Combining Syntactic Co-occurrences and Nearest Neighbours in Distributional Methods to Remedy Data Sparseness.

Lonneke van der Plas
Department of Linguistics
University of Geneva
Geneva, Switzerland

Abstract

The task of automatically acquiring semantically related words have led people to study distributional similarity. The distributional hypothesis states that words that are similar share similar contexts. In this paper we present a technique that aims at improving the performance of a syntax-based distributional method by augmenting the original input of the system (syntactic co-occurrences) with the output of the system (nearest neighbours). This technique is based on the idea of the transitivity of similarity.

1 Introduction

The approach described in this paper builds on the DISTRIBUTIONAL HYPOTHESIS, the idea that semantically related words are distributed similarly over contexts. 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.' In other words, you can grasp the meaning of a word by looking at its context.

Context can be defined in many ways. In this paper we look at the syntactic contexts a word is found in. For example, the verbs that are in a object 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 object relation with the verb *drink* have something in common: they are liquid. We will refer to words linked by a syntactic relation, such as *drink -OBJ-beer*, as SYNTACTIC CO-OCCURRENCES. Syntactic co-occurrences have often been used in work on lexical acquisition (Lin, 1998b; Dagan et al., 1999; Curran and Moens, 2002; Alfonseca and Manandhar, 2002).

Distributional methods for automatic acquisition of semantically related words suffer from data sparseness. They generally perform less well on low-frequency words (Weeds and Weir, 2005; van der Plas, 2008). This is a pity because the available resources for semantically related words usually cover the frequent words rather well. It is for the low-frequency words that automatic methods would be most welcome.

This paper tries to find a way to improve the performance on the words that are most wanted: the middle to very-low-frequency words. At the basis of the proposed technique lies the intuition that semantic similarity between concepts is transitive: if A is like B and B is like C → A is like C. As explained in the second paragraph of this section, the fact that both *milk* and *water* are found in object relation with the verb *to drink* tells us that they might be similar. However, even if we had never seen *lemonade* in the same syntactic contexts as *water*, we could still infer that *lemonade* and *water* are similar because we have found evidence that both *water* and *lemonade* are similar to *milk*.

In an ideal world we would be able to infer that *milk* and *water* are related from the syntactic co-occurrences alone, however, because of data sparseness we might not always encounter this evidence directly. We hope that nearest neighbours are able to account for the missing information. Nearest neighbours such as *milk* and *water*, and *water* and *lemonade* are the output of our system. We used the nearest neighbours (the output of our system) as input to our system that normally takes syntactic co-

Proceedings of the NAACL HLT Workshop on Unsupervised and Minimally Supervised Learning of Lexical Semantics, pages 45–53,
Boulder, Colorado, June 2009.

occurrences as input. Thus it uses the output of the system as input in a second round to smooth the syntactic co-occurrences.

Grefenstette (1994) discusses the difference between FIRST- AND SECOND-ORDER AFFINITIES. There exists a first-order affinity between words if they often appear in the same context, i.e., if they are often found in the vicinity of each other. Words that co-occur frequently such as *orange* and *squeezed* have a first-order affinity. There exists a second-order affinity between words if they share many first-order affinities. These words need not appear together themselves, but their contexts are similar. *Orange* and *lemon* appear often in similar contexts such as being the object of *squeezed*, or being modified by *juicy*.

In this paper we will use second-order affinities as input to the distributional system. We are thus computing THIRD-ORDER AFFINITIES.[1] There exists a third-order affinity between words, if they share many second-order affinities. If *pear* and *watermelon* are similar and *orange* and *watermelon* are similar, then *pear* and *orange* have a third-order affinity.

We will refer to traditional approaches that compute second-order affinities as second-order techniques. In this paper we will compare a second-order technique with a third-order technique, a technique that computes third-order affinities. In addition we use a combined technique that combines both second-order and third-order techniques.

