Automatic lexico-semantic acquisition for question answering

Plas, Marie Louise Elizabeth van der

Important note: You are advised to consult the publisher's version (publisher's PDF) if you wish to cite from it. Please check the document version below.

Document version
Publisher's PDF, also known as Version of record

Publication date:
2008

Link to publication in University of Groningen/UMCG research database

Citation for published version (APA):

Copyright
Other than for strictly personal use, it is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license (like Creative Commons).

Take-down policy
If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Downloaded from the University of Groningen/UMCG research database (Pure): http://www.rug.nl/research/portal. For technical reasons the number of authors shown on this cover page is limited to 10 maximum.
Chapter 3

Syntax-based distributional similarity

Part of the material in this chapter has been published as Van der Plas and Bouma (2005a).

3.1 Introduction

The approach described in this chapter (and the following two methodological chapters) builds on the idea that semantically related words occur in similar contexts. The idea that semantically related words are distributed similarly over contexts is referred to by the term: DISTRIBUTIONAL HYPOTHESIS. Harris (1968) claims that, ‘the meaning of entities and the meaning of grammatical relations among them, is related to the restriction of combinations of these entities relative to other entities.’ This is in line with the Firthian saying that, ‘You shall know a word by the company it keeps.’ (Firth, 1957). In other words, you can grasp the meaning of a word by looking at its contexts.

Context can be defined in many ways. In this chapter we look at the syntactic contexts a word is found in. For example, the verbs that are in a subject relation with a particular noun form a part of its context. In accordance with the Firthian tradition these contexts can be used to determine the semantic relatedness of words. For instance, words that occur in a subject relation with the verb bark have something in common: they are dogs or when bark is used metaphorically they are angry persons.

We explained that Kilgarriff and Yallop (2000) use the terms loose and tight to refer to the different types of semantic similarity that result from using different types of context to determine distributional similarity. Methods using
syntactic information have the tendency to generate tighter thesauri, putting words together that are in the same semantic class, i.e. words that are the same kind of things. Such methods would recognise a semantic similarity between doctor and dentist (both professions, persons, ...), but not between doctor and hospital or disease. The reason for this is the fact that doctor and hospital are not found in the same syntactic contexts. Doctors talk, walk, and are asked. Hospitals do not talk nor walk. They are built, for example.

Also, the nearest neighbours found by syntax-based methods belong to the same syntactic category. Verbs are found as the nearest neighbours of verbs and nouns of nouns. This is the result of the fact that, as we explained in section 1.3, we are interested in second-order affinities between words. We are looking for words that share the same contexts. Words that belong to different syntactic categories will by consequence be found in different syntactic contexts. A noun such as aardbei ‘strawberry’ will appear in different contexts than an adjective such as zoet ‘sweet’. In fact they appear in each other’s syntactic context. aardbei will have zoet_adj as a feature and zoet will have aardbei_adjrev as a feature. There exists a first-order affinity between them, but not a second-order affinity. They do not share the same contexts.

This is different from methods that use unstructured text as context, i.e. the proximity-based methods. When using these methods, the word aardbei ‘strawberry’ can have zoet ‘sweet’ as a nearest neighbour, since both words appear in the same proximity-based contexts, i.e. they both appear with words in the same sentence.

We started our research with syntax-based methods because we deemed tighter relations more helpful for the task at hand: question answering. We have seen in chapter 2 that components, such as answer selection and extraction, require tighter semantic relations such as synonymy. We expect that syntactic methods are particularly apt at finding words that are rather tightly related semantically.

However, we also know from previous work in the field that we cannot expect the syntactic methods to find only synonyms. The nearest neighbours resulting from the syntax-based method are not that tight. Often co-hyponyms are among the lists of nearest neighbours as well as hyponyms and hypernyms and even antonyms. The reason for this is the fact that words such as wine and beer are often found in the same syntactic contexts: the direct object of drink, modified by the adjective non-alcoholic etc.

The aim of this chapter is to show the nature of the nearest neighbours found by the syntactic approach. We will first compare various similarity measures and weights. However, we should keep in mind that our main goal is not to find the best measures and weights; we seek to give a typology of the nearest
neighbours retrieved by the syntactic methods as opposed to those retrieved by other methods. This will help us to decide which type of lexico-semantic information is most useful for which components of our QA system. We will also show that combining multiple syntactic relations is beneficial for the quality of the nearest neighbours and provide scores resulting from corpora of different size.

In the next section (3.2) we will discuss the syntactic methods in greater detail. The following sections will be concerned with the methodology used in our experiments (3.3), the evaluation framework (3.4), the results (3.5), and finally the conclusion (3.6).

3.2 Syntax-based methods

In this section we explain the syntactic approaches to distributional similarity. We will give some examples of syntactic contexts (3.2.1) and we will explain how measures and weights serve to determine the similarity of these contexts (3.2.2). We end this section with a discussion of related work (3.2.3).

3.2.1 Syntactic context

Words that are distributionally similar are words that share a large number of contexts. One can define the context of a word in several ways. In this chapter we will explain the syntactic context. In this case, the words with which the target word is in a syntactic relation form the context of that word.

Most research has been done using a limited number of syntactic relations (Lee, 1999; Weeds, 2003). However, Lin (1998a) shows that a system that uses a range of syntactic relations surpasses Hindle’s (1990) results, which were based on using information from just the subject and object relation. We use several syntactic relations: subject, object, adjective, coordination\(^1\), apposition, and prepositional complement. In Table 3.1 examples are given for these types of syntactic relations. In section 3.3.1 we will explain how we collected these syntactic relations.

3.2.2 Measures and feature weights

Co-occurrence vectors, such as the vector given in Table 3.2 for the headword *kat*, are used to find distributionally similar words. Every cell in the vector

\(^{1}\)The reader might find it surprising to find the coordination relation among the syntactic relations that establish a direct link between two words. In the case of coordination we established a direct link (coordination) between the two elements of the coordination. In Table 3.1, *Jip* and *Janneke*. 
Chapter 3. Syntax-based distributional similarity

Table 3.1: Types of syntactic relations extracted

<table>
<thead>
<tr>
<th></th>
<th>Subject</th>
<th>Object</th>
<th>Adjective</th>
<th>Adverb</th>
<th>Preposition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subj</td>
<td>De kat eet.</td>
<td>'The cat eats.'</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Obj</td>
<td>Ik voer de kat.</td>
<td>'I feed the cat.'</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Adj</td>
<td>De langharige kat loopt.</td>
<td>'The long-haired cat walks.'</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Coord</td>
<td>Jip and Janneke spelen.</td>
<td>'Jip and Janneke are playing.'</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Appo</td>
<td>De clown Bassie lacht.</td>
<td>'The clown Bassie is laughing.'</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Prep</td>
<td>Ik begin met mijn werk.</td>
<td>'I start with my work.'</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 3.1 refers to a particular SYNTACTIC CO-OCCURRENCE TYPE, for example *kat* ‘cat’ in object relation with *voer* ‘feed’. The values of these cells indicate the number of times the co-occurrence type under consideration is found in the corpus. In the example *kat* ‘cat’ is found in object relation with *voer* ‘feed’ 10 times. In other words, the CELL FREQUENCY for this co-occurrence type is 10.

The first column of this vector shows the HEADWORD, i.e. the word for which we determine the contexts it is found in. Here, we only find *kat* ‘cat’. The first row shows the contexts that are found, i.e. the syntactic relation plus the accompanying word. These contexts are referred to by the terms FEATURES or ATTRIBUTES.

Each co-occurrence type has a cell frequency. Likewise each headword has a ROW FREQUENCY. The row frequency of a certain headword is the sum of all its cell frequencies. In our example the row frequency for the word *kat* ‘cat’ is 66. Cut-offs for cell and row frequency can be applied to discard certain infrequent co-occurrence types or headwords, respectively. We use cutoffs because we have too little confidence in our characterisations of words with low frequency. For example the adjective *volautomatisch* ‘fully automatic’ seems to be a peculiar feature for *kat* ‘cat’. A cut-off of 2 will discard the co-occurrence of *volautomatisch* ‘fully automatic’ with *kat* ‘cat’.

<table>
<thead>
<tr>
<th></th>
<th>heb_obj</th>
<th>voer_obj</th>
<th>langharig_adj</th>
<th>volautomatisch_adj</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>kat</em> ‘cat’</td>
<td>50</td>
<td>10</td>
<td>5</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 3.2: Syntactic co-occurrence vector for *kat*

The more similar the vectors are for any two headwords, the more distributionally similar the headwords are. We need a way to compare the vectors for any two headwords to be able to express the similarity between them by means of a score. Various methods can be used to compute the distributional similarity between words. Weeds (2003) gives an extensive overview of existing measures. We will explain in section 3.3.3 what measures we have chosen in the current experiments.
3.2. Syntax-based methods

The results of vector-based methods can be further improved if we take into account the fact that not all combinations of a word and syntactic relation have the same information value. A large number of nouns can occur as the subject of the verb *hebben* ‘have’. The verb *hebben* is selectionally weak (Resnik, 1993) or a light verb. A verb such as *voer* ‘feed’ on the other hand occurs much less frequently, and only with a restricted set of nouns as direct object. Intuitively, the fact that two nouns both occur as subject of *hebben* tells us less about their semantic similarity than the fact that two nouns both occur as the direct object of *feed*.

To account for this intuition, the frequency of occurrence of a particular feature in combination with a certain noun can be weighted by using a weighting function. The value thus achieved is an indication of the amount of information carried by that particular combination of a word, the syntactic relation, and the word heading the syntactic relation. We will explain what weights we have used in the experiments in section 3.3.4.

Our methods for computing distributional similarity between two words consist of a measure for assigning weights to the co-occurrence types (cells) present in the vector and a measure for computing the similarity between two (weighted) co-occurrence vectors.

3.2.3 Related work

Syntax-based distributional methods have been around for some time. The work in this field varies in a number of respects. In general the size of the corpora used in these works has grown. Also, recent work has often included a large number of measures and weights. The syntactic relations included in the data is another point of difference. Some work is limited to just one syntactic relation (Weeds and Weir, 2005; Pereira et al., 1993), whereas other work uses many (Lin, 1998a; Curran and Moens, 2002; Padó and Lapata, 2007).

The evaluation framework used to evaluate the nearest neighbours has undergone some changes over the last few years. Early work was often limited to showing some convincing examples (Hindle, 1990; Grefenstette, 1994b). Some work uses gold standards that are a combination of several dictionaries or thesauri (Lin, 1998a; Curran and Moens, 2002; Weeds and Weir, 2005). Other work evaluates the outcome of the system on a task, such as pseudo disambiguation, (Pereira et al., 1993; Dagan et al., 1999; Lee, 1999; Weeds and Weir, 2005) or similarity-based language modelling (Dagan et al., 1999). The test sets used in early evaluations only comprised a set of very frequent words. In later studies (Curran and Moens, 2002; Weeds and Weir, 2005) the test set is composed of words distributed over various frequency bands.
We will provide a short summary of the previous work known to us. We conclude with discussions of some work for languages other than English.

