Syntactic Contexts for Finding Semantically Related Words

Lonneke van der Plas and Gosse Bouma

Humanities Computing, University of Groningen

Abstract

Finding semantically related words is a first step in the direction of automatic ontology building. Guided by the view that similar words occur in similar contexts, we looked at the syntactic context of words to measure their semantic similarity. Words that occur in a direct object relation with the verb *drink*, for instance, have something in common (*liquidity*, ...). Co-occurrence data for common nouns and proper names, for several syntactic relations, was collected from an automatically parsed corpus of 78 million words of newspaper text. We used several vector-based methods to compute the distributional similarity between words. Using Dutch EuroWordNet as evaluation standard, we investigated which vector-based method and which combination of syntactic relations is the strongest predictor of semantic similarity.

1 Introduction

Ontologies comprise semantically related words structured in IS-A relations. An IS-A relation or *hyponym-hypernym* relation holds between a word and a more general word in the same semantic class, e.g. *cat* IS-A *animal*. This type of knowledge is useful for an application such as Question Answering (QA). In many cases, QA systems classify questions as asking for a particular type of *Named Entity*. For instance, given the question *What actor is used as Jar Jar Binks's voice?*, question classification tells the system to look for strings that contain the name of a person. This requires ontological knowledge in which it is stated that an *actor* IS-A *person*. An IS-A hierarchy can also be useful for answering more general WH-questions such as: *What is the profession of Renzo Piano?* In the document collection the following sentence might be found: *Renzo Piano is an architect and an Italian*. Knowing that *Italian* is not a profession but *architect*, helps in deciding which answer to select.

We want to incorporate ontological information in a Dutch QA system. Lexical knowledge bases such as Dutch EuroWordnet (Vossen [1998]) can be used to provide this type of information. However, its coverage is not exhaustive, and thus, we are interested in techniques to automatically extend it. One method to extend an existing IS-A hierarchy is to find words that are semantically related to words already present in the hierarchy. That is, given an ontology which contains an IS-A relation between *banana* and *fruit*, we want to find words related to *banana* (e.g. *orange, strawberry, pineapple, pear, apple, ...*) and include IS-A relations between these words and *fruit* as well.

To find semantically related words, we use a corpus-based method which finds distributionally similar words. Grefenstette [1994] refers to such words as words which have a *second-order affinity*: Words that co-occur frequently (*sinaasappel* (*orange*) and *uitgeperst* (*squeezed*)) have a first-order affinity, words that share the same first-order affinities have a second order affinity, for example, both *sinaasappel* and *citroen* (*lemon*) can be modified by *uitgeperst*.

	hebben	ziekenhuis	zeggen	vrouwelijk	besmettelijk
	(*have*)	(*hospital*)	(*say*)	(*female*)	(*contagious*)
	obj	coord	subj	adj	adj
tandarts	4	4	10	4	0
arts	17	24	148	26	0
ziekte	114	0	0	0	99
telefoon	81	0	0	0	0

Table 1: Sample of the syntactic co-occurrence vectors for various nouns

In this paper, we report on an experiment aimed at finding semantically related words. We briefly discuss previous work on finding distributionally similar words using large corpora. Next, we describe how we collected data for Dutch. Finally, we present the results of an evaluation against Dutch EuroWordNet. We investigated which vector-based methods and which (combinations of) grammatical relations are the strongest predictors of semantic similarity.

2 Related work

2.1 Using Syntactic Context

Words that are distributionally similar are words that share a large number of contexts. There are basically two methods for defining contexts. One can define the context of a word as the n words surrounding it (n-grams, bag-of-words). Another approach is one in which the context of a word is determined by grammatical dependency relations. In this case, the words with which the target word is in a dependency relation form the context of that word.

In both cases, computing distributional similarity requires that a corpus is searched for occurrences of a word, and all relevant words or words plus grammatical relations are counted. The result is a vector. A part of the vectors we collected (using syntactic contexts) for the words *tandarts (dentist), arts (doctor), ziekte (disease)* and *telefoon (telephone)* is given in table 1. Each row represents the vector for the given word. Each column is headed by a word and the grammatical relation it has with the corresponding row word. We can see that *tandarts* appeared four times as the object of the verb *hebben* (*have*) and that *ziekte* never appeared in coordination with *ziekenhuis* (*hospital*).