2 Previous work

In Edmonds (1997) the term third-order is used to refer to a different concept. Firstly, we have to mention that the author is working in a proximity-based framework, that is, he is concerned with co-occurrences of words in text, not relations between words in syntactic dependencies. Secondly, the notion of higher-order co-occurrences refers to connectivity paths in networks, i.e. the network of relations between words co-occurring is augmented by connecting words that are connected by a path of length 2 (second-order co-occurrences) and paths of length 3 (third-order co-occurrences) and so on. In the above example *water* and *lemonade* would be connected by a second-order relation implied by the network in which *water* and *lemonade* both co-occur with for example *to pour*. A third-order relation would be implied between *lemonade* and *drink* if *drink* should co-occur with *water*. We define third-order affinity as an iterative process of calculating similarity. The output of the system is fed into the system again. There exists a third-order affinity between words if they share many nearest neighbours with another word, not if a word shares a context that in turn shares a context with the other word. The same perspective on higher-order co-occurrence, that of connectivity paths in networks, is taken in literature of computational modelling of the acquisition of word meaning (Lemaire and Denhire, 2006).

Although Biemann et al. (2004) work in the same proximity-based tradition as the previous authors their notion of third-order is closer to our definition. It is defined as an iterative process in which words are linked when their co-occurrence score trespasses a certain threshold. These nth-order co-occurrences are then used to construct an artificial corpus consisting of the co-occurrence sets retrieved from the original corpus.

Schütze and Walsh (2008) present a graph-theoretic model of lexical-syntactic representation in which higher-order syntactic relations, those that require some generalisation, are defined recursively. The problem they are trying to solve, lexical syntactic acquisition, is different form ours and so is the evaluation method: discriminating sentences that exhibit local coherence from those that do not. Again the method is proximity-based, but since the context are defined very locally (left and right neighbours) the results are likely to be more comparable to a syntax-based method than proximity-based methods that use larger contexts.

3 Limits of the transitivity of similarity

The validity of the third-order affinities is dependent on the transitivity of the similarity between concepts. Unfortunately, it is not always the case that the similarity between A and B and B and C implies the similarity between A and C.

[1] Grefenstette (1994) uses the term third-order affinities for a different concept, i.e. for the subgroupings that can be found in list of second-order nearest neighbours.

When two concepts are identical, the transitivity of similarity holds. If A=B AND B=C → A=C. Does the same reasoning hold for similarity of a lesser degree? For (near-)synonyms the transitivity holds and it is symmetric. If *felicity* is like *gladness*, and *gladness* is like *joy* → *felicity* is like *joy*. Also, the near-synonymy relation is symmetric. We can infer that *gladness* is like *felicity*.

Tversky and Gati (1978) give an example of co-hyponymy where transitivity does not hold. Jamaica is similar to Cuba (with respect to geographical proximity); Cuba is similar to Russia (with respect to their political affinity), but Jamaica and Russia are not similar at all. Geographical proximity and political affinity are SEPARABLE FEATURES. *Cuba* and *Jamaica* are co-hyponyms if we imagine a hypernym *Caribbean islands* of which both concepts are daughters. *Cuba* and *Russia* are co-hyponyms too, but being daughters of another mother, i.e. the concept *communist countries*. The concept *Jamaica* thus inherits features from multiple mothers. What can we say about the transitivity of meaning in this case? The transitivity between two co-hyponyms holds when restricted to single inheritance.

When words are ambiguous, we come to a similar situation. Widdows (2004) gives the following example: *Apple* is similar to *IBM* in the domain of computer companies; *Apple* is similar to *pear*, when we are thinking of fruit. *Pear* and *IBM* are not similar at all. Again, there is the problem of multiple inheritance. *Apple* is a daughter both of the concept *computer manufacturers* and of *fruits*. For co-hyponyms similarity is only transitive in case of single inheritance. The same holds for synonyms. If a word has multiple senses we get into trouble when applying the transitivity of meaning.

Although we have seen many examples of cases where the transitivity of meaning does not hold, we hope to find improvements for finding semantically related words, when using third-order affinity techniques.

4 Methodology

We will now describe the methodology used to compute nearest neighbours (subsection 4.1). In subsection 4.2 we will describe how we have used these nearest neighbours as input to the third-order and combined technique.

4.1 Syntax-based distributional similarity

In this section we will describe the syntactic contexts selected, the data we used, and the measures and weights applied to retrieve nearest neighbours.