Hindle (1990) gives examples of noun classifications from subject and object relations. The corpus used to extract these relations is a 6 million-word corpus. Similarity is defined as a combination of object and subject similarity in terms of minimum shared co-occurrence weights. These are determined by the mutual information (MI) of verbs and arguments. We will explain in section 3.3.4 how the mutual information of a headword and its features are determined.

Ruge (1992) uses the head/modifier relation in noun phrases to determine the distributional similarity between words. She extracts these from a corpus of 200K patent abstracts (130MB of text). Only the 30 most frequent heads and modifiers of every term are taken into account. A comparison of similarity measures is performed on manually selected synonyms.

A task-based evaluation of distributional similarity is given in Pereira et al. (1993). Object relations are extracted from a 44 million-word newswire corpus. The Kulback-Leibler measure is used to compare the vectors of headwords and to retrieve clusters of distributionally similar words. The predictive power of the extracted clusters is evaluated on a decision task in which the system has to judge which of two verbs $v$ or $v'$ is more likely to take the given noun as an object.

Grefenstette’s (1994) work is concerned with semantic axes expressing nuances of a word’s meaning, distinguishing its corpus-based meanings. Subject, object, and direct object relations are considered. Examples are given of groupings applied to the most frequent words in the 6 MBytes of Wall Street Journal articles on mergers and a corpus of medical abstracts.

Lin’s (1998a) work is not restricted to one or two of the previously mentioned syntactic relations. He uses several: noun modifier, noun being modified, preposition etc. A combination of corpora is used (64 million words) including the Wall Street Journal corpus, San Jose Mercury, and AP Newswire. Also, he compares several similarity measures. The evaluation is done on 4,294 nouns with a frequency of 100 or higher. For the purpose of evaluating, the gold standards WordNet (Fellbaum, 1998) and Roget’s thesaurus (Roget, 1911) are turned into a ranked list, similar to the neighbours provided by the system. Then the ratio between the scores of common neighbours and other neighbours is determined. Several interesting conclusions are drawn from his experiments: He shows that using several syntactic relations improves results in a comparison with Hindle (1990). Another interesting conclusion is that the distributionally similar words are more closely related to WordNet than Roget is. Roget is known to be a rather loose thesaurus. An example from Kilgarriff and Yallop (2000) is the inclusion of nouns such as churl and wench, and adjectives such as
boorish and provincial under the section headed bush. These semantic relations can be useful for typical thesaurus use, but they go beyond the relations such as synonymy and hypernymy. As we explained in the first introductory paragraph of this chapter, we do not expect this type of semantic relations to result from syntax-based distributional methods.

Dagan et al. (1999) evaluate models based on distributional word similarity on two tasks: language modeling and pseudo-word disambiguation. The authors use 44 million words of 1988 Associated Press newswire to extract noun-verb pairs in direct object relation. They selected noun-verb pairs for the 1000 most frequent nouns only. Statistically significant though relatively modest improvements are achieved over a bigram back-off model for the task of language modeling. Similarity-based methods performed much better in a detailed study concerning a word sense disambiguation task.

Lee (1999) evaluates a large number of distributional similarity measures in the context of distance-weighted averaging (also known by the term similarity-based smoothing). It is an approach to solve the problem of unseen occurrences that arrives at estimates by combining estimates for co-occurrences involving similar words. She uses the object relation only and evaluates on a frequency-controlled pseudo word disambiguation task considering the 1000 most frequent nouns in her data. She concludes from her experiments that similarity measures that focus on the intersection of the features of two headwords result in better performance.

Curran and Moens (2002) ran a large-scale evaluation of different similarity measures and weights. They used several syntactic relations: subject, direct and indirect object, noun and adjectival modifiers, and prepositional phrase. The corpus they used for the extraction of these is the BNC corpus (approximately 100 million words). Evaluation is done on a randomly, but carefully stratified test set of 70 terms covering a range of values for the properties frequency, number of senses, specificity, and concreteness. A gold standard is created from the union of the synonyms of three thesauri: the Macquarie (Bernard, 1990), Roget’s (Roget, 1911), and Moby (Ward, 1996). Several measures of system performance are given: direct matches between neighbours and synonyms from the gold standard, precision of the top \( n \) synonyms, and inverse rank, i.e. the sum of the inverse ranks of each matching synonym. The combination of \( t \)-test and Dice\textsuperscript{†}, a variant of Dice gives the best results.

Weeds and Weir (2005) introduce a framework for lexical distributional similarity: co-occurrence retrieval (CR), a parameterized framework for calculating distributional similarity. They compare this framework in several set-
Chapter 3. Syntax-based distributional similarity

tings to existing measures and weights. They use data from the BNC corpus (100 million words) and they limit their experiments to the object relation. The system is evaluated on a test set of 2,000 words. The test set is composed of the 1000 most frequent words in their data and low-frequency words (words at frequency ranks 3001-4000). Two evaluation methods are applied: a WordNet prediction task and pseudo-disambiguation. They find no significant differences between using the $t$-test or mutual information (MI) as weights in the WordNet prediction task. The best performing measure is the additive CR model.

Padó and Lapata (2007) have compared the performance of a traditional word-based model, i.e. proximity-based model, with a syntax-based model on three tasks: semantic priming, synonymy detection and word sense disambiguation. The corpus used to select 14 dependency relations is the BNC corpus. An interesting aspect of their framework is that the combination of several syntactic relations in one dependency path is allowed. Syntactically enriched models outperform the word-based models in all cases. Medium syntactic content, when combined with a path value function that penalizes longer paths, yields consistently better performance than other syntactic contexts. Medium syntactic content has dependency paths of length 3 or less. They cover phenomena such as coordination, genitive construction, noun compounds, and different kinds of modification.

All the work described above is focused on the English language. There are however many researchers working on languages other than English to find distributionally similar words.

Gasperin et al. (2001) use a parsed Brazilian Portuguese corpus of 1.4 million words. The authors are concerned with the quality and preciseness of the syntactic features. They use several syntactic relations. The evaluation is done manually and some examples are given.

Bourigault and Galy (2005) present an explorative study of using distributional similarity to extract synonyms for French. They use two corpora: the 200 million-word corpus of newspaper text from Le Monde, and a 30 million-word corpus consisting of 515 twentieth century novels. Several syntactic relations are extracted. They evaluate the nearest neighbours found on the Dictionnaire Electronique de Synonymes (DES, Ploux and Manguin (1998, released 2007)). The authors note that for the large corpus only a small percentage of synonyms from the DES are found in their data (22%), and more importantly, that only a very small proportion of the nearest neighbours found are actually synonyms (1%). For the smaller corpus the numbers are 10% and 3%, respectively. Analyses for two interesting phenomena in synonym discovery are given: They show how distributional methods can help in the contextualization of synonymy and how different senses of words can be found by using different corpora for syn-
3.3 Methodology

In the following subsections we describe the set up for our experiments. We describe the corpora we have used and the syntactic relations we extracted from them (3.3.1). In the subsections 3.3.3 and 3.3.4 we describe which similarity measures and weights we have applied, respectively.

3.3.1 Data collection

As our data we used 500 million words of Dutch newspaper text: the Twente Nieuws Corpus (TwNC, Ordelman (2002)) that is parsed automatically using the Alpino parser (Van Noord, 2006). The result of parsing a sentence is a dependency graph according to the guidelines of the Corpus of Spoken Dutch (Moortgat et al., 2000). In later sections we will encounter the CLEF corpus, an 80 million-word corpus of Dutch newspaper text used in the Cross Language Evaluation Forum (CLEF). The forum runs a series of evaluation campaigns to test monolingual and cross-language information retrieval systems. The corpus
Chapter 3. Syntax-based distributional similarity

is a subset of the TwNC corpus.

From these dependency graphs, we extracted tuples consisting of the (non-pronominal) head of an NP (either a common noun or a proper name), the dependency relation, and either (1) the head of the dependency relation (for the object, subject, and apposition relation), (2) the head plus a preposition (for NPs occurring inside PPs which are prepositional complements), (3) the head of the dependent (for the adjective and apposition relation) or (4) the head of the other elements of a coordination (for the coordination relation).

The extraction process makes use of a few linguistic features present in the Alpino dependency graphs. For example, in a relative clause starting with a relative pronoun, such as *dat* ‘that’ or *die* ‘who’, the antecedent is extracted by means of an index available in the output of Alpino. For example in the following sentence (1) the noun *jongen* ‘boy’ is found to be the the subject of *groeten* ‘greet’.

(1) De jongen, die me gisteren groette, is weggelopen.
   ‘The boy, who greeted me yesterday, ran away.’

Furthermore, it takes the infinitival verb *openen* ‘to open’ and not the modal verb *proberen* ‘try’ to construct the tuple ⟨fles, obj, open⟩ instead of ⟨fles, obj, probeer⟩ in the following example:

(2) Ik probeerde de fles te openen.
   ‘I tried to open the bottle.’

The number of ⟨Noun,Relation,OtherWord⟩ triples (tokens) and the number of non-identical triples (types) found are given in Table 3.3. We will refer to these tokens and types based on syntactic relations as SYNTACTIC CO-OCCURRENCE TOKENS and TYPES. Not surprisingly, the subject relation results in the largest list of dependency triples. The prepositional complement is the least frequent dependency relation found in the corpus. Note that a single coordination can give rise to various dependency triples, as from a single coordination such as *bier, wijn, en noten* ‘beer, wine, and nuts’ we extract the triples ⟨bier, coord, wijn⟩, ⟨bier, coord, noten⟩, ⟨wijn, coord, bier⟩, ⟨wijn, coord, noten⟩, ⟨noten, coord, bier⟩, and ⟨noten, coord, wijn⟩. Similarly, from the apposition *premier Kok* ‘prime minister Kok’ we extract both ⟨premier, app, Kok⟩ and ⟨Kok, app, premier⟩. These two syntactic relations hold between words of the same syntactic category, so we include both direction in our evaluations. We use ⟨app, Kok⟩ as a feature for premier. And vice versa we use ⟨app, premier⟩ as a feature for Kok.

In our experiments we have chosen to disregard hapaxes, i.e. occurrences
of 1. After removing the hapaxes we are left with a total of roughly 7M co-
occurrence types, combinations of one of the roughly 433K words and one of
the 554K distinct features. A simple calculation shows us that the matrix has
approximately 239.925M cells of which only 0.003% (7.1M co-occurrence types)
is filled in. This indicates that the matrix is very sparse.