Kilgarriff and Yallop [2000] use the terms *loose* and *tight* to refer to the different types of semantic similarity that are captured by methods using surrounding words only and methods using syntactic information. The semantic relationship between words generated by approaches which use context only seems to be of a *loose*, associative kind. These methods put words together according to subject fields. For example, the word *doctor* and the word *disease* are linked in an associative way. Methods using syntactic information have the tendency to generate *tighter* thesauri, putting words together that are in the same semantic class, i.e. words for 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*. The tighter thesauri generated by methods that take syntactic information into account seem to be more appropriate for ontology building. Therefore, we concentrate on this method.

Most research has been done using a limited number of syntactic relations (Lee [1999], Weeds [2003]). However, Lin [1998a] shows that a system which uses a range of grammatical relations outperforms Hindle's (1990) results that were based on using information from just the subject and object relation. We use several syntactic relations.

2.2 Measures and feature weights

Vector-based methods for finding distributionally similar words, need a way to compare the vectors for any two words, and 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. In our experiments, we have only used Cosine and a variant of Dice. These measures are explained in section 3.2. We chose these methods, as they performed best in a large-scale evaluation experiment reported on in Curran and Moens [2002].

The results of vector-based methods can be further improved if we take into account the fact that not all words, or not all combinations of a word and grammatical 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 *uitpersen* (*squeeze*) on the other hand occurs much less frequently, and only with a restricted set of nouns as 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 object of *uitpersen*. To account for this intuition, the frequency of occurrence in a vector such as in 1 can be multiplied by a feature weight (each cell in the vector is seen as a feature). The weight is an indication of the amount of information carried by that particular combination of a noun, the grammatical relation, and the word heading the grammatical relation. Various techniques for computing feature weights exist. Curran and Moens [2002] perform experiments using (Pointwise) Mutual Information (MI), the t-test, χ^2, and several other techniques. MI and t-test, the best performing weighting methods according to Curran and Moens, are introduced in section 3.2.

Applying MI to the matrix in 1, results in the matrix in table 2, where frequency counts have been replaced by MI scores. Note that the values for cells involving the verb *hebben* no longer exceed those of the other cells, and that the value for *besmettelijke ziekte* (*contagious disease*) now out-ranks all other values.

2.3 Evaluation

One method for evaluating the performance of a corpus-based method for finding semantically similar words, is to compare the similarity scores assigned by the system

	hebben (*have*) obj	ziekenhuis (*hospital*) coord	zeggen (*say*) subj	vrouwelijk (*female*) adj	besmettelijk (*contagious*) adj
tandarts	0	4.179	0.155	4.158	0
arts	0	3.938	0.540	3.386	0
ziekte	0.550	0	0	0	7.491
telefoon	0.547	0	0	0	0

Table 2: Sample of the MI-weighted syntactic co-occurrence vectors for various nouns

to a pair of words with human judgements. In this form of evaluation, a fixed set of word pairs is used, which are assigned similarity scores by both human judges and the system. If the correlation between the two is high, the system captures human notions of semantic similarity. This evaluation technique has been used for English, using a set of word pairs and human judgements collected originally by Rubenstein and Goodenough [1965]. Resnik [1995] used it to evaluate various measures for computing semantic similarity in WordNet (Fellbaum [1998]) and Weeds [2003] uses it for evaluating distributional measures. Selecting suitable word pairs for comparison, and collecting human judgements for them, is difficult. Furthermore, as Weeds [2003] points out, assigning scores to word pairs is hard for human judges, and human judges tend to differ strongly in the scores they assign to a given word pair.