4.1.1 Syntactic context

Most research has been done using a limited number of syntactic relations (Lee, 1999; Weeds, 2003). We use several syntactic relations: subject, object, adjective, coordination, apposition, and prepositional complement. In Figure 1 examples are given for these types of syntactic relations.[2]

Subj:	De *kat eet*. 'The cat eats.'
Obj:	Ik *voer* de *kat*. 'I feed the cat.'
Adj:	De *langharige kat* loopt. 'The long-haired cat walks.'
Coord:	*Jip and Janneke* spelen. 'Jip and Janneke are playing.'
Appo:	De *clown Bassie* lacht. 'The clown Bassie is laughing.'
Prep:	Ik *begin met* mijn *werk*. 'I start with my work.'

Figure 1: Types of syntactic relations extracted

4.1.2 Data collection

Because we believe that the method will remedy data sparseness we applied the method to a medium-sized corpus. Approximately 80 million words of Dutch newspaper text.[3] All data 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).

4.1.3 Syntactic co-occurrences

For each noun we find its syntactic contexts in the data. This results in CO-OCCURRENCE VECTORS, such as the vector given in Table 1 for the headword *kat*. These are used to find distributionally similar

[2] We are working on Dutch and we are thus dealing with Dutch data.

[3] This is the so-called CLEF corpus as it was used in the Cross Language Evaluation Forum (CLEF). The corpus is a subset of the TwNC corpus (Ordelman, 2002).

	heb_OBJ	voer_OBJ	harig_ADJ
	‘have_OBJ’	‘feed_OBJ’	‘furry’_ADJ’
kat ‘cat’	50	10	25

Table 1: Syntactic co-occurrence vector for *kat*

words. Every cell in the vector 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 table 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 85. 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. We have set a row cut-off of 10. So only headwords that appear in 10 or more co-occurrence tokens in total are taken into account. We have not set a cutoff for the cell frequency.

4.1.4 Measures and feature weights

Some syntactic contexts are more informative than others. Large frequency counts do not always indicate an important syntactic co-occurrence. 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*. The results of vector-based methods can be improved if we take into account the fact that not all combinations of a word and syntactic relation have the same information value. We have used POINTWISE MUTUAL INFORMATION (PMI, Church and Hanks (1989)) to account for the differences in information value between the several headwords and attributes.

The more similar the co-occurrence vectors of any two headwords are, the more distributionally similar the headwords are. In order to compare the vectors of any two headwords, we need a similarity measure. In these experiments we have used a variant of Dice: Dice†, proposed by Curran and Moens (2002). It is defined as:

$$Dice\dagger = \frac{2\sum min(wgt(W1, *_r, *_{w'}), wgt(W2, *_r, *_{w'}))}{\sum wgt(W1, *_r, *_{w'}) + wgt(W2, *_r, *_{w'})}$$

We describe the function using an extension of the notation used by Lin (1998a), adapted by Curran (2003). Co-occurrence data is described as relation tuples: ⟨word, relation, word′⟩, for example, ⟨cat, obj, have⟩.

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 with any other word it has been found in. $W1$ and $W2$ are the two words we are comparing, and wgt is the weight given by PMI.

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 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† gives the same ranking as the well-known Jaccard measure, i.e. there is a monotonic transformation between their scores. Dice† is easier to compute and therefore the preferred measure (Curran and Moens, 2002). Choices for measures and weights are based on previous work (van der Plas and Bouma, 2005).

4.2 Syntactic co-occurrences and nearest neighbours

The syntactic co-occurrence vectors have co-occurrence frequencies as values. An example is given in Figure 2.

GRACHT ‘canal’		
97	Amsterdams_ADJ	‘Amsterdam_ADJ’
26	ben_SUBJ	‘am_SUBJ’
12	word_SUBJ	‘become_SUBJ’
9	straat_CONJ	‘street_CONJ’
9	gedempt_ADJ	‘closed_ADJ’
8	Utrechts_ADJ	Utrecht_ADJ
5	wal_CONJ	‘shore_CONJ’
5	muur_CONJ	‘wall_CONJ’
5	moet_SUBJ	‘has to_SUBJ’
5	graaf_OBJ	‘ditch_OBJ’

Figure 2: Syntactic co-occurrences for the word *gracht* ‘canal’

To retrieve nearest neighbours, needed for the third-order technique, we computed for each noun a ranked list of most similar words using the methodology described in the two previous sections, i.e. by comparing the weighted feature vector of the headword with all other words in the corpus. We collected the 3 most similar nouns to all nouns. These are the nearest neighbours that will be input to our third-order system.