<table>
<thead>
<tr>
<th>Syntactic relation</th>
<th># tokens</th>
<th># types</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subject</td>
<td>28.2M</td>
<td>2.3M</td>
</tr>
<tr>
<td>Adjective</td>
<td>16.5M</td>
<td>1.3M</td>
</tr>
<tr>
<td>Object</td>
<td>14.0M</td>
<td>1.1M</td>
</tr>
<tr>
<td>Apposition</td>
<td>6.0M</td>
<td>1.1M</td>
</tr>
<tr>
<td>Coordination</td>
<td>5.0M</td>
<td>753K</td>
</tr>
<tr>
<td>Prep. compl.</td>
<td>4.1M</td>
<td>499K</td>
</tr>
<tr>
<td>All</td>
<td>73.8M</td>
<td>7.1M</td>
</tr>
</tbody>
</table>

Table 3.3: Number of co-occurrence tokens and co-occurrence types extracted
per syntactic relation (hapaxes excluded)

In Figure 3.1 we see the number of co-occurrence types that are left over
when augmenting the cell frequency cutoff. We have gathered numbers for
several cutoffs (2, 3, 5, 10, 100, 1000, 10K, and 50K). The data points are
plotted on a log scale. The first half of the curve corresponds to a power law,
since the distribution appears to be (nearly) linear on a log-log scale. This means
that infrequent occurrences are extremely common, whereas frequent instances
are extremely rare.

Zipf’s Law (Zipf, 1949) states that the frequency of the nth occurrence of the
event is inversely proportional to it’s rank. On a log-log scale the distribution is
roughly linear and the slope is -1. In this figure we have drawn the cell frequency
cutoff against the number of co-occurrence types, which comes down to plotting
the frequency of the co-occurrence types against the rank. At cell frequency
cutoff 10K we are left with 153 co-occurrence types. These are co-occurrence
types at ranks 1-153. Below the frequency cutoff 1000 the distribution is nearly
linear, but the slope is a little less steep than -1. This means that the frequencies
are lower than expected on the basis of their Zipf rank. Frequencies are not
decreasing as rapidly as we would expect, when going down in the ranked list of
co-occurrence types. This is even more obvious for the most frequent words. It
is known (Manning and Schütze, 1999) that Zipf’s Law is often a bad fit for low
and high frequency words. In a study of frequency distributions of single words
in Alice in Wonderland Baayen (2001) shows that frequencies of the words at
the lowest ranks deviate substantially from expected values according to Zipf’s
Law. We can conclude that the distribution of syntax-based co-occurrence data
is rather similar to the distribution of single words.
Chapter 3. Syntax-based distributional similarity

3.3.2 Definitions

We have chosen to describe the functions used in this chapter using an extension of the notation used by Lin (1998a), adapted by Curran (2003). Co-occurrence data is described as relation tuples: \( \langle \text{word}, \text{relation}, \text{word}' \rangle \), for example, \( \langle \text{cat}, \text{obj}, \text{have} \rangle \).

Asterisks indicate a set of values ranging over all existing values of that component of the relation tuple. For example, \( (w, *, *) \) denotes for a given word \( w \) all relations it has been found in with any other word. For the example of \textit{cat} in Table 3.4, this would denote all values for all syntactic context the word is found in: \( \text{have}_\text{obj}:50, \text{feed}_\text{obj}:10, \text{long-haired}_\text{adj}:5 \), but not \( \text{walk}_\text{obj} \).

Everything is defined in terms of co-occurrence data with non-zero frequencies.

The set of attributes or features for a given corpus is defined as:

\[
(w, *, *) \equiv \{(r, w')|\exists (w, r, w')\}
\]
3.3. Methodology

In the example in Table 3.4 the \( r:w' \) pairs are obj:heb, adj:langharig etc. Each pair yields a frequency value, and the sequence of values is a vector indexed by \( r:w' \) values, rather than natural numbers. A subscripted asterisk indicates that the variables are bound together:

\[
\sum (w_m, *r, *w') \times (w_n, *r, *w')
\]

The above refers to a dot product of the vectors for word \( w_m \) and word \( w_n \) summing over all the \( r:w' \) pairs that these two words have in common. For example we could compare the vectors for cat and dog in Table 3.4 by applying the dot product to all bound variables, i.e. have\(_{obj}\), feed\(_{obj}\), and long-haired\(_{adj}\).

We explained in 3.2.2 that some attributes contain more information than other attributes, for example a verb such as feed contains more information than a light verb such as have. We want to account for that using a weighting function, that will modify the cell values. There is a placeholder for the weighting function:

\[
\sum \text{weight}(w_m, *r, *w') \times \text{weight}(w_n, *r, *w')
\]

This is an abbreviation of:

\[
\sum_{(r,w')\in (w_m,*,*) \cap (w_n,*,*)} \text{weight}(w_m, r, w') \times \text{weight}(w_n, r, w')
\]

3.3.3 Similarity measures

To compare the vectors of the headwords we need similarity measures. We have limited our experiments to using Cosine and Dice\(^\dagger\), a variant of Dice. We chose these methods, since they performed best in a large-scale evaluation experiment reported in Curran and Moens (2002). We will now explain these measures in greater detail.

Cosine is a geometrical measure. It returns the cosine of the angle between the vectors of the words and is calculated as the dot product of the (in this case, weighted) vectors:

\[
\text{Cosine} = \frac{\sum \text{weight}(W1, *r, *w') \times \text{weight}(W2, *r, *w')}{\sqrt{\sum \text{weight}(W1, *, *)^2} \times \sqrt{\sum \text{weight}(W2, *, *)^2}}
\]

If the two words have the same distribution the angle between the vectors is zero. The maximum value of the Cosine measure is 1. Weight is the placeholder for the weighting function we have used. It will be discussed in the next section.

Dice is a combinatorial measure that underscores the importance of shared features. It measures the ratio between the size of the intersection of the two
Chapter 3. Syntax-based distributional similarity

feature sets and the sum of the sizes of the individual feature sets. It is defined as follows:

\[ \text{Dice}(A, B) = \frac{2 \cdot |A \cap B|}{|A| + |B|} \]

where \( A \) stands for the set of features of the first word (W1) and \( B \) for the set of features of the second word (W2).

Curran and Moens (2002) propose a variant of Dice, which they call Dice\( \dagger \). It is defined as:

\[ \text{Dice}^{\dagger} = \frac{2 \sum \min(\text{weight}(W1, *, r, *, w'), \text{weight}(W2, *, r, *, w'))}{\sum \text{weight}(W1, *, r, *, w')} + \text{weight}(W2, *, r, *, w')} \]

Whereas Dice does not take feature weights into account, Dice\( \dagger \) does. For each feature two words share, the minimum is taken. If W1 occurred 15 times with relation \( r \) and word \( w' \) and W2 occurred 10 times with relation \( r \) and word \( w' \), it selects 10 as the minimum (if weighting is set to 1). Note that Dice\( \dagger \) gives the same ranking as the well-known Jaccard measure, i.e. there is a monotonic transformation between their scores. Dice\( \dagger \) is easier to compute and therefore the preferred measure (Curran and Moens, 2002).

3.3.4 Weights

To take into account that certain features are more informative than others we need weights. We used pointwise mutual information (MI, Church and Hanks (1989)) and the \( t \)-test as weights. Raw frequency was used as a baseline. It simply assigns every co-occurrence type a weight of 1 (i.e. every frequency count in the matrix is multiplied by 1).

Pointwise mutual information (MI) measures the amount of information one variable contains about the other. In this case it measures the relatedness or degree of association between the target word and one of its features. For a word \( w \), a syntactic relation \( r \) and another word \( w' \), e.g. the word \( ziekte \) ‘disease’, the adjective relation and the word \( besmettelijk \) ‘contagious’, MI is computed as follows:

\[ \text{MI} = \log \frac{P(w, r, w')}{P(w, *, *)P(*, r, w')} \]

Here, \( P(w, r, w') \) is the probability of seeing \( besmettelijke \) in adjective relation with \( ziekte \) in the corpus, and \( P(w, *, *)P(*, r, w') \) is the product of the probability of seeing \( ziekte \) and the probability of seeing \( besmettelijk \) in an adjective relation with any word.

Applying MI to a co-occurrence matrix will result in a matrix where fre-
quency counts will be replaced by MI scores. The values for cells involving light verbs such as hebben will be lowered, and that the values for informative attributes such as besmettelijke ‘contagious’ will be promoted.

An alternative weight method is the $t$-test. It tells us how probable a certain co-occurrence is. The $t$-test looks at the difference of the observed and expected mean scaled by the standard deviation of the data. The $t$-test takes into account the number of co-occurrences of the bi-gram, e.g. a word $w$ in a syntactic relation $r$ with another word $w'$, relative to the frequencies of the words and features by themselves. Curran and Moens (2002) give the following formulation, which we also used in our experiments:

$$t = \frac{P(w, r, w') - P(w, *, *)P(*, r, w')}{\sqrt{P(w, *, *)P(*, r, w')}}$$

Note that we do not need to include sample size (normally part of the t-statistic) as it is the same for all co-occurrences. All co-occurrences are taken from the same corpus.

There are other weight functions, such as the logarithm of the frequency of co-occurrences (Ruge, 1992), and conditional probability of the feature given the word (Pereira et al., 1993; Dagan et al., 1999). Geffet and Dagan (2004) have even gone further improving the feature vector quality by applying relative feature focus to the vectors. The idea is to promote features that are shared by words that are highly similar to the headword. We will limit our investigations to $t$-test and pointwise mutual information.

3.4 Evaluation

We have explained in 2.5.1 how we plan to evaluate taxonomically related words on the gold standard EuroWordNet (EWN, Vossen (1998)). In section 3.4.1 we will summarize how we translate the distance between two nearest neighbours in EWN to a score. We will explain how we decomposed this overall score into its distinct semantic relations in section 3.4.2. In section 3.4.3 we will explain what test set we have used in the experiments.

3.4.1 EWN similarity measure

For each word we collected its 100 nearest neighbours according to the syntactic co-occurrences found. For each pair of words (target word plus one of the nearest neighbours) we calculated the semantic similarity according to EWN.
We discard words that are not found in EWN in the evaluation framework. Because we know EWN is incomplete, we do not want to penalize for words that are not found in EWN. They might be valuable additions.

The output of the system is a ranked list of nearest neighbours (with the nearest neighbours on top). We compared this output to EWN. We are aware of the fact that using the resource that we are trying to expand as a gold standard is not the optimal solution (as we explained in chapter 2), however, it is the most practical solution for the moment.

There are a number of measures that try to translate the distance between two concepts in WordNet to a score that correlates well with human judgements. We explained in section 2.5.1 that we use Wu and Palmer’s (1994) measure to calculate the relatedness of the words according to EWN. We will repeat below, how we calculated the EWN score.

The Wu and Palmer measure for computing the semantic similarity between two words (W1 and W2) in a word net, whose most specific common subsumer (lowest super-ordinate) is W3, is defined as follows:

\[
Sim = \frac{2(D3)}{D1 + D2 + 2(D3)}
\]

We computed, D1 (D2) as the distance from W1 (W2) to the lowest common ancestor of W1 and W2, W3. D3 is the distance of that ancestor to the root node.

