Utilizing the One-Sense-per-Discourse Constraint for Fully Unsupervised Word Sense Induction and Disambiguation

Reinhard Rapp

University of Mainz, FASK
76711 Germersheim, Germany
rapp@mail.fask.uni-mainz.de

Abstract

Recent advances in word sense induction rely on clustering related words. In this paper, instead of using a clustering algorithm, we suggest to perform a Singular Value Decomposition (SVD) which can be guaranteed to always find a global optimum. However, in order to apply this method to the problem of word sense induction, a semantic interpretation of the dimensions computed by the SVD is required. Our finding is that in our specific setting the first dimension relates to semantic similarities between words, and the second dimension distinguishes between the two main senses of an ambiguous word. Based on this result we present an algorithm for fully unsupervised word sense induction and disambiguation.

1. Introduction

Past work on word senses has concentrated on disambiguation, i.e. on choosing among a predefined set of senses when given an ambiguous word in context. However, hardly any work was done on sense induction, that is the automatic discovery of the possible senses for a word. Only recently a number of papers have been published on this topic, among them Pantel & Lin (2002), Neill (2002), Dorow & Widdows (2003), and Rapp (2003).

All these papers rely on the distributional hypothesis, i.e. the observation that words with similar meanings tend to have similar lexical surroundings. In that they use global co-occurrence vectors, i.e. vectors that represent the overall behavior of a word in an entire corpus, they also have a common limitation: Since most words are semantically ambiguous, the observed vectors reflect a mixture of the contextual behavior of a word's senses.

Therefore, when starting from global vectors the task of sense induction requires determining the co-occurrence vectors of the senses given the co-occurrence vector of an ambiguous word. This is a difficult problem that can be approached in different ways:

1. Pantel & Lin (2002) cluster all words in a vocabulary, then deduct a word's co-occurrence vector from the centroid of its cluster, and finally look at the similarity of the differential vector to other clusters.
2. Rapp (2003) postulates that the best descriptors for the senses of an ambiguous word can be found by looking for words whose co-occurrence vectors are as dissimilar as possible but nevertheless add up to the co-occurrence vector of the ambiguous word.
3. A similar approach is to look at the co-occurrence vectors of the, say, 20 strongest first-order associations to the ambiguous word, and to cluster the product vectors resulting from all 190 possible pairs. The clustering works especially well if we consider binary vectors and eliminate all vector positions that are not significantly associated with the ambiguous word.
4. Rapp (2004) assumes that the sense distribution of an ambiguous word varies with genre. On the basis of three corpora from different domains for each word he derives three co-occurrence vectors that are considered to be different mixes of the underlying senses. He then uses independent component analysis to recover the vectors of the senses from the observed mixtures.

However, since reconstructing the sense vectors from the mixtures is difficult and often suffers from the sparse data problem, the question is if we really need to base our work on mixtures or if there is some way to directly observe the contextual behavior of the senses, thereby avoiding the mixing beforehand. For this we suggest two possibilities:

1. Some semantic ambiguities (usually the coarser ones) can be resolved by syntactic considerations. For example, the word *sound* can be used as a noun, a verb, or an adjective and has a different sense in each case. This kind of ambiguity can be taken into account by working with a part-of-speech tagged or a parsed corpus and by using different co-occurrence vectors for each possible part of speech (Pantel & Lin, 2002).
2. We can look at local instead of global co-occurrence vectors. As can be seen from human performance, in almost all cases the local context of an ambiguous word is sufficient to disambiguate the word. From this observation we conclude that the local context of a word usually carries no ambiguities.

The first approach has already been applied by Pantel & Lin (2002) with reasonably good results for finding the main senses of a word. Therefore, in this paper we will concentrate on the second approach. It is closely related to the one-sense-per-discourse constraint formulated by Gale et al. (1992) that has been proven effective in word sense disambiguation. The argument here is that in a given discourse an ambiguous word usually tends to occur in only a single sense. By looking at close neighborhoods instead of entire documents we take this position to an extreme. The aim of this paper is to show how the one-sense-per-discourse constraint, whose application tends to suffer from the sparse data problem, can be successfully exploited for word sense induction.