An alternative evaluation method measures how well similarity scores assigned by the system correlate with similarity in a given lexical resource. Curran and Moens [2002], for instance, computed for each word its nearest neighbours according to a number of similarity measures. Next, they checked whether these pairs were listed as synonyms in one of three different thesauri (the MacQuarie (Bernard [1990]), Moby (Ward [1996]) and Roget (Roget [1911])). A somewhat similar approach is to evaluate nearest neighbours against a lexical resource such as WordNet. A number of measures exist to compute semantic similarity of words in WordNet (Resnik [1995]). A system performs well if the nearest neighbours it finds for a given word are also assigned a high similarity score according to the WordNet measure. An advantage of this evaluation technique is that not only synonyms are taken into account, but also words closely related to the target word. In our experiments, we have used Dutch EuroWordNet (Vossen [1998]) as lexical resource and used the measure of Wu and Palmer [1994]. This method for calculating WordNet similarity is one that correlates well with human judgements according to Lin [1998b] and it can be implemented without the need for frequency information which is difficult to acquire.

3 Experiment

In this section, we describe the data collection process, and the similarity measures and weights we used.

subject-verb	de *kat eet*.
verb-object	ik *voer* de *kat*.
adjective-noun	de *langharige kat* loopt.
coordination	*Bassie en Adriaan* spelen.
apposition	de *clown Bassie* lacht.
prepositional complement	ik *begin met* mijn *werk*.

Table 3: Types of dependency relations extracted

grammatical relation	tuples	types
subject	5.639.140	2.122.107
adjective	3.262.403	1.040.785
object	2642.356	993.913
coordination	965.296	2.465.098
prepositional complement	770.631	389.139
apposition	526.337	602.970

Table 4: Number of tuples and non-identical dependency triples (types) extracted per dependency relation.

3.1 Data collection

As our data we used 78 million words of Dutch newspaper text (Algemeen Dagblad and NRC Handelsblad 1994/1995), that were parsed automatically using the Alpino parser (van der Beek et al. [2002], Malouf and van Noord [2004]). The result of parsing a sentence is a dependency graph according to the guidelines of the Corpus of Spoken Dutch (Moortgat et al. [2000]).

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). Examples are given in table 3. The number of tuples and the number of non-identical ⟨`Noun,Relation,OtherWord`⟩ triples (types) found are given in table 4. Note that a single coordination can give rise to various dependency triples, as from a single coordination like *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* we extract both ⟨*premier, hd_app, Kok*⟩ and ⟨*Kok, app, premier*⟩.

For each noun that was found at least 10 times in a given dependency relation (or combination of dependency relations), we built a vector. Using this cutoff of 10 the matrix built using the the subject relation contained 30.327 nouns, whereas the matrix built using apposition only contained 5.150 nouns. Combining the data for all grammatical relations into a single matrix means that vectors are present for 83.479 nouns.

3.2 Similarity measures used

Methods for computing distributional similarity consist of a measure for assigning weights to the dependency triples present in the matrix, and a measure for computing similarity between two (weighted) word vectors.

As weights we used identity, MI and the t-test. Identity was used as a baseline, and simply assigns every dependency triple a weight of 1 (i.e. every count in the matrix is multiplied by 1).

(Pointwise) Mutual Information (Church and Hanks [1989]) 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 and a feature f (e.g. the word *ziekte (disease)* and the feature *besmettelijk_adj (contagious_adj)*) is computed as follows:

$$I(W,f) = log \frac{P(W,f)}{P(W)P(f)}$$

Here, $P(W,f)$ is the probability of seeing *besmettelijke ziekte* (in a modifier-head relation) in the corpus, and $P(W)P(f)$ is the product of the probability of seeing *besmettelijke* and the probability of seeing *ziekte*.

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 variance of the data. The t-test takes into account the number of co-occurrences of the bi-gram (e.g., a word W and a feature f in a grammatical relation) 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:[1]

$$t = \frac{P(W,f) - P(W)P(f)}{\sqrt{P(W)P(f)}}$$

We used two different similarity measures to calculate the similarity between two word vectors: *Cosine* and *Dice*† (Curran and Moens [2002]). We describe the functions using an extension of the asterisk notation of Lin [1998b]. An asterisk indicates a set ranging over all existing values of that variable. A subscripted asterisk indicates that the variables are bound together.