For each pair of a headword and a candidate similar word we calculate the EWN score according to the Wu and Palmer measure. If a word is ambiguous according to EWN, i.e. is a member of several synsets, the highest similarity score is used. The EWN score of a set of word pairs is defined as the average of the similarity between the pairs.

### 3.4.2 Synonyms, hypernyms and (co-)hyponyms

The EWN score, described above, gives an indication of the degree of semantic relatedness in the retrieved neighbours. The fact that it combines several lexical relations is an advantage on the one hand, but on the other hand it is coupled with the disadvantage that it is rather opaque. We discussed this in section 2.5.1. We would like to decompose this score and see how many of the neighbours found by the system are synonyms, and how many are hypernyms or (co-)hyponyms.

The evaluation of the system with respect to the number of synonyms found is pretty straightforward. We simply used the synsets in EWN as our gold standard. If two words are found in the same synsets they are synonymous; else they are not. For hypo- co-hypo- and hypernyms we used the same gold standard. For example, to determine if a candidate word is in a hyponym relation with
the test word we determined if there is one sense of the candidate word and test word that are in a hyponym relation in EWN. If so, this contributes to the hyponym score for that test word. Note that it is possible for one polysemous word to contribute to the percentages of multiple semantic relations. Therefore, the percentages of the several semantic relations added together may be above 100%.

### 3.4.3 Test set

Early work in this area often considered most frequent nouns only. For example Lee (1999) bases the evaluation on the 1000 most frequent nouns. Lin (1998a) considers only nouns with a corpus frequency of more than 100 in a 64 million-word corpus. However, Weeds (2003) has divided the test set up in the 1000 most frequent nouns and 1000 nouns with a lower frequency. Frequencies are taken from the extracted data, i.e. words in the object relation. Curran and Moens (2002) have randomly selected 70 nouns from WordNet so that they cover a range of values for the characteristics frequency, specificity, concreteness, and number of senses. Curran (2003) gives results for a larger test set of 300, covering several frequency bands.

We have chosen to build a large test set of 3000 nouns selected from EWN. We have split up the test set in high-frequency, middle-frequency and low-frequency words. It is expected that frequency is a determining factor for the performance of the system, because there is less data available for infrequent words and similarity calculations based on these limited amounts of data will be less reliable.\(^4\)

We adopt the approach as given by Weeds (2003). However, we take the frequencies from the corpus as a whole, whereas Weeds (2003) uses the context of the object relation to calculate frequencies. Moreover, we add a third test set that includes words of even lower frequencies.

Every occurrence of a word in the corpus (no matter what syntactic relation it is found in) contributes to the frequency for that word. Our choice is motivated by the fact that we plan to compare several methods in the next chapters, not just syntax-based methods. Building the test sets based on the frequency of occurrence of the test words in the syntactic data would bias our results.

For every noun appearing in EWN we have determined its frequency in 80 million-word corpus of newspaper text: the CLEF corpus.\(^5\) The corpus was

\(^4\)Weeds and Weir (2005) report in their conclusions that the performance of low-frequency nouns is not significantly lower than that of high-frequency nouns. However, in the Wordnet prediction task the difference between the low- and high-frequency nouns is large and for some measures even more than 50%.

\(^5\)We have used the CLEF corpus, which is a subset of the TwNC corpus, for frequency calculations, because the proximity-based method presented in Chapter 5 uses this corpus.
Chapter 3. Syntax-based distributional similarity

annotated with PoS-information. We have chosen nouns at ranks 1-1000, 3001-4000 and 9001-10000 as high-frequency, middle-frequency and low-frequency test sets. For the high-frequency test set the frequency ranges from 258,253 (jaar, ‘year’) to 2,278 (scène, ‘scene’). The middle-frequency test set has frequencies ranging between 541 (celstraf, ‘jail sentence’) and 364 (vredesverdrag, ‘peace treaty’). For the test set of infrequent nouns the frequency goes from 91 (charter, ‘charter’) down to 73 (basisprincipe, ‘basic principle’).

3.5 Results

In the current section we will give results for applying the evaluation framework introduced in the previous section.\(^6\) We will first show how frequency cutoffs influence the performance of the system (3.5.1). In section 3.5.2 we will compare combinations of measures and weights. In section 3.5.3 we will show the differences in performance when corpus size increases. We compare the performance of the proximity-based methods (from Chapter 5) to the performance of the syntax-based methods in section 3.5.4. The proportion of synonyms, hyper/hypo, and co-hyponyms in the lists of nearest neighbours will be discussed in 3.5.5. We will show the contribution of the individual syntactic relations in section 3.5.6. We conclude with a comparison to results reported in our previous work.

3.5.1 Cell and row frequency cutoffs

In 3.2.2 we explained that we distinguish row and cell frequencies for a headword and the combination of a headword and an attribute, respectively. The cell frequency indicates how often the combination of the headword and the syntactic context is found in the corpus. The row frequency of a certain headword is the sum of all its cell frequencies that are above the given cell frequency cutoff.

Augmenting the cell frequency cutoffs reduces the number of infrequent syntactic co-occurrences. For reasons of efficiency and noise reduction we discarded hapaxes, i.e. syntactic co-occurrence types that only occurred once in our data. We ran experiments with cell frequency cutoffs 2, 4, and 6.

Augmenting the row frequency cutoffs will result in smaller numbers of infrequent nouns in the lists of nearest neighbours. For example, setting the row frequency cutoff to 60 will result in nearest neighbours that have a total frequency of more than 60 in our data. We experimented with row frequency cutoffs from 2 up to 60.

\(^6\)An interactive demo based on the syntax-based method can be found on http://www.let.rug.nl/gosse/bin/verwant_twnc.py
Table 3.5: Average EWN similarity at the top-$k$ candidates for different cell and row frequency cutoffs

<table>
<thead>
<tr>
<th>Cell Freq</th>
<th>Row Freq</th>
<th>HF $k=1$</th>
<th>HF $k=5$</th>
<th>MF $k=1$</th>
<th>MF $k=5$</th>
<th>LF $k=1$</th>
<th>LF $k=5$</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>2</td>
<td>0.765</td>
<td>0.697</td>
<td>0.737</td>
<td>0.656</td>
<td>0.666</td>
<td>0.620</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>0.765</td>
<td>0.697</td>
<td>0.735</td>
<td>0.655</td>
<td>0.671</td>
<td>0.613</td>
</tr>
<tr>
<td>2</td>
<td>10</td>
<td>0.765</td>
<td>0.697</td>
<td>0.729</td>
<td>0.654</td>
<td>0.665</td>
<td>0.601</td>
</tr>
<tr>
<td>2</td>
<td>20</td>
<td>0.764</td>
<td>0.696</td>
<td>0.729</td>
<td>0.652</td>
<td>0.672</td>
<td>0.600</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>0.766</td>
<td>0.685</td>
<td>0.702</td>
<td>0.627</td>
<td>0.604</td>
<td>0.545</td>
</tr>
<tr>
<td>4</td>
<td>8</td>
<td>0.765</td>
<td>0.685</td>
<td>0.699</td>
<td>0.625</td>
<td>0.593</td>
<td>0.536</td>
</tr>
<tr>
<td>4</td>
<td>20</td>
<td>0.765</td>
<td>0.685</td>
<td>0.700</td>
<td>0.622</td>
<td>0.594</td>
<td>0.527</td>
</tr>
<tr>
<td>4</td>
<td>40</td>
<td>0.765</td>
<td>0.685</td>
<td>0.702</td>
<td>0.621</td>
<td>0.598</td>
<td>0.524</td>
</tr>
</tbody>
</table>

Before discussing the results it should be noted that these tests were done using MI as weight and Cosine as measure. This combination gives the best results in our experiments, as will be shown in the next paragraphs and showing results for all combination of weights and measures would take too much space here. We have used the EWN score to evaluate the settings as this score combines several semantic relations.

We can see in Table 3.5 that evaluations on the high-frequency test set clearly perform best, followed by the middle-frequency test set. The low-frequency test set performs worst. This is in line with expectations since the low-frequency words suffer most from data sparseness. Weeds (2003) reports the same tendency in the WordNet prediction task. Still, the performance of the low-frequency test set outperforms the baseline by far: 0.66 versus 0.26. The EWN similarity of the nearest neighbours found is rather high for all three test sets.

Furthermore we can see that row-frequency and also cell-frequency cutoffs have almost no effect on the test set of high-frequency nouns. This result is expected since high-frequency nouns are less affected by cell or row frequency cutoffs at these values. Words that appear 2,278 up to 258K times in a corpus of 80 million words will appear more often in the syntactic contexts taken from a 500 million-word corpus than the cutoffs used in these experiments. We see that for the middle-frequency test set the scores only drop a little when augmenting the row frequency cutoff. However, augmenting the cell frequency cutoff lowers the scores considerably. For the low-frequency test set the effect is even a little stronger for the row frequency cutoff. This is in line with findings in Curran and Moens (2002). They report a considerable difference in performance, when
augmenting the the cell frequency cutoff from 2 to 4.

We have set both row and cell frequency cutoffs to 2 for the remainder of the experiments in this chapter.

### 3.5.2 Comparing measures and weights

We compared the performance of the various combinations of weight functions (frequency, MI, and t-test) and the measures for computing the similarity between word vectors (Dice† and Cosine). The results are given in Table 3.6. For each of the three test sets EWN scores are given for each combination of a measure and a weight. All combinations significantly outperform the random baseline, i.e. the score obtained by picking 100 random words from EWN as nearest neighbours of a given target word, which is 0.26. Note also that the maximal score is not 1.00, but significantly lower, as words do not have \(k\) synonyms (which would give the hypothetical, maximal score of 1.00).

<table>
<thead>
<tr>
<th>Measure +Weight</th>
<th>HF (k=1)</th>
<th>HF (k=5)</th>
<th>MF (k=1)</th>
<th>MF (k=5)</th>
<th>LF (k=1)</th>
<th>LF (k=5)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dice†+FR</td>
<td>0.647</td>
<td>0.576</td>
<td>0.607</td>
<td>0.545</td>
<td>0.578</td>
<td>0.501</td>
</tr>
<tr>
<td>Dice†+MI</td>
<td>0.747</td>
<td>0.665</td>
<td>0.702</td>
<td>0.629</td>
<td><strong>0.674</strong></td>
<td>0.598</td>
</tr>
<tr>
<td>Dice†+TT</td>
<td>0.757</td>
<td>0.681</td>
<td>0.683</td>
<td>0.628</td>
<td>0.628</td>
<td>0.570</td>
</tr>
<tr>
<td>Cosine+FR</td>
<td>0.685</td>
<td>0.589</td>
<td>0.638</td>
<td>0.569</td>
<td>0.616</td>
<td>0.540</td>
</tr>
<tr>
<td>Cosine+MI</td>
<td><strong>0.765</strong></td>
<td><strong>0.697</strong></td>
<td><strong>0.737</strong></td>
<td><strong>0.656</strong></td>
<td>0.666</td>
<td><strong>0.620</strong></td>
</tr>
<tr>
<td>Cosine+TT</td>
<td>0.741</td>
<td>0.667</td>
<td>0.656</td>
<td>0.604</td>
<td>0.548</td>
<td>0.524</td>
</tr>
</tbody>
</table>

**Table 3.6:** Average EWN similarity at the top-\(k\) candidates for different similarity measures and weights

Cosine in combination with MI gives the best results at all but one point of evaluation, followed by Dice† in combination with MI. The worst performance is attained when Dice† is used without any weighting and the raw frequencies are used to compare word vectors.