2. Approach

The basic idea is that we do not cluster the global co-occurrence vectors of the words (on the basis of an entire corpus) but local ones which are based on the contexts of a single word. That is, our matrix is derived from the concordance of this word. Also, we do not consider a term/term but a term/context matrix. For each word in our vocabulary we obtain an entire matrix whose context vectors when summed up form one global vector. When for all

words of a vocabulary such matrices are constructed, by putting them together one could speak of a three dimensional array, with two dimensions being the words in the corpus and the third dimension being all contexts.

Let us exemplify this using the ambiguous word *palm* with its *tree* and *hand* senses. If we assume that our corpus has six occurrences of *palm*, i.e. there are six local contexts, then we can derive six local co-occurrence vectors for *palm*. Considering only strong associations to *palm*, these vectors could, for example, look as in table 1.

	arm	beach	coco-nut	fin-ger	hand	shoul-der	tree
c1	•			•	•	•	
c2		•	•				•
c3	•				•		
c4			•				•
c5		•	•				
c6	•				•	•	

Table 1. Context matrix for the word *palm*.

The dots in the matrix indicate if the respective word occurs in a context or not. We use binary vectors since we assume short contexts where words usually occur only once. By looking at the matrix it is easy to see that contexts c1, c3, and c6 seem to relate to the *hand* sense of *palm*, whereas contexts c2, c4, and c5 relate to the *tree* sense. Our intuitions can be resembled by using a method for computing vector similarities, for example the (binary) Jaccard-measure. If we then applied an appropriate clustering algorithm to the context vectors, we would probably obtain the two expected clusters, and the words closest to the geometric centers of the clusters should be good descriptors for the senses of *palm*.

However, as matrices of the above type can be very large and extremely sparse, clustering is a difficult task, and common algorithms often deliver sub-optimal results because they can easily get stuck in local minima. Fortunately, the problems of matrix size and sparseness can be minimized by reducing the dimensionality of the matrix. An appropriate algebraic method that has the capability to reduce the dimensionality of a rectangular or square matrix in an optimal way is SVD. As shown by Landauer & Dumais (1997), Schütze (1997), Rapp (2003), and others, by reducing the dimensionality a generalization effect is achieved that can improve results. As Schütze (1997:190) puts it: "The role of SVD is to transform distributional matrices in order to bring out generalizations in distributional patterns that the original matrix does not show due to natural randomness in sampling from a corpus."

As this method is rather sophisticated, we can not go into the details here. Good descriptions can be found in Landauer & Dumais (1997), Manning & Schütze (1999), and Press et al. (1992). The essence is that by computing the singular values of a matrix and by truncating the smaller ones, SVD allows to significantly reduce the number of columns, thereby (in a least squares sense) optimally preserving the euclidean distances (and angles) between the lines (Schütze, 1997:191). Alternatively, it is also possible to reduce the number of lines thereby preserving the distances between the columns.

A more or less obvious way to utilize SVD in our setting would involve reducing the number of lines (contexts) to, say, 100, and then applying a clustering algorithm to the column vectors of the resulting matrix. This approach should work well since it is a strength of SVD to reduce the effect of sampling errors and to close gaps in the data. However, since our local matrices (based on the concordance of a word) are smaller and more sparse than the global co-occurrence matrices used in other studies, it is probably advisable to use fewer dimensions. Depending on the corpus frequency of the given word, we suggest to use 50 to 100 dimensions, which compares to 200 or 300 dimensions used in the literature for co-occurrence matrices of global vectors derived from a full corpus (Rapp, 2003).

However, in the work described here we did not further pursue this approach but instead decided to implement a more innovative method. It has the potential of being easier to implement, more elegant, and more accurate, but also rises difficult questions.

Our claim is that we don't need a separate clustering algorithm since the process of SVD involves an implicit clustering. However, to see this we need to semantically interpret the dimensions of the reduced space. This is a difficult problem, as can be seen from the following citation (Schütze, 1997:195): "I have not been able to find good interpretations of the dimensions of the reduced SVD-spaces. Apparently, meaningful generalizations can only be made about patterns of values that involve all dimensions. This is precisely what the clustering techniques I have used do."

Of course, from algebra we know that the dimensionality reduced vectors form an orthonormal basis of the vector space (Press et al. 1992:61), which means that all column vectors are orthogonal to each other (linearly independent) and their lengths are 1.

But what does this mean for the semantic interpretation of the reduced vectors? One could think that SVD collapses those columns of our matrix that correspond to the same senses. Therefore, if in the matrix of table 1 we reduced the number of columns to two, this would mean that one column of the reduced matrix should correspond to the columns for *arm*, *finger*, *hand*, and *shoulder* in the original matrix, and the other to *beach*, *coconut*, and *tree*. However, this is a misinterpretation of what SVD does.