Now, how do we construct a second-order vector from these nearest neighbours? The cells of the second-order vectors that we want to construct should reflect the similarity between pairs of words. The scores given to the pairs of words by the system do not usually reflect the similarity very well across different headwords and discriminates too little between different nearest neighbours for a given headword.

Instead we used the ranks or rather reversed ranks for a given candidate word. However, the decrease in similarity between the first candidate and the second is not linear. It decreases more rapidly. After inspecting the average decrease in similarity for nearest neighbours, when going down the ranked list, we decided to use a scoring method that is in line with Zipf’s law (Zipf, 1949). We decided to attribute similarity scores that are decreasing very rapidly for the first ranks and less as we go down the ranked list of nearest neighbours.

Apart from deciding on the slope of the similarity score we needed to set a start value. We decided to choose a start value according to the highest co-occurrence frequency (in the syntactic co-occurrences) for that headword. So if a headword’s highest co-occurrence frequency was 100, a similarity score of 100 is given to the word at the first rank (that is itself) and a score of 50 to the candidate word at the second rank and so on. The intuition between this is that we want to balance the importance given to nearest neighbours and syntactic co-occurrences. The importance of the nearest neighbours will not tresspass the importance of the syntactic co-occurrences.

GRACHT ‘canal’		
97	gracht	‘canal’
48	laan	‘avenue’
32	sloot	‘ditch’

Figure 3: Nearest neighbours for the word *gracht* ‘canal’

The highest score will be given to the second-order affinity between a headword and itself. This seems an unnecessary addition, but it is not, because we want *canal* to be similar to words that have *canal* as a second-order affinity as well.

The second-order similarity score (SOSS) for a given headword (h) and a given nearest neighbour (nn) is defined as follows:

$$\textit{SOSS(h,nn)} = \frac{max.freq.of.coocc(h)}{rank(nn)}$$

We have given an example of the second-order feature vector of the word *gracht* ‘canal’ in Figure 3. As we see the highest score is given to second-order affinity between the headword and the headword itself : *gracht-gracht*. This score is taken from the highest co-occurrence frequency found for the word *gracht* as can be seen in Figure 2. Second-order feature vectors such as given in Figure 3 are constructed for all headwords to be used as input to the third-order technique. For the combined technique we concatenated both types of data. So the input to the combined technique for the word *canal* would be all its syntactic co-occurrences of which a subset is given in Figure 2 plus the three nearest neighbours given in Figure 3.

5 Evaluation

In the following subsections we will first explain how we determined the semantic similarity of the retrieved nearest neighbours (subsection 5.1) and then we will describe the test sets used (subsection 5.2).

5.1 EWN similarity measure and synonyms

Like most researchers in the field of distributional methods we have little choice but to evaluate our work on the resource that we want to enrich. We want to be able to enrich Dutch EuroWordNet (EWN, Vossen (1998)), but at the same time we use it to evaluate on. Especially for Dutch there are not many resources to evaluate semantically related words available.

For each word we collected its k nearest neighbours according to the system. For each pair of words[4] (target word plus one of the nearest neighbours) we calculated the semantic similarity according to EWN. We used the Wu and Palmer measure (Wu and Palmer, 1994) applied to Dutch EWN for computing the semantic similarity between two words.[5] The EWN similarity of a set of word pairs is defined as the average of the similarity between the pairs.

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:

$$\mathit{Sim(W1,W2)} = \frac{2(D3)}{D1 + D2 + 2(D3)}$$

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.

Some words returned by the system as nearest neighbours cannot be found in EWN. Because counting the words not found in EWN as errors would be too harsh[6] we select the next nearest neighbour that is found in EWN, when encountering a not-found word.

The Wu and Palmer measure gives an indication of the degree of semantic similarity among the retrieved neighbours. The fact that it combines several lexical relations, such as synonymy, hyponymy, an co-hyponymy is an advantage on the one hand, but it is coupled with the disadvantage that it is a rather opaque measure. We have therefore decided to look at one lexical relation in particular: We calculated the percentage of synonyms according to EWN. Note that it is a very strict evaluation and the numbers will therefore be relatively low. Because Dutch EWN is much smaller than Princeton WordNet many synonyms are missing.