It is hard to compare our results to other work for several reasons. First of all, the fact that most work is done for English complicates the comparison. It is not possible to compare on the same gold standard. Dutch EWN is smaller than English Princeton WordNet. Dutch EWN covers about 56% of the nouns in WordNet 5.1. Another difference springs from the nature of the two languages with respect to compounding. Some languages, such as Dutch and German, build compounds orthographically in one word. In English compounds are mostly composed of two words, e.g table cloth and hard disk versus database. Even though we decided to discard multi-word terms we still include the single word compounds in our data and evaluation. In English the word
3.5. Results

*table cloth* will in most of the cases contribute to the data for *cloth*. The same holds for *hard disk*. It is clear that the data for Dutch will suffer more from data sparseness due to this difference. On the other hand the English data will suffer more from ambiguity.

Of the work that compares several similarity measures and weights the evaluation framework Weeds (2003) chooses in chapter 6 is most similar to ours. She evaluates the system on predicting semantic similarity compared to the gold standard WordNet. In her experiments on the high frequency test set Jaccard’s and Cosine’s performance is comparable. For the low-frequency test set Jaccard performs much worse. In our experiments Cosine outperforms Dice for all three test sets. An important difference between her work and ours is the fact that she limits the outcome of the system (nearest neighbours) to the input to their system, 1000 frequent and 1000 less frequent headwords.

In the experiments (without weighting) by Curran and Moens (2002) Dice† performed considerably better than Cosine. Dice† in combination with \( t \)-test performed best. It should be noted that they did not try the combination of Cosine with \( t \)-test, nor MI because they believe the measures and weights to be independent and Dice † performed best with the best-performing weight \( t \)-test. In our experiments Cosine outperforms Dice†. However, they evaluate on a loosely structured gold standard, a combination of three thesauri. We will see in Chapter 5 that the combination Dice† \( t \)-test also performs well when evaluating on association norms, a very loosely structured gold standard. Furthermore, their evaluation is done on a smaller test set (70 words) and a smaller corpus. In Lin (1998a) the Cosine measure did not perform as badly as in Curran and Moens (2002).

The evaluation done by Lee (1999) is very different from ours. The author evaluates on a decision task and considers the 1000 most frequent nouns only. Cosine is in the group of second-best measures, whereas Jaccard (equivalent to Dice) is in the group of the best performing measures.

The absolute EWN scores are even harder to compare to other work than the relative performance of the measures and weights. We have explained in section 3.4.1 that we have chosen to evaluate the top-N nearest neighbours as in Curran (2003). However, Curran (2003) has used a combination of several gold standards among which thesauri that are much looser in nature than Wordnet. Other work that uses WordNet as a goldstandard has often calculated the correlation between the gold standard and the nearest neighbours (Lin, 1998a; Weeds, 2003). The thesaurus is transformed to a ranked list by using a WordNet similarity measure and the correlation between the two ranked lists, one from the thesaurus and one from the system is calculated. Over and above most of the work done is on English.
Work by Van der Cruys (2006) does apply the EWN score in the same way as we did, and he works on Dutch as well. However the scores are based on cluster averages and not top-\(k\) nearest neighbours. The highest Wu-and-Palmer-score achieved in his work is at 1500 clusters and amounts to a score of 60.40\%. This corresponds to on average 3.33 words per cluster. At \(k=5\) we reach a score of 0.697 (69.7\%) for the high-frequency test set.

<table>
<thead>
<tr>
<th>Meas+Weight</th>
<th>HF</th>
<th>MF</th>
<th>LF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dice†+FR</td>
<td>100.0</td>
<td>97.5</td>
<td>100.0</td>
</tr>
<tr>
<td>Dice†+MI</td>
<td>100.0</td>
<td>95.0</td>
<td>100.0</td>
</tr>
<tr>
<td>Dice†+TT</td>
<td>100.0</td>
<td>91.7</td>
<td>100.0</td>
</tr>
<tr>
<td>Cosine+FR</td>
<td>100.0</td>
<td>66.0</td>
<td>100.0</td>
</tr>
<tr>
<td>Cosine+MI</td>
<td>100.0</td>
<td>88.7</td>
<td>100.0</td>
</tr>
<tr>
<td>Cosine+TT</td>
<td>100.0</td>
<td>25.5</td>
<td>100.0</td>
</tr>
</tbody>
</table>

Table 3.7: Coverage and traceability for the various measures and weights

Table 3.7 shows the coverage and traceability for the several test sets and combinations of measures and weights. Coverage is defined as the percentage of test words that are found in the data. A high number indicates that many of the words in the test sets are found in the data. As we explained in section 3.4.3 we used three sets of 1000 words to test our data. In most cases the test words were found in our data. Even for the low-frequency test set only 1 out of 1000 words was not found in our data. We can conclude from this that the coverage of the data is very good. This is to be expected since the test sets are built on the basis of frequency information from the 80 million-word CLEF corpus, a subset of the 500 million-word TwNC corpus. However, not all words, although rather frequent individually, appear in the syntactic relations we have selected.

Traceability is a characteristic that we have introduced as a result of our decision to discard words that are missing in EWN. It is calculated by determining what percentage of words the system proposes are actually found in EWN. Our reasons for this are mainly that we know EWN is incomplete and we do not want to penalize for words that might be valuable additions. The fact that a measure returns many words that are not found in EWN thus does not affect the scores in any way. However, we believe that it is still important to know how many of the nearest neighbours returned by the system are found in EWN. If the traceability is low, the average EWN score will be calculated on a smaller number of pairs and will hence be less reliable.

On the other hand, a low traceability score will show how much can be gained from using the methods to add missing words to existing resources. The results for the low-frequency test set were relatively good, however, as we can
see from the traceability scores in Table 3.7, the percentage is much lower for
the low-frequency test set. This seems to be an indication that resources for
which coverage is a problem could benefit from these methods. However, we
must be cautious, because we are not at all sure that the words not found in
EWN will be good additions.

A low traceability score in combination with a relatively low EWN score
seems to indicate that the nearest neighbours are of low quality. For exam-
ple, we will see some examples later of how the combination of Cosine + t-test
results in infrequent words such as aalbaars ‘something cuddly’ and aandeel-
houderspact ‘shareholders agreement’ for the headword bedenking ‘objection’.
This is a very unwelcome effect that is not enough visible from the EWN score.
This is another reason why we have included traceability as a characteristic.
Very low traceability scores in combination with relatively low EWN scores are
unfavourable.

Dice† in general produces lists of nearest neighbours that are more easily
found in EWN than Cosine. Lee (1999) explains how Jaccard (equivalent to
Dice†) is a measure in which shared features are key. We can see from formulas
given in section 3.3.3 that non-shared features are not taken into account for
Dice†. Although shared features are still key, Cosine also takes non-shared
features into account. We can see that the length of the vectors for both words
are in the denominator of the formula for Cosine. The Cosine measure therefore
seems to prefer words that do not have many co-occurrences. These are often
infrequent words. For example, the word Breakdance is only found in subject
relation with ben ‘am’, word ‘become’, and heb ‘have’. Fusie ‘fusion’, the nearest
neighbour resulting from using the combination Dice† + freq, is found in 1,019
distinct contexts.

For the combination Cosine+t-test the situation is even worse: only around
25% is found in EWN for all test sets. The MI weight improves the results
considerably: Cosine+MI has a traceability score of 88.7%. The MI weight
must have some characteristic that compensates for the behaviour of Cosine. It
is in fact the aim of applying weights to take care of frequently occurring verbs
such as ben ‘am’ word ‘become’, and heb ‘have’. Using the t-test for natural
language problems has been criticised (Church and Mercer, 1993), because it is
based on the assumption that the data is normally distributed, which is never
the case. The t-test is not able to sufficiently downplay the effect of verbs
such as ben ‘am’ and heb ‘have’. The MI measure has been criticised for not
being a very good measure of dependence between words. It has, however,
been said to be a good measure for indicating independence (non-association)
between words (Manning and Schütze, 1999). This is precisely what we need
to downplay occurrences with verbs, such as ben ‘am’ word ‘become’, and heb
Based on these findings we decided to take a closer look at the nearest neighbours returned by the different measures and weights. Examples are given in Table 3.8. The difference between the combinations Dice+\textit{t}-test, Dice+MI and Cosine+MI is not large enough to be seen at first sight by manual inspection. It is only in the quantitative evaluations as presented in Table 3.6 that the differences are visible. However, the inspection did reveal that for the combination Cosine+\textit{t}-test many unrelated, infrequent nouns were returned as nearest neighbours. The combination has the tendency to select low-frequency nouns. The combination of Dice† and \textit{t}-test does not result in many infrequent unrelated nouns. The \textit{t}-test seems to be particularly harmful in combination with Cosine.

Due to the fact that Dice† computes the similarity between words based on shared features, it is less prone to the drawback of the \textit{t}-test.