Cosine is a geometrical measure. It returns the cosine of the angle between the vectors of the words and is calculated using the dot product of the vectors:

$$\textit{Cosine} = \frac{\sum_f \textit{weight}(W1, *_f) \times \textit{weight}(W2, *_f)}{\sqrt{\sum \textit{weight}(W1, *)^2 \times \sum \textit{weight}(W2, *)^2}}$$

[1] Note, however, that this formulation of the t-test differs from that in Manning and Schütze [1999], in spite of the fact that Curran and Moens explicitly refer to Manning and Schütze as their source.

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 either identity, MI or t-test.

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

$$Dice(A,B) = \frac{2. \mid A \cap B \mid}{\mid A \mid + \mid B \mid}$$

,where A stands for the set of features of word 1 and B for the set of features of word 2.

Curran and Moens [2002] propose a variant of Dice, which they call *Dice*†. It is defined as:

$$Dice\dagger = \frac{2 \sum_f min(weight(W1, *_f), weight(W2, *_f))}{\sum_f weight(W1, *_f) + weight(W2, *_f)}$$

Whereas *Dice* does not take feature weights into account, *Dice*† does. For each feature two words share, the minimum is taken. If $W1$ occurred 15 times with feature f and $W2$ occurred 10 times with f, and if identity is used for *weight*, it selects 10 as the minimum.

4 Evaluation

Given a matrix consisting of word vectors for nouns, and a similarity method (combination of a weight and similarity measure),the similarity between any pair of nouns can be computed (provided that they are found in the data).[2] On the basis of this, the nouns that are most similar to a given noun can be produced. In this section, we present an evaluation of the system for finding semantically similar words. We evaluated the system against data extracted from EuroWordNet, using various similarity measures and weights, and using various (combinations of) dependency relations.

4.1 Evaluation Framework

The Dutch version of the multilingual resource EuroWordNet (EWN) (Vossen [1998]) was used for evaluation. We randomly selected 1000 target words from Dutch EWN with a frequency of more than 10, according to the frequency information present in Dutch EWN. For each word we collected its 100 most similar words (nearest neighbours) according to our system, and for each pair of words (target word + one of the most similar words) we calculated the semantic similarity according to Dutch EWN. A system scores well if the nearest neighbours found by the system also have a high semantic similarity according to EWN.

EWN is organised in the same way as the well-known English WordNet Fellbaum [1998], that is word senses with the same meaning form *synsets*, and IS-A relations

[2] A demo of the system, using the combination of all grammatical relations, and MI+*Dice*† as similarity method, can be found on `www.let.rug.nl/~gosse/Sets`

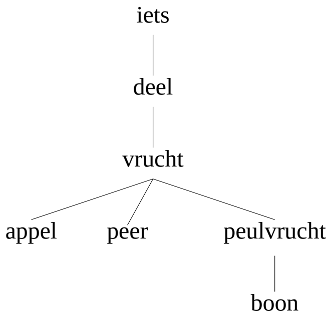

Figure 1: Fragment of the IS-A hierarchy in Dutch EuroWordNet.

between synsets are defined. Together, the IS-A relations form a tree, as illustrated in figure 1. The tree shows that *appel* (*apple*) IS-A *vrucht* (*fruit*), which IS-A *deel* (*part*), which IS-A *iets* (*something*). A *boon* (*bean*) IS-A *peulvrucht* (*seed pod*), which IS-A *vrucht*.

For computing the WordNet similarity between a pair of words we used the Wu/Palmer [1994] measure. It correlates well with human judgements and can be computed without using frequency infomation. The Wu/Palmer measure for computing the semantic similarity between two words W1 and W2 in a wordnet, whose most-specific common ancestor is W3, is defined as follows:

$$\mathit{Sim} = \frac{2(D3)}{D1 + D2 + 2(D3)}$$

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. The similarity between *appel* and *peer* according to the example in 1 would be $4/6 = 0.66$, whereas the similarity between *appel* and *boon* would be $4/7 = 0.57$.