What actually happens is that SVD determines the main dimensions of our semantic space. In the following explanations, for simplicity we assume that the columns of our reduced matrix are sorted according to the size of their corresponding singular values, i.e. the first column corresponds to the most important dimension, the second to the next important dimension, and so on.

A dimension can be characterized by a scale which is best described by its extremes. In our setting, the first dimension seems to relate to the contextual similarity towards the given word, i.e. to *palm*. Therefore, at one end of the first dimension's scale we have words that are similar to *palm*, for example *tree* or *hand*, at the other end words that are not similar, for example *synthesis* or *volt*.

In determining the second dimension, SVD removes the first dimension from the data. What we then find is that the extremes of the second dimension's scale are the two meanings of *palm*, i.e. at one end we find *tree* and at the other we find *hand*.

Now if SVD also removes this second dimension from the data, one could hope that the third dimension gives us further distinctions. For example, given the *tree* sense of

palm, one could distinguish products (*palm oil*) and names (*Palm Beach*). Or given the *hand* sense, further distinctions could be *body part* versus *handheld computer*. So the third and following dimensions would capture the next strongest regularities hidden in the data that only become visible after removal of the previous dimensions.

However, since with higher dimensions the effects become less clear and sampling errors tend to dominate, we have not yet been able to clearly support this view empirically. In what follows we therefore restrict our discussion to the first and the second dimensions.

Having formulated interpretations concerning the dimensions of the reduced space, the question is for what kinds of applications this is of relevance, and in how far this view leads to useful practical results. Let us therefore briefly look at the computation of semantically similar words and then move on to the tasks mentioned in the title of the paper, namely word sense induction and word sense disambiguation.

3. Word similarities and sense induction

Our computations are based on a partially lemmatized version of the British National Corpus (BNC) which has the function words removed (Rapp, 2002). With partial lemmatization we mean that only those words in the corpus have been replaced by their root forms that, according to a large lexicon of English, can be unambiguously assigned to a stem. This makes the corpus more manageable, the computations faster, and reduces the sparse data problem without introducing errors (other than those resulting from errors or omissions in the lexicon). Our vocabulary consists of all 374240 different word forms occurring in this corpus after lemmatization.

Next we have to specify how we define a context. Since the documents in the BNC are rather long (average sample size is 24274 words), it is probably better to choose shorter contexts, for example sentences, paragraphs, or text windows of a fixed size. We decided to use text windows of ±20 words around the given word. Since function words were removed from our corpus, this corresponds to a larger window size of perhaps ±40 words in the original corpus if we assume that roughly every second word is a function word.

Due to space constraints, the following considerations and results will be exemplified by only a single test word. We chose the word *palm* which has already been used in the previous examples. In our corpus, *palm* occurs 2054 times. Note that this occurrence frequency relates to several inflected forms of *palm* (e.g. *palms* and *palming*) as the corpus has been lemmatized.

Using the window size of ±20 words we created a concordance of *palm*, with each line in the concordance relating to one context window. From this concordance we computed a term/context-matrix whose binary entries indicate if a word occurs in a particular context. When constructing the matrix, for the lines we took all 2054 contexts of *palm* into account, but restricted the columns to those words other than *palm* that have a corpus frequency of at least 100 in the lemmatized BNC. The reason for this restriction is that the following SVD can be computationally demanding, and that in a previous study the omission of infrequent words seemed to have little impact on the results (Rapp, 2003). The resulting matrix had a size of 2054 lines × 10610 columns.

As proposed in other papers (e.g. Landauer & Dumais, 1997; Rapp, 2003), we considered applying some association measure to the matrix entries. However, simply adopting the association formulas used elsewhere seemed not appropriate, as we are working with a different type of matrix that consists of local instead of global vectors. Unfortunately, finding the best formula is difficult, as many considerations must be taken into account. Possibilities include replacing the matrix entries by the association strengths of the respective word to *palm*, or to simply use the salience of a word which can, for example, be expressed by its entropy, by the standard deviation of its distribution, or possibly by the logarithm of its inverse frequency. Since we have not yet completed these experiments, the results that we present below are based on the unmodified binary matrix.