Based on these evaluations we decided to use the combination Cosine + MI for the rest of our evaluations. It results in high scores in the EWN evaluation and it gives us lists of nearest neighbours, that are found in EWN.
### Table 3.8: Examples of nearest neighbours at the top-3 ranks

<table>
<thead>
<tr>
<th>Test Word</th>
<th>Measure (Freq)</th>
<th>k=1</th>
<th>k=2</th>
<th>k=3</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>HF</strong></td>
<td>dice</td>
<td>fusion</td>
<td>opname</td>
<td>uitbreiding</td>
</tr>
<tr>
<td></td>
<td>+MI</td>
<td>relation</td>
<td>geboorte</td>
<td>fusie</td>
</tr>
<tr>
<td></td>
<td>+Tt</td>
<td>'relation'</td>
<td>'birth'</td>
<td>'fusion'</td>
</tr>
<tr>
<td><strong>Cosine (Freq)</strong></td>
<td>Breakdance</td>
<td>doejong</td>
<td>'80</td>
<td></td>
</tr>
<tr>
<td></td>
<td>+MI</td>
<td>'relation'</td>
<td>'birth'</td>
<td>'engagement'</td>
</tr>
<tr>
<td></td>
<td>+Tt</td>
<td>huwelijksritueel</td>
<td>partnerregistratie</td>
<td>'marriage of convenience'</td>
</tr>
<tr>
<td><strong>MF</strong></td>
<td>dice</td>
<td>lef</td>
<td>beschikking</td>
<td>medelijden</td>
</tr>
<tr>
<td></td>
<td>+MI</td>
<td>bezwaar</td>
<td>aarzelingsbezwaar</td>
<td>‘objection’</td>
</tr>
<tr>
<td></td>
<td>+Tt</td>
<td>misnoegen</td>
<td>bezorgdheid</td>
<td>paranoia-genre</td>
</tr>
<tr>
<td><strong>Cosine (Freq)</strong></td>
<td>regeringsexperiens</td>
<td>‘government experience’</td>
<td>‘discovered’</td>
<td>‘inexactness margin’</td>
</tr>
<tr>
<td></td>
<td>+MI</td>
<td>bezwaar</td>
<td>misnoegen</td>
<td>grief</td>
</tr>
<tr>
<td></td>
<td>+Tt</td>
<td>‘objection’</td>
<td>‘displeasure’</td>
<td>‘objection’</td>
</tr>
<tr>
<td></td>
<td>huwelijksakkoord</td>
<td>aandeelhouderspact</td>
<td>‘shareholder agreement’</td>
<td>‘announced’</td>
</tr>
<tr>
<td><strong>LF</strong></td>
<td>dice</td>
<td>bloeitijd</td>
<td>glorietijd</td>
<td>hoogconcurrentuur</td>
</tr>
<tr>
<td></td>
<td>+MI</td>
<td>‘florescence’</td>
<td>‘golden age’</td>
<td>‘period of’ boom’</td>
</tr>
<tr>
<td></td>
<td>+Tt</td>
<td>‘florencie’</td>
<td>‘golden age’</td>
<td>‘boom’</td>
</tr>
<tr>
<td></td>
<td>+Freq</td>
<td>‘florencie’</td>
<td>‘golden age’</td>
<td>‘boom’</td>
</tr>
<tr>
<td></td>
<td>+MI</td>
<td>‘florencie’</td>
<td>‘golden age’</td>
<td>‘boom’</td>
</tr>
<tr>
<td></td>
<td>+Tt</td>
<td>‘beginner’s problem’</td>
<td>‘constitutionalism’</td>
<td>‘purification process’</td>
</tr>
</tbody>
</table>
3.5.3 Comparing corpora

In previous work (Van der Plas and Bouma, 2005a) we used the 80 million-word CLEF corpus. In this chapter we have seen results based on the 500 million-word TwNC corpus of which the CLEF corpus is a subset. In Table 3.9 we see that the number of syntactic co-occurrence tokens and types is much smaller for the 80 million-word CLEF corpus than for the 500 million-word TwNC corpus. The numbers are the result of adding the number of co-occurrences found for the different syntactic relations, as we have seen in Table 3.3.

In Table 3.10 we have put the results of the two corpora next to each other. As expected, the larger corpus produces better results. The same tendency is found by Curran (2003) in chapter 3. The author reports that the average direct matches rise from 22.6 at a corpus size of 75 million words to 25.3 at a corpus size of 150 million words.

Furthermore we can see from Table 3.10 that low-frequency nouns benefit most from the larger corpus, followed by the middle-frequency nouns. That is in line with expectations, since the low-frequency words suffer most from data sparseness. The high-frequency words are relatively well-presented in the smaller corpus compared to the low-frequency words.

Coverage and traceability for the 80 million-word corpus are lower as can be seen in Table 3.11 and the low-frequency words are most affected. This is again in line with expectations.

<table>
<thead>
<tr>
<th>Corpus</th>
<th># tokens</th>
<th># types</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwNC (500M)</td>
<td>73.8M</td>
<td>7.1M</td>
</tr>
<tr>
<td>Clef (80M)</td>
<td>10.5M</td>
<td>1.4M</td>
</tr>
</tbody>
</table>

Table 3.9: Number of co-occurrences (tokens and types) for the two corpora (hapaxes excluded)

<table>
<thead>
<tr>
<th>Corpus</th>
<th>HF k=1</th>
<th>HF k=5</th>
<th>MF k=1</th>
<th>MF k=5</th>
<th>LF k=1</th>
<th>LF k=5</th>
</tr>
</thead>
<tbody>
<tr>
<td>500M</td>
<td>0.765</td>
<td>0.697</td>
<td>0.737</td>
<td>0.656</td>
<td>0.666</td>
<td>0.620</td>
</tr>
<tr>
<td>80M</td>
<td>0.747</td>
<td>0.680</td>
<td>0.644</td>
<td>0.577</td>
<td>0.488</td>
<td>0.431</td>
</tr>
</tbody>
</table>

Table 3.10: Average EWN score at the top-k candidates for the two corpora

3.5.4 Comparison to proximity-based method

In the introduction to this chapter we explained that there are two widespread methods for finding distributionally similar words, namely, the syntax-based
3.5. Results

<table>
<thead>
<tr>
<th>Corpus</th>
<th>HF Cov.</th>
<th>Trace.</th>
<th>MF Cov.</th>
<th>Trace.</th>
<th>LF Cov.</th>
<th>Trace.</th>
</tr>
</thead>
<tbody>
<tr>
<td>500M</td>
<td>100.0</td>
<td>88.7</td>
<td>100.0</td>
<td>65.3</td>
<td>99.9</td>
<td>32.8</td>
</tr>
<tr>
<td>80M</td>
<td>100.0</td>
<td>87.8</td>
<td>99.9</td>
<td>42.4</td>
<td>98.9</td>
<td>23.1</td>
</tr>
</tbody>
</table>

Table 3.11: Coverage and traceability for the two corpora

<table>
<thead>
<tr>
<th>Corpus</th>
<th># tokens</th>
<th># types</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proximity(80M)</td>
<td>526.4M</td>
<td>34.6M</td>
</tr>
<tr>
<td>Syntax1 (500M)</td>
<td>73.8M</td>
<td>7.1M</td>
</tr>
<tr>
<td>Syntax2 (80M)</td>
<td>10.5M</td>
<td>1.4M</td>
</tr>
</tbody>
</table>

Table 3.12: Number of co-occurrences (tokens and types) for the several corpora (hapaxes excluded)

methods that we have seen in this chapter and the proximity-based method that we will discuss in chapter 5. However, we would like to give a comparison of the two methods here.

The proximity-based method makes use of the 80-million word corpus (CLEF) due to efficiency problems. It is clear from Table 3.12 that the proximity-based method, which uses unstructured context results in more data, although it uses the same corpus: 34.6M types versus 1.4M types for the syntax-based method.

We explained that the proximity-based methods find looser relations, and more associations than the syntax-based methods. We therefore expect the syntax-based methods to perform better when evaluating on a highly structured resource such as EWN. We can see from Table 3.13 that the syntactic methods are indeed better at finding words that are tightly related semantically. However, when the corpora used are equally large, the proximity-based method approaches the performance of the syntax-based method for the low-frequency test set. Data sparseness is a bigger problem for the syntax-based method than for the proximity-based method simply because the contexts are more limited. Both methods still outperform the baseline of randomly assigning

<table>
<thead>
<tr>
<th>Method</th>
<th>HF k=1</th>
<th>HF k=5</th>
<th>MF k=1</th>
<th>MF k=5</th>
<th>LF k=1</th>
<th>LF k=5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Syntax1 (500M)</td>
<td>0.765</td>
<td>0.697</td>
<td>0.737</td>
<td>0.656</td>
<td>0.666</td>
<td>0.620</td>
</tr>
<tr>
<td>Syntax2 (80M)</td>
<td>0.747</td>
<td>0.680</td>
<td>0.644</td>
<td>0.577</td>
<td>0.488</td>
<td>0.431</td>
</tr>
<tr>
<td>Proxi(80M)</td>
<td>0.524</td>
<td>0.493</td>
<td>0.451</td>
<td>0.429</td>
<td>0.401</td>
<td>0.385</td>
</tr>
</tbody>
</table>

Table 3.13: Average EWN score at the top-k candidates for the syntax-based and proximity-based method
nearest neighbours to a headword, which is 0.26.

3.5.5 Distribution of semantic relations

So far we have evaluated the performance of the syntax-based method by calculating the EWN score, a score that combines several semantic relations. We will now decompose this score and check what kind of semantic relations are found among the nearest neighbours.

An important semantic relation for building semantic resources is synonymy. Also, in view of our application, question answering, it seems interesting to see how well the system does on the acquisition of synonyms. From previous work we know that there are many semantic relations other than synonymy among the nearest neighbours found (Weeds, 2003; Bourigault and Galy, 2005; Van der Cruys, 2006).

Table 3.14 shows the proportion of synonyms among the nearest neighbours. Approximately 21% of the nearest neighbours at rank 1 are synonyms. Note that we do not expect to find 100%, because not every word in EWN has one or more synonyms. According to our calculations 60% of the nouns in EWN have one or more synonyms.

We expect to find more related words overall in the high-frequency test set as there is more data for those words. Also, we expect that words in the high-frequency test set have in general more senses and thus more correct lexico-semantic relations. We see higher scores for the high-frequency test set for almost all lexico-semantic relations but very strongly for the hyponym relation. The percentage of hyponyms found is very different for the three test sets. The more frequent the test words are, the more hyponyms are found. This result relates very well to our intuition that frequent words are often more general and thus have a larger set of hyponyms. The percentage of hypernyms found decreases less rapidly because, along the same lines of reasoning, the low frequency words are often less general terms, that typically have more hypernyms than general terms. The decrease in performance is compensated by this counter-effect.

We will try again to compare our scores to previous work. The scores reported by Bourigault and Galy (2005) are based on all nearest neighbours found above a certain similarity threshold. For the newspaper corpus only 1% of the neighbours above that similarity threshold are synonyms.

Curran and Moens (2002) report a proportion of 76% of synonyms at the first rank. The largest difference between their work and ours is the results of the gold standard used. They use a combination of several rather loose thesauri. The lists of synonyms are therefore much larger, and this is reflected in the
3.5. Results

For Dutch Van der Cruys (2006) gives rather low scores for percentage of synonyms found. However, his scores are based on cluster averages and not top-\(k\) nearest neighbours and he uses a limited number of syntactic relations. At 1600 clusters for the 5000 (most frequent) nouns, that is on average 3.1 words per cluster, he reports 6.98% of synonyms. At \(k=5\) we reach a precision of approximately 11%. As in his case, the largest part is taken by co-hyponyms. He finds twice as many hyponyms than synonyms and hypernyms. We also find more hyponyms than hypernyms for high-frequency nouns. His evaluations are done on the most frequent nouns only, so this result is expected.