		EWN similarity			
		k=1	k=3	k=5	k=10
VLF	2	0.391	0.378	0.364	0.350
	2-3	0.395	0.392	0.376	0.359
	3	0.413	0.412	0.411	0.410
LF	2	0.433	0.408	0.392	0.371
	2-3	0.434	0.417	0.401	0.381
	3	0.437	0.426	0.426	0.428
MF	2	0.644	0.605	0.586	0.555
	2-3	0.646	0.608	0.589	0.561
	3	0.643	0.608	0.589	0.575
HF	2	0.719	0.672	0.645	0.610
	2-3	0.718	0.674	0.645	0.612
	3	0.720	0.670	0.639	0.615

Table 2: EWN similarity several values of k for the four test sets

5.2 Test sets

To evaluate on EWN, we have used four test sets of each 1000 words ranging over four frequency bands: high-frequency, middle frequency, low-frequency, and very-low frequency. For every noun appearing in EWN we have determined its frequency in the 80 million-word corpus of newspaper text. 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’). The low-frequency test set has frequencies ranging between 28 (*röntgenonderzoek*, ‘x-ray research’) and 23 (*vriendenprijs*, ‘paltry amount’). For the very low frequency test set the frequency goes from 9 (*slaginstrument* ‘percussion instrument’) to 8 (*cederhout* ‘cedar wood’).

[4] If a word is ambiguous according to EWN, i.e. is a member of several synsets, the highest similarity score is used.

[5] This measure correlates well with human judgements (Lin, 1998b) without the need for sense-tagged frequency information, which we believe is not available for Dutch.

[6] Dutch EWN is incomplete. It is about half the size of Princeton WordNet (Fellbaum, 1998). Nearest neighbours that are not found in EWN might be valuable additions that we do not want to penalise the system too much for.

6 Results and discussion

In Table 2 the results of using second-order (2), combined (2+3), and third-order (3) techniques is presented. The average EWN similarity is shown at several values of k. At k=1 the average EWN similarity between the test word and the nearest neighbour at the first rank is calculated. For k=3 we average over the top-three nearest neighbours returned by the system and so on. Results are given for each of the four test sets, the very-low-frequency set (VLF), the low-frequency test set (LF), the middle-frequency test set (MF), and high-frequency test set (HF).

We can easily compare the scores from the second-order technique and the combined technique. The scores for the third-order technique is a little more difficult to compare because, since there is very little data, it is often not possible for all test words to find the number of nearest neighbours given under k. The coverage of the third-order technique is low, especially for the very-low to low-frequency test set. Already at k=1 the number of test word is about 60% and 70% (resp.) of the number of nearest neighbours found when using the second-order technique. For the middle and high-frequency test set the number of nearest neighbours found is comparable, but less for high values of k.

Let us compare the second-order and combined techniques since coverage of these techniques is more comparable.[7] We see that the combined method outperforms the second-order method for almost all test sets. For the high frequency test set there is no difference in performance and for the middle-frequency testset the differences are very small too. The largest improvements are for the very-low-frequency and low-frequency test set. This is expected, since the method was introduced to remedy data sparseness and for these words data sparseness is most severe. We can conclude that exploiting the transitivity of meaning by augmenting the input to the system with nearest neighbours from a previous round results in a higher degree of semantic similarity among very-low and low-frequency words. The differences in performance are small, but we should keep in mind that that EWN similarity does not go from 0 to 1. The random baseline reported in van der Plas (2008), i.e. the score obtained by picking random words from EWN as nearest neighbours of a given target word, is 0.26 at k=5 and a score of 1 is impossible unless all words in the testset have k synonyms.