Below, we report EWN similarity for the 1, 5, 10, 20, 50, and 100 most similar words of a given target word. If a word is ambiguous according to EWN (i.e. is a member of several synsets), the highest similarity score is used. The EWN similarity of a set of word pairs is defined as the average of the similarity between the pairs.

4.2 Results

In a first experiment, we compared the performance of the various combinations of weight measures (identity, MI, and t-test) and the measures for computing the distance between word vectors (Cosine and Dice†). The results are given in table 5. All combinations significantly outperform the random baseline (i.e. the score obtained by picking 100 random words as nearest neighbours of a given target word), which, for EWN, is 0.26. Note also that the maximal score is not 1.00, but significantly lower, as words do not have 100 synonyms (which would give, the hypothetical, maximal score

Measure	EWN Similarity at					
+Weight	k=1	k=5	k=10	k=20	k=50	k=100
Dice† +MI	**0.560**	**0.499**	**0.477**	**0.458**	**0.433**	**0.415**
Cosine+MI	0.544	0.489	0.468	0.453	0.428	0.410
Dice† +t-test	0.518	0.482	0.461	0.449	0.425	0.408
Dice† +identity	0.492	0.452	0.430	0.415	0.394	0.375
Cosine+identity	0.494	0.434	0.412	0.396	0.376	0.362
Cosine+t-test	0.472	0.425	0.410	0.402	0.388	0.376

Table 5: Average EWN similarity at k candidates for different similarity measures and weights, using data from the object relation

Dependency	EWN Similarity at					
Relation	k=1	k=5	k=10	k=20	k=50	k=100
Object	**0.560**	**0.499**	**0.477**	**0.458**	**0.433**	**0.415**
Adjective	0.556	0.492	0.463	0.444	0.414	0.395
Coordination	0.495	0.488	0.468	0.453	0.432	0.414
Apposition	0.508	0.465	0.449	0.437	0.418	0.400
Prep. comp.	0.482	0.443	0.431	0.415	0.393	0.380
Subject	0.451	0.426	0.414	0.396	0.380	0.369

Table 6: Average EWN similarity at k candidates for different dependency relations based on Dice† + MI

of 1.00). *Dice*† in combination with MI gives the best results at all points of evaluation, followed by *Cosine* in combination with MI. It is clear that MI makes an important contribution. Also, the difference in performance between *Cosine* and *Dice*† is much bigger when no weight is used (identity) and biggest when t-test is used. t-test and *Cosine* do not work well together, t-test and *Dice*† are a better combination. As *Dice*† *+MI* performs best, this combination was used in the other experiments.

In table 6, the performance of the data collected using various dependency relations is compared. The object relation is best at finding semantically related words. Adjective and coordination are also relatively good, except for the fact that the score for coordination at $k = 1$ is quite a bit lower than for the other two relations. In spite of the fact that using the subject relation most data was collected, this is not a good relation for finding semantically similar words.

In table 7, we give results for various combinations of dependency relations. We started by combining the best performing relations, and then added the remaining relations. In general, it seems to be true that combining data from various relations improves results. Removing the subject relation data from *all*, for instance, decreases performance, in spite of the fact that using only the subject relation leads to poor

	EWN Similarity at					
Combination	k=1	k=5	k=10	k=20	k=50	k=100
Obj	0.560	0.499	0.477	0.458	0.433	0.415
Obj+adj	0.584	0.529	0.499	0.473	0.442	0.420
Obj+adj+coord	0.589	0.533	0.512	0.487	0.459	0.436
Obj+adj+coord+pc	0.585	0.532	0.512	0.491	0.460	0.437
All	**0.603**	**0.542**	0.519	0.494	0.464	0.442
All-appo	0.596	0.541	**0.520**	**0.497**	**0.466**	**0.444**
All-subj	0.588	0.530	0.509	0.488	0.458	0.435

Table 7: Average EWN similarity at k candidates when combining dependency relations based on Dice† + MI

results. The only exception to this rule might be the apposition data. Removing these from *all*, means that slightly better scores are obtained for $k \geq 20$.