<table>
<thead>
<tr>
<th>Semantic Relation</th>
<th>HF (k=1)</th>
<th>HF (k=5)</th>
<th>MF (k=1)</th>
<th>MF (k=5)</th>
<th>LF (k=1)</th>
<th>LF (k=5)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Synonyms</td>
<td>21.31</td>
<td>10.55</td>
<td>22.97</td>
<td>10.11</td>
<td>19.21</td>
<td>11.63</td>
</tr>
<tr>
<td>Hypernyms</td>
<td>11.95</td>
<td>7.35</td>
<td>8.42</td>
<td>6.43</td>
<td>5.79</td>
<td>4.12</td>
</tr>
<tr>
<td>Hyponyms</td>
<td>20.74</td>
<td>17.34</td>
<td>7.20</td>
<td>5.17</td>
<td>3.05</td>
<td>2.80</td>
</tr>
<tr>
<td>Co-hyponyms</td>
<td>41.71</td>
<td>32.74</td>
<td>43.03</td>
<td>30.29</td>
<td>37.80</td>
<td>31.42</td>
</tr>
</tbody>
</table>

Table 3.14: Distribution of semantic relations over the top-\(k\) candidates

3.5.6 Comparing syntactic relations

In Table 3.15, the performance of the data collected using various syntactic relations is compared. Scores are given for the adjective relation (adj), the object relation (obj), the subject relation (subj), the prepositional complement relation (prep), the coordination relation (coord) and the apposition relation (appo). Examples of these syntactic relations were given in the second section of this chapter (3.2.1). The last row of each test set is reserved for the combination of all syntactic relations (all).

The performance of the several relations is partly influenced by the amount of data the syntactic relation accounts for and partly by the nature of the relation.

The adjective relation is one that describes the attributes of an entity and is a very good relation to use when trying to find semantically related words. The object relation describes what is done to entities and is again a very good feature to use for acquiring semantically related words. In 3.16 we have duplicated Table 3.3 from section 3.3.1 for the reader’s convenience. We can see from Table 3.16 that both relations are rather frequent.

The subject relation is much more frequent. It describes what actions entities are taking. It is less good at determining the semantic relatedness between words. When inspecting the nearest neighbours found by using the subject and object relation only (Table 3.17), we see that the subject relation is good
Table 3.15: Average EWN similarity at top-k candidates for various syntactic relations

<table>
<thead>
<tr>
<th>Syntactic relation</th>
<th># tokens</th>
<th># types</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subject</td>
<td>28.2M</td>
<td>2.3M</td>
</tr>
<tr>
<td>Adjective</td>
<td>16.5M</td>
<td>1.3M</td>
</tr>
<tr>
<td>Object</td>
<td>14.0M</td>
<td>1.1M</td>
</tr>
<tr>
<td>Apposition</td>
<td>6.0M</td>
<td>1.1M</td>
</tr>
<tr>
<td>Coordination</td>
<td>5.0M</td>
<td>753K</td>
</tr>
<tr>
<td>Prep. compl.</td>
<td>4.1M</td>
<td>499K</td>
</tr>
<tr>
<td>All</td>
<td>73.8M</td>
<td>7.1M</td>
</tr>
</tbody>
</table>

Table 3.16: Number of co-occurrence tokens and co-occurrence types extracted per syntactic relation (hapaxes excluded)

at finding nearest neighbours that are animate things, things that are rather active, such as hartpatient ‘heart patient’. It is less good at finding nearest neighbours for less active inanimate things, such as verminking ‘mutilation’, as these are less often found in subject position. The fact that the subject relation is not the best performing relation shows that the number of occurrences the relation accounts for is not the determining factor in all cases.

Table 3.17: Examples of nearest neighbours for the object and subject relation

The prepositional complement is a variant of the object relation. However, it performs less well due to the fact that it is a less frequent syntactic relation.
that hence results in less data. Especially for the low-frequency test set, where the scores are heavily depressed by data sparseness, this relation scores lowest of all.

The coordination relation is not very good at finding semantically related words, but it is not very frequent either. However, when taking a closer look at its performance on several semantic relations individually (Table 3.18), we find that it is bad at finding synonyms (penultimate position), hypernyms (penultimate position), and hyponyms (last position), but it is rather good at finding co-hyponyms (third position), although it is one of the least frequently found syntactic relations. The coordination relation is a relation that links co-hyponyms in sentences, e.g. apples and pears, bear and wine, salt and pepper.

It is to be expected that this relation will do well on finding co-hyponyms.

<table>
<thead>
<tr>
<th>Semantic Relation</th>
<th>Syntactic Relation</th>
<th>HF k=1</th>
<th>HF k=5</th>
<th>MF k=1</th>
<th>MF k=5</th>
<th>LF k=1</th>
<th>LF k=5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Synonyms</td>
<td>Adj</td>
<td>21.90</td>
<td>10.49</td>
<td>19.21</td>
<td>9.06</td>
<td>12.55</td>
<td>6.56</td>
</tr>
<tr>
<td></td>
<td>Appo</td>
<td>7.43</td>
<td>4.70</td>
<td>8.68</td>
<td>4.06</td>
<td>5.65</td>
<td>3.31</td>
</tr>
<tr>
<td></td>
<td>Coord</td>
<td>10.03</td>
<td>4.60</td>
<td>7.64</td>
<td>3.57</td>
<td>3.82</td>
<td>2.45</td>
</tr>
<tr>
<td></td>
<td>Obj</td>
<td>17.54</td>
<td>8.46</td>
<td>14.68</td>
<td>6.90</td>
<td>8.15</td>
<td>5.53</td>
</tr>
<tr>
<td></td>
<td>Prep</td>
<td>11.40</td>
<td>5.98</td>
<td>6.10</td>
<td>3.33</td>
<td>2.70</td>
<td>2.13</td>
</tr>
<tr>
<td></td>
<td>Subj</td>
<td>13.22</td>
<td>6.53</td>
<td>6.08</td>
<td>4.38</td>
<td>2.24</td>
<td>2.07</td>
</tr>
<tr>
<td>Hypernyms</td>
<td>Adj</td>
<td>11.94</td>
<td>6.83</td>
<td>7.72</td>
<td>5.33</td>
<td>2.51</td>
<td>2.39</td>
</tr>
<tr>
<td></td>
<td>Appo</td>
<td>3.82</td>
<td>2.47</td>
<td>2.74</td>
<td>1.72</td>
<td>5.65</td>
<td>2.81</td>
</tr>
<tr>
<td></td>
<td>Coord</td>
<td>3.93</td>
<td>2.69</td>
<td>2.12</td>
<td>1.47</td>
<td>1.39</td>
<td>0.98</td>
</tr>
<tr>
<td></td>
<td>Obj</td>
<td>10.77</td>
<td>6.64</td>
<td>5.87</td>
<td>4.22</td>
<td>3.13</td>
<td>2.30</td>
</tr>
<tr>
<td></td>
<td>Prep</td>
<td>8.83</td>
<td>5.14</td>
<td>2.85</td>
<td>1.89</td>
<td>0.68</td>
<td>0.57</td>
</tr>
<tr>
<td></td>
<td>Subj</td>
<td>9.94</td>
<td>6.58</td>
<td>3.68</td>
<td>3.73</td>
<td>1.12</td>
<td>1.12</td>
</tr>
<tr>
<td>Hyponyms</td>
<td>Adj</td>
<td>20.37</td>
<td>17.82</td>
<td>6.21</td>
<td>4.69</td>
<td>2.51</td>
<td>1.49</td>
</tr>
<tr>
<td></td>
<td>Appo</td>
<td>15.29</td>
<td>11.62</td>
<td>4.11</td>
<td>2.03</td>
<td>1.61</td>
<td>0.83</td>
</tr>
<tr>
<td></td>
<td>Coord</td>
<td>7.32</td>
<td>5.90</td>
<td>2.97</td>
<td>1.61</td>
<td>0.69</td>
<td>0.49</td>
</tr>
<tr>
<td></td>
<td>Obj</td>
<td>20.87</td>
<td>15.60</td>
<td>6.39</td>
<td>4.22</td>
<td>2.51</td>
<td>1.31</td>
</tr>
<tr>
<td></td>
<td>Prep</td>
<td>15.87</td>
<td>11.00</td>
<td>2.44</td>
<td>2.46</td>
<td>0.00</td>
<td>0.33</td>
</tr>
<tr>
<td></td>
<td>Subj</td>
<td>15.71</td>
<td>11.36</td>
<td>3.29</td>
<td>2.96</td>
<td>1.87</td>
<td>1.03</td>
</tr>
<tr>
<td>Co-hyponyms</td>
<td>Adj</td>
<td>40.16</td>
<td>29.38</td>
<td>34.09</td>
<td>24.09</td>
<td>26.36</td>
<td>18.19</td>
</tr>
<tr>
<td></td>
<td>Appo</td>
<td>24.20</td>
<td>18.18</td>
<td>15.07</td>
<td>11.46</td>
<td>12.90</td>
<td>9.77</td>
</tr>
<tr>
<td></td>
<td>Coord</td>
<td>33.20</td>
<td>26.31</td>
<td>27.81</td>
<td>21.82</td>
<td>17.36</td>
<td>15.79</td>
</tr>
<tr>
<td></td>
<td>Obj</td>
<td>37.96</td>
<td>28.65</td>
<td>32.82</td>
<td>22.43</td>
<td>26.96</td>
<td>19.97</td>
</tr>
<tr>
<td></td>
<td>Prep</td>
<td>29.16</td>
<td>23.50</td>
<td>21.34</td>
<td>14.22</td>
<td>13.18</td>
<td>8.60</td>
</tr>
<tr>
<td></td>
<td>Subj</td>
<td>32.45</td>
<td>24.60</td>
<td>22.09</td>
<td>18.15</td>
<td>13.51</td>
<td>9.77</td>
</tr>
</tbody>
</table>

Table 3.18: Distribution of semantic relations over the top-k candidates for the several syntactic relations

We already noted that the coordination relation is a special relation in that a coordination such as Jip and Janneke does not establish a direct relation between Jip on the one hand and Janneke on the other hand. We have however established that direct link in our data.

There is one other phenomena that needs to be discussed. In van der Plas and Bouma (2005a) we explained that a single coordination consisting of many conjuncts gives rise to a large number of dependency triples (i.e. the coordina-
tion beer, wine, cheese, and nuts leads to three dependency triples per word, which is 12 in total). Especially for coordinations involving rare nouns, this has a negative effect. The example we gave was a listing of nicknames lovers use for each other:

Bobbelig Beertje, IJsbeertje, Koalapuppy, Hartebeer, Baloba Beer, Gere beer, Bolbuikmannie, Molletje, Knagertje, Lief Draakje, Hummeltje, Zeeuwse Poeperd, Egeltje, Bulletje, Tiiger, Woeste Wolf, Springende Spetter, Aap van me, Nunnepun, Trekkie, Bikkel en Nachtegaaltje

This generates 20 triples per name occurring in this coordination alone, although many of these occur nowhere else in the corpus. As a consequence, the results for a noun such as aap ‘monkey’ are highly polluted.