	Synonyms			
	k=1	k=3	k=5	k=10
HF				
2	143(14.39)	276(9.26)	357(7.18)	461(4.64)
2+3	148(14.89)	275(9.22)	356(7.16)	465(4.68)
3	154(15.54)	259(8.84)	315(6.73)	382(5.26)
MF				
2	105(10.56)	194(6.51)	245(4.93)	312(3.14)
2+3	109(10.97)	200(6.71)	250(5.03)	318(3.20)
3	107(11.38)	173(6.60)	198(5.07)	214(3.95)
LF				
2	33(3.75)	65(2.47)	87(2.00)	108(1.28)
2+3	34(3.86)	73(2.77)	88(2.01)	113(1.32)
3	25(4.01)	41(3.18)	48(3.10)	54(3.20)
VLF				
2	2(0.54)	4(0.36)	8(0.44)	10(0.30)
2+3	2(0.54)	4(0.36)	9(0.49)	10(0.29)
3	2(0.91)	2(0.50)	2(0.44)	2(0.42)

Table 3: Number of synonyms at several values of k for the four test sets

To get a better idea of what is going on we inspected the nearest neighbours that are the output of the system. There seemed to be many more synonyms in the output of the combined method than in the output of the second-order method. Because synonymy is the lexical relation that is at the far end of semantic similarity, it is important to find many synonyms. To quantify our findings we determined the number of synonyms among the nearest neighbours according to EWN.

In Table 3 the number of synonyms as well as the percentage of synonyms found at several values of k is shown.[8]

Our initial findings proved quantifiable. The combined technique (2+3) results in more synonyms. Most surprising are the results for the high-frequency testset. Whereas, based on evaluations with the EWN similarity scores, we believed the method did not do much good for the high-frequency

[7] In fact, the coverage of the combined method is a bit higher, because it combines two types of data, but the differences are not as big as between the third-order and the second-order technique.

[8] At k=n we do not always find n nearest neighbour for all words in the test set. That is the reason for showing both counts and percentages in the table.

Second-order				Combined
cassette	**videoband**	**bandje**	**CDi**	**cassette**
cassette	videoband	bandje	CDi	cassette
videoband	cassette	cassette	DCC	videoband
CDi	videofilm	videoband	CD	bandje

Figure 4: Nearest neighbours for *videoband* 'video tape', *cassette* 'cassette' *bandje* 'tape' and *CDi* 'CDi'

method, we now see that the number of synonyms found is higher when using the combined technique, especially at k=1. This holds for all but one test set. Only for the very low frequency test set there is hardly any difference.

We explained before that coverage of the third-order technique is low. However, we see that the technique results in higher numbers of synonyms found at k=1 for the high-frequency (+11) and the middle-frequency test set (+2). At higher values of k the absolute numbers are smaller for the third-order technique and also for the low and very-low-frequency test set. This is to be expected because the number of nearest neighbours found dramatically decreases, when using a third-order technique on its own. But it is surprising that we are able to extract more synonyms, when using only the two nearest neighbours (plus the headword itself) computed by the system before as input.

Manual inspection showed that what happens is that nearest neighbours that have each other as nearest neighbour are promoted. As can be seen in Figure 4, *cassette* 'cassette' has *videoband* 'video tape', and *CDi* as nearest neighbour. Because CDi has no nearest neighbours in common with *cassette*, except itself, it is demoted in the output of the combined method. The word *bandje* 'tape' has two neighbours in common with *cassette*. *Bandje* is promoted in the output of the combined method.

This finding bring us to work by Lin (1998a), where the author shows that, when selecting only respective nearest neighbours (words that have each other as the one most nearest neighbour), the results are rather good. Our technique incorporates that notion, but is less restricted, especially in the combined technique.

7 Conclusion and future work

Guided by the idea of the transitivity of meaning we have shown that by augmenting syntactic co-occurrences (that are usually input to distributional methods) with nearest neighbours (the output of the system from a previous round) we are able to improve the performance on low- and middle-frequency words with respect to semantic relatedness in general. This result is encouraging, because distributional methods usually perform rather poorly on low- and middle-frequency words. In addition, these are the words that are most sought after, because they are the ones that are missing in existing resources. There is something to be gained for the high-frequency to low-frequency words in addition. The percentage of synonyms found is larger when using combined techniques.

