guidelines for the Penn Treebank Project. *University of Pennsylvania, 3rd Revision, 2nd Printing*.

SÉAGHDHA, D. 2008. Learning compound noun semantics. Technical Report.

VAN VALIN JR, R. 1990. Semantic parameters of split intransitivity. *Language,* 66**,** 221-260.

WARREN, B. 1978. Semantic patterns of noun-noun compounds.

WUBBEN, S. 2010. UvT: Memory-based pairwise ranking of paraphrasing verbs.