To remedy this problem we have tried to normalize coordination data from long lists. We gained better performances for the coordination data, when dividing the frequencies by \( n(n - 1) \). However, the effect was negative when used in combination with the other syntactic relations. The reason for this is probably the fact that many of the normalized co-occurrence types as a result receive a value that is below the threshold set. Hence, these co-occurrences are not taken into account. The evaluation that resulted in improved scores was based on the evaluation of 119 headwords out of 1000 only, whereas the baseline (not using normalization) was based on 541 words. The normalization resulted in too many words for which there was no data. We decided therefore not to use normalization for the coordination relation data.

The apposition relation performs badly on all semantic relations. It is a very typical relation, that links named entities and their category: president Clinton, prince Claus, the province Limburg. The categories named entities are related to are often functions people have or categories countries belong to. When we take a closer look at the nearest neighbours resulting from the apposition relation, we see that it does well for these function nouns but badly for words that are not often the category a named entity belongs to. In Table 3.19 we see an example of this effect.

<table>
<thead>
<tr>
<th>Test Word</th>
<th>k=1</th>
<th>k=2</th>
<th>k=3</th>
</tr>
</thead>
<tbody>
<tr>
<td>huwelijk</td>
<td>Nederland-Indonesie</td>
<td>Emanuel Muris</td>
<td>tangonummer</td>
</tr>
<tr>
<td>‘marriage’</td>
<td>‘The Netherlands-Indonesia’</td>
<td>,</td>
<td>‘tango number’</td>
</tr>
<tr>
<td>scheidsrechter</td>
<td>arbiter</td>
<td>voetbalscheidsrechter</td>
<td>top-arbiter</td>
</tr>
<tr>
<td>‘referee’</td>
<td>‘umpire’</td>
<td>‘football referee’</td>
<td>‘top referee’</td>
</tr>
</tbody>
</table>

Table 3.19: Examples of nearest neighbours for the apposition relation
3.5. Results

<table>
<thead>
<tr>
<th>Syntactic Relation</th>
<th>HF cov.</th>
<th>HF trace.</th>
<th>MF cov.</th>
<th>MF trace.</th>
<th>LF cov.</th>
<th>LF trace.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adj</td>
<td>100.0</td>
<td>85.4</td>
<td>99.8</td>
<td>53.2</td>
<td>98.2</td>
<td>24.3</td>
</tr>
<tr>
<td>Appo</td>
<td>100.0</td>
<td>47.1</td>
<td>93.7</td>
<td>23.4</td>
<td>55.1</td>
<td>22.5</td>
</tr>
<tr>
<td>Coord</td>
<td>98.1</td>
<td>75.2</td>
<td>87.9</td>
<td>53.6</td>
<td>66.1</td>
<td>43.6</td>
</tr>
<tr>
<td>Obj</td>
<td>100.0</td>
<td>90.1</td>
<td>100.0</td>
<td>57.9</td>
<td>99.1</td>
<td>32.2</td>
</tr>
<tr>
<td>Prep</td>
<td>100.0</td>
<td>89.5</td>
<td>99.7</td>
<td>49.3</td>
<td>88.3</td>
<td>33.5</td>
</tr>
<tr>
<td>Subj</td>
<td>100.0</td>
<td>88.5</td>
<td>100.0</td>
<td>51.6</td>
<td>99.6</td>
<td>26.9</td>
</tr>
<tr>
<td>All</td>
<td>100.0</td>
<td>88.7</td>
<td>100.0</td>
<td>65.3</td>
<td>99.9</td>
<td>32.8</td>
</tr>
</tbody>
</table>

Table 3.20: Coverage and traceability for various syntactic relations

Also, we must add a note to the figures given in Table 3.16 for the apposition relation. The apposition relation links named entities and their category. Only half of the data of the apposition relation is in fact used in the present evaluation, that includes only nouns and no proper names. We can see in Table 3.20 that the traceability and coverage scores for the apposition relation are very low. This means that many of the words in the low-frequency test set are not found in the data and many of the nearest neighbours given by the system are not found in EWN.

Combining all relations gives the best results. This is in line with Lin (1998a) and Padó and Lapata (2007), where the authors show that using multiple syntactic relations instead of only subject and object relations is beneficial for the scores. As we have seen in section 3.2.3, a lot of related work has been done on a limited number of syntactic relations (Hindle, 1990; Pereira et al., 1993; Dagan et al., 1999; Lee, 1999; Weeds and Weir, 2005).

The difference between using all relations or only the best one is larger when the frequency of the words in the test set is lower. In other words, the difference between the adjective relation and the combination of all relations (all) is largest for the low-frequency test set and smallest for the high-frequency test set. That is to be expected, since for low-frequency test words the data sparseness introduced by using just one relation is of more importance than for high-frequency words.

Should the scores not have convinced the reader yet that the combination of syntactic relations is a good idea, we would like to point the reader again to Table 3.20. We see that the coverage of the system is never as good for the individual syntactic relations as for the combination of all relations. It is clear that only the subject relation comes close to the combination of all relations with regard to coverage. However, the performance of this relation is not very good. The adjective relation that shows the best scores overall does not reach the coverage of the combination of all syntactic relations. Of the syntactic relations
that have a high coverage, none reaches the traceability reached by combining all relations. These results are based on a corpus of 500 million words. In case of a smaller corpus the difference between the coverage of the individual dependency relations and the combination of all dependency relations will be even larger.

3.5.7 Comparison to our previous work

There are a number of differences between the experiments in Van der Plas and Bouma (2005a) and the current methodology.

Firstly, the corpus has changed. In Van der Plas and Bouma (2005a) we used a smaller corpus: 80 million instead of 500 million words of newspaper text. The corpus has been parsed by a different version of Alpino, the dependency parser. Although in general the parses are better in the current version, one difficulty that was added by using the current version is the handling of compounds. The current version of Alpino does compound splitting. A compound such as *hondenhok* ‘dog house’ is split up to the lemma *hond*+*hok*. However, in our gold-standard *hond*+*hok* is not found. We had to translate all split-up compound lemmas back to the normal compound lemmas as they are found in EWN. It was not possible to do this conversion for all compound lemmas. This resulted in a small part (3.4%) of the compounds to be left in the data as split-up compounds. These compounds were therefore not found in EWN and no score could be calculated for them during evaluation.

Running the three test sets on the newly parsed 80 million word corpus with identical cell and row frequency cutoffs results in comparable maximal scores as the ones reported in Table 3.10 for the corpus of 80 million words. We can conclude from this that the effect of differences due to parsing is minimal.

Secondly, a difference that is of greater importance is the cell frequency and row frequency cutoffs we set. With the larger corpus it was no longer feasible to include hapaxes, i.e. words that occur only once in our data. Also, after careful testing the row frequency was set to 2 instead of 10. When running the old test set on the old data with with cell frequency cutoff 2 (instead of 1), and row frequency cutoff 2 (instead of 10), we get considerable improvements. The positive effect is mainly due to the lowering of the row frequency cutoff.

Thirdly and most importantly, the test set has changed. In Van der Plas and Bouma (2005a) we used a test set of 1000 random words from EWN with a frequency of more than 10 according to frequency information in EWN. We have now used more reliable frequency information, i.e. counts from the 80 million-word corpus (CLEF). And we have split up the test set in 1000 high-frequency nouns, 1000 middle-frequency nouns, and 1000 low-frequency nouns.
3.6. Conclusions

Only 19% of the nouns in the previous test set has a frequency equal or higher than our middle frequency test set. More than 55% is less frequent than the current low-frequency test set. It is therefore not surprising that the current scores are higher than the ones reported in Van der Plas and Bouma (2005a).

We can conclude that the differences between the current scores and the scores reported in Van der Plas and Bouma (2005a) are mainly due to the test sets used, the increased corpus size, and the row frequency cutoff set.

With respect to the ranking of the combinations of measures and weights we would like to note the following. In Van der Plas and Bouma (2005a) Dice †+MI was the best performing measure. When running the three test sets with the current cutoffs on the 80 million corpus, we noticed that for the low frequency test set the same phenomenon occurred: Dice †+MI scored higher than Cosine+MI. The ranking of the combinations is clearly dependent on the size of the corpus and the frequency of the words in the test set.

3.6 Conclusions

In this chapter we have tried to provide information about the nature and quality of the nearest neighbours found by the syntactic methods. We have evaluated the nearest neighbours on the gold standard EWN with a measure that combines the several semantic relations in different degrees in one score. We have also determined the proportion of synonyms, hyper- and (co-)hyponyms to get an idea of the decomposition of the score in the several semantic relations.

The most important outcome is perhaps that the syntax-based method finds many semantically related words, among which synonyms, hypernyms, hyponyms and (co)hyponyms. The proportion of synonyms is on average 21% for the high frequency test set at the first ranks. The number of co-hyponyms is about twice as high.

The syntax-based method gives better results than the proximity-based method, that will be further discussed in chapter 5. Syntactic information is helpful for this task. However, both methods outperform the baseline.

The nearest neighbours of high-frequency nouns are of a better quality than the middle-frequency neighbours, and these in turn are of a better quality than the neighbours of low-frequency nouns. Also, a larger corpus results in better scores. The differences between the two corpora of different size is largest for the low-frequency test set. These phenomena can all be explained by data sparseness, a problem that is more severe for smaller corpora and for words in lower frequency bands.

Another important outcome is that combining all relations gives the best results. This is in line with Lin (1998a) and Padó and Lapata (2007), where the
authors show that using multiple syntactic relations instead of only subject and object relations is beneficial for the scores. As a positive side-effect the number of words that are covered by the system and EWN is higher when all syntactic relations are taken into account as reflected by coverage and traceability scores.

The performance of the several syntactic relations is partly explainable by data sparseness. Syntactic relations that are common and result in many co-occurrence tokens and types give the best results. However, the nature of the syntactic relation also plays an important role.

The adjective and object relation perform best and are relatively common. The subject relation is the most common relation, but it does less well than the two best performing relations. The subject relation is limited to active things. It is typically not very good at describing less animate, less active things, such as *verminking* ‘mutilation’.

The apposition relation is very limited with regard to the type of nouns it has data for, because it relates named entities with their function/hypernym. It does well on functions people have, such as *scheidsrechter* ‘referee’, but it does not well on other nouns, such as *huwelijk* ‘marriage’.

The coordination relation is interesting because of the fact that it performs much better with regard to co-hyponymy than with respect to other semantic relations, such as synonymy. Because the coordination relation is one that typically relates co-hyponyms in text this is expected.

Furthermore, from our experiment we can conclude that using no cutoffs (except the exclusion of hapaxes) gives the best results. Also, Cosine in combination with Mutual Information is the best combination of measures and weights we tried, followed by Dice†+t-test and Dice†+MI. Weighting is beneficial. The settings without weighting, where the raw frequencies are used in the calculations perform worst. The combination of Cosine and t-test results in many infrequent, unrelated words. The relative performance of the combinations of weights and measures is dependent on the corpus size and the frequency of the test words. Comparing our results to previous work is difficult due to differences in methodology and evaluation framework.

The usefulness of the found neighbours will be tested on a real application, question answering, in Chapter 6.