4.3 Discussion of results

The fact that MI does so well is at first sight surprising and conflicts with results from earlier research by Curran and Moens [2002]. They show that t-test is the best performing method for setting feature weights. MI in general is known to overemphasise low frequency events. The reason for the fact that MI performs rather well in our experiment could be explained by the cutoffs we set. In section 3.1 we explained that we discarded words that occurred less than 10 times in the relevant configuration.

In accordance with the experiments done by Curran and Moens [2002] we show that *Dice†* outperforms *Cosine*.

From table 6 we can see that there is a difference in performance of the different dependency relations and in table 7 we see that the apposition relation hurts the performance at k =10. However, the evaluation framework is not always a fair one for all relations. Not all similar words found by our system are also found in Dutch EWN. Approximately 60% of the most similar words returned by our system were not found in Dutch EWN. Word pairs found by the system but absent in EWN were discarded during evaluation. This is especially harmful for the apposition relation. The apposition relation always holds between a noun and a proper name. Proper names are not very well presented in EWN, and as a consequence they do not play a role in the evaluation. Therefore, we suspect that the observed effect may well be due to our evaluation method. Other evaluation methods (i.e. in particular a task-based evaluation of using ontological information in QA [3]) may well show that the inclusion of information from appositions has a positive effect. This does suggest that our corpus-based approach indeed finds many words that are absent from the only lexical resource which systematically provides IS-A relations for Dutch, and thus, that automatic or semi-automatic extension of Dutch EWN might be promising.

[3] see van der Plas and Bouma [2005])

dependency relation	Coverage (%)
apposition	11.2
prepositional complement	29.8
object	47.8
adjective	50.3
coordination	56.0
subject	57.9
object+adjective	62.3
object+adjective+coordination	72.9
all-subject-apposition	74.5
all-apposition	78.8
all	78.9

Table 8: Percentage of target words from EWN found in the data set for various (combinations of) dependency relations.

In general we show that combining grammatical relations leads to better results (table 7). In table 8 the percentage of target words that are found in the data collected for different (combinations of) dependency relations (and using a cutoff of 10 occurrences) is given. The fact that coverage increases when combining dependency relations provides further motivation for using systems that combine information from various dependency relations.

The subject relation produces a lot of tuples, but performs surprisingly poorly. Inspection of some sample output, suggests that this may be due to the fact that nouns which denote passive things (i.e. *strawberries* or *tables*) are typically not very well represented in the subject data. Nouns which are clearly agentive, such as *president*, performed much better.

A final note concerns our treatment of coordination. A single coordination consisting of many conjuncts, gives rise to a large number of dependency triples (i.e. the coordination *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. A case in point is the example below, which is a listing of nicknames lovers use for each other:

> Bobbelig Beertje, IJsbeertje, Koalapuppy, Hartebeer, Baloeba Beer, Gerebeer, Bolbuikmannie, Molletje, Knagertje, Lief Draakje, Hummeltje, Zeeuwse Poeperd, Egeltje, Bulletje, Tijger, Woeste Wolf, Springende Spetter, Aap van me, Nunnepun, Trekkie, Bikkel en Nachtegaaltje

This generates 20 triples per name occurring in this coordination alone. As a consequence, the results for a noun such as *aap* (*monkey*) are highly polluted.

5 Conclusion

From our experiment we can conclude that *Dice*† in combination with Mutual Information is the best technique for finding semantically related words. This result is in contrast with results in Curran and Moens [2002].

Another conclusion we can draw is that the object relation is the best performing relation for this task, followed by the adjective relation. The results from coordination can probably be improved, if we adopt a more principled approach to dealing with long coordinations.

However, although some dependency relations perform rather poorly, combining all dependency relations improves the performance of our system. The number of words covered is higher and in almost all cases the average EWN similarity is higher.

In the near future we would like to combine our method for finding similar words with methods for acquiring IS-A relations automatically. Promising results on learning the latter on the basis of data parsed by Alpino are reported in IJzereef [2004]. In addition, we would like to investigate methods for expanding Dutch EWN (semi-)automatically. Finally, we would like to apply the knowledge gathered in this way for QA-tasks, such as question classification, and answering of general WH-questions.