In future work we are planning to implement a more principled way of combining syntactic-co-occurrences and nearest neighbours. The method and results presented here sufficed to support our intuitions, but we believe that more convincing numbers could be attained when fully exploiting the principle. Since the method uses a combination of labelled and unlabelled data (although in our case the labelling is the result of the same unsupervised method and not of manual annotation), we plan to consult the literature on co-training (Blum and Mitchell, 1998). Also, instead of expanding the syntactic co-occurrences of words with their nearest neighbours we could expand them with the syntactic co-occurrences of their nearest neighbours to arrive at more uniform data. Lastly, the technique allows for iteration. We could measure the performance at several iterations.

Acknowledgements

The research leading to these results has received funding from the EU FP7 programme (FP7/2007-2013) under grant agreement nr 216594 (CLASSIC project: www.classic-project.org) and from NWO, the Dutch Organisation for Scientific Research in the framework of the research program for *Interactive Multimedia Information eXtraction*, IMIX.

References

E. Alfonseca and S. Manandhar. 2002. Extending a lexical ontology by a combination of distributional semantics signatures. In *Proceedings of EKAW*.

C. Biemann, S. Bordag, and U. Quasthoff. 2004. Automatic acquisition of paradigmatic relations using iterated co-occurrences. In *Proceedings of LREC*.

A. Blum and T. Mitchell. 1998. Combining labeled and unlabeled data with co-training. In *Proceedings of the 1998 conference on computational learning theory*.

K.W. Church and P. Hanks. 1989. Word association norms, mutual information and lexicography. *Proceedings of the Annual Conference of the Association of Computational Linguistics (ACL)*.

J.R. Curran and M. Moens. 2002. Improvements in automatic thesaurus extraction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing, EMNLP*, pages 222–229.

J.R. Curran. 2003. *From Distributional to Semantic Similarity*. Ph.D. thesis, University of Edinburgh.

I. Dagan, L. Lee, and F. Pereira. 1999. Similarity-based models of word cooccurrence probabilities. *Machine Learning*, 34(1-3):43–69.

P. Edmonds. 1997. Choosing the word most typical in context using a lexical co-occurrence network. In *Proceedings of the European chapter of the Association for Computational Linguistics*, pages 507–509.

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

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

Z.S. Harris. 1968. *Mathematical structures of language*. Wiley.

L. Lee. 1999. Measures of distributional similarity. In *37th Annual Meeting of the Association for Computational Linguistics (ACL)*.

B. Lemaire and G. Denhire. 2006. Effects of high-order co-occurrences on word semantic similarities. *Current Psychology Letters - Behaviour, Brain and Cognition*, 18(1).

D. Lin. 1998a. Automatic retrieval and clustering of similar words. In *Proceedings of COLING/ACL*.

D. Lin. 1998b. An information-theoretic definition of similarity. In *Proceedings of the 15th International Conference on Machine Learning*.

M. Moortgat, I. Schuurman, and T. van der Wouden. 2000. CGN syntactische annotatie. Internal Project Report Corpus Gesproken Nederlands, available from `http://lands.let.kun.nl/cgn`.

R.J.F. Ordelman. 2002. Twente nieuws corpus (TwNC). Parlevink Language Techonology Group. University of Twente.

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

H. Schütze and M. Walsh. 2008. A graph-theoretic model of lexical syntactic acquisition. In *Proceedings of EMNLP*.

A. Tversky and I. Gati, 1978. *Cognition and Categorisation*, chapter Studies of similarity, pages 81–98. Erlbaum.

L. van der Plas and G. Bouma. 2005. Syntactic contexts for finding semantically similar words. In *Proceedings of Computational Linguistics in the Netherlands (CLIN)*.

L. van der Plas. 2008. *Automatic lexico-semantic acquisition for question answering*. Ph.D. thesis, University of Groningen.

G. van Noord. 2006. At last parsing is now operational. In *Actes de la 13eme Conference sur le Traitement Automatique des Langues Naturelles*.

P. Vossen. 1998. EuroWordNet a multilingual database with lexical semantic networks.

J. Weeds and W. Weir. 2005. Co-occurrence retrieval: A flexible framework for lexical distributional similarity. *Computational Linguistics*, 31(4):439–475.

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

D. Widdows. 2004. *Geometry and Meaning*. Center for the Study of Language and Information/SRI.

Z. Wu and M. Palmer. 1994. Verb semantics and lexical selection. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

G.K. Zipf. 1949. *Human behavior and the principle of the least effort.* Addison-Wesley.