6 Acknowledgements

This research was carried out in the project *Question Answering using Dependency Relations*, which is part of the research program for *Interactive Multimedia Information Extraction*, IMIX, financed by NWO, the Dutch Organisation for Scientific Research.

References

J.R.L. Bernard. The Macquairie encyclopedic thesaurus. The Macquairie Library, Sydney, Australia, 1990.

K.W. Church and P. Hanks. Word association norms, mutual information and lexicography. *Proceedings of the 27th annual conference of the Association of Computational Linguistics*, pages 76–82, 1989.

J.R. Curran and M. Moens. Improvements in automatic thesaurus extraction. In *Proceedings of the Workshop on Unsupervised Lexical Acquisition*, pages 59–67, 2002.

C. Fellbaum. Wordnet, an electronic lexical database. MIT Press, 1998.

G. Grefenstette. Corpus-derived first-, second-, and third-order word affinities. In *Proceedings of Euralex*, pages 279–290, 1994.

D. Hindle. Noun classification from predicate-argument structures. In *Proceedings of ACL-90*, pages 268–275, 1990.

L. IJzereef. Automatische extractie van hyponiemrelaties uit grote tekstcorpora, 2004. URL `www.let.rug.nl/alfa/scripties.html`. Masters thesis, Rijksuniversiteit Groningen.

A. Kilgarriff and C. Yallop. What's in a thesaurus? In *Proceedings of the Second Conference on Language Resource an Evaluation*, pages 1371–1379, 2000.

L. Lee. Measures of distributional similarity. In *37th Annual Meeting of the Association for Computational Linguistics*, pages 25–32, 1999.

Dekang Lin. Automatic retrieval and clustering of similar words. In *COLING-ACL*, pages 768–774, 1998a.

Dekang Lin. An information-theoretic definition of similarity. In *Proc. 15th International Conf. on Machine Learning*, pages 296–304. Morgan Kaufmann, San Francisco, CA, 1998b.

Robert Malouf and Gertjan van Noord. Wide coverage parsing with stochastic attribute value grammars. In *IJCNLP-04 Workshop Beyond Shallow Analyses - Formalisms and stati stical modeling for deep analyses*, Hainan, 2004.

Christopher D. Manning and Hinrich Schütze. *Foundations of Statistical Natural Language Processing*. The MIT Press, Cambridge, Massachusetts, 1999.

Michael Moortgat, Ineke Schuurman, and Ton van der Wouden. CGN syntactische annotatie, 2000. Internal Project Report Corpus Gesproken Nederlands, see http://lands. let.kun.nl/cgn.

P. Resnik. Selection and information. Unpublished doctoral thesis, University of Pennsylvania, 1993.

P. Resnik. Using information content to evaluate semantic similarity. In *Proceedings of the 14th international joint conference on artificial intelligence*, pages 448–453, 1995.

P. Roget. Thesaurus of English words and phrases, 1911.

H. Rubenstein and J.B. Goodenough. Contextual correlates of synonymy. *Communications of the ACM*, 1965.

Leonoor van der Beek, Gosse Bouma, and Gertjan van Noord. Een brede computationele grammatica voor het Nederlands. *Nederlandse Taalkunde*, 7(4):353–374, 2002.

Lonneke van der Plas and Gosse Bouma. Automatic acquisition of lexico-semantic knowledge for QA. Proceedings of Ontolex, 2005. to appear.

P. Vossen. Eurowordnet a multilingual database with lexical semantic networks, 1998. URL `citeseer.ist.psu.edu/vossen98eurowordnet.html`.

G. Ward. Moby thesaurus. Moby Project, 1996.

J. Weeds. *Measures and Applications of Lexical Distributional Similarity*. PhD thesis, University of Sussex, 2003.

Z. Wu and M. Palmer. Verb semantics and lexical selection. In *The 23rd Annual Meeting of the Association for Computational Linguistics*, pages 133–138, 1994.