We applied the SVD to this matrix and reduced its number of columns from 10610 to 2, whereas the number of lines remained at 2054. This took only a few seconds of computing time, and the singular values that we obtained were 62.9 and 25.7 (subsequent values are 22.1 and 19.7).

Next comes the essential step which is to find out about the semantic interpretations of the two computed columns. This we did by comparing each of the two column vectors of the reduced matrix to all column vectors (words) of the original binary matrix and by ranking the results according to the similarities obtained. As our similarity measure we used the well known cosine coefficient (Rapp, 2003) which computes the cosine of the angle between two vectors.

The resulting ranked word list obtained from the comparison with the first vector of the reduced space (first dimension) is shown in table 2, and the list relating to the second dimension is shown in table 3. For each word, the tables show its similarity value (cosine coefficient) and its rank among all words in the vocabulary. As we have a large vocabulary, we had to restrict the lists to a selection of words. As such we took the top 30 words with the strongest associative relationship to *palm*. These first-order associations had been automatically computed using the log-likelihood ratio as our association measure and the lemmatized BNC as our corpus. The exact procedure is described in Rapp (2002).

RK	COS	WORD
1	0.57	hand
3	0.45	back
6	0.42	hands
10	0.38	tree
13	0.35	hold
15	0.33	finger
16	0.32	head
21	0.30	eyes
53	0.24	arms
56	0.24	beach
62	0.23	shoulder
63	0.23	mouth
64	0.23	white
90	0.21	skin
97	0.21	slowly

RK	COS	WORD
104	0.20	upward
109	0.20	lip
127	0.19	gently
134	0.18	facing
164	0.17	fist
170	0.17	island
200	0.16	thumb
206	0.16	elbow
243	0.15	slap
252	0.15	oil
328	0.14	coconut
459	0.12	frond
600	0.11	attacker
611	0.11	Florida
633	0.10	breathe

Table 2. Cosine similarities between the column vectors of the original matrix and the first column (first dimension) of the reduced matrix.

RK	COS	WORD
1	0.44	hand
2	0.33	hands
3	0.27	finger
4	0.27	hold
5	0.26	back
6	0.23	facing
7	0.23	arms
8	0.22	upward
9	0.21	head
11	0.19	slowly
15	0.18	shoulder
17	0.17	mouth
18	0.16	lip
24	0.16	elbow
29	0.15	eyes

RK	COS	WORD
35	0.14	thumb
43	0.12	gently
50	0.12	fist
67	0.11	skin
104	0.09	breathe
111	0.09	attacker
119	0.08	slap
374169	-0.12	oil
374177	-0.12	frond
374191	-0.13	Florida
374227	-0.19	coconut
374232	-0.21	white
374238	-0.27	island
374239	-0.38	beach
374240	-0.58	tree

Table 3. Similarities regarding the second dimension. The total number of words in the vocabulary is 374240.

If we look at table 2, we can see that our first dimension seems to correspond to association strength, as all of our 30 test words are among the top 633 in our ranked list of altogether 374240 words. Of course, ideally one could have hoped that all test words would end up on exactly the top 30 positions. But this is unrealistic, as different methods for computing association strengths (in our case log-likelihood versus SVD) will always lead to variations in results. Also, one should keep in mind that these results have been achieved by applying the SVD to a binary co-occurrence matrix that had not been transformed using any association measure.

When considering the results for the second dimension as shown in table 3, we see that 22 of our 30 test words are ranked among the top 119, but that 8 are among the bottom 72. Interestingly, according to our judgement, all of the top words relate to the *hand* sense of *palm*, and all of the bottom words relate to its *tree* sense. This is a result that is extremely unlikely to occur incidentally. Moreover, the best descriptors for the two senses of *palm*, namely *hand* and *tree*, are placed at the very first respectively last position of our ranked list. From this it becomes clear that the second dimension that SVD computed is capable of distinguishing between the two main senses of *palm*.

4. Word sense disambiguation

A nice feature of the proposed SVD-based method for word sense induction is that it implicitly performs sense disambiguation for each of the 2054 contexts of *palm*. The second column in our reduced matrix (i.e. the one that distinguishes between the senses) represents the results of this disambiguation process. Remember that this vector has a length of 2054 so that each value corresponds to one of the contexts. If we now assume that a positive value in this vector indicates that *palm* is used in the sense of *hand* in the respective context, and that a negative value indicates the use of *palm* in the sense of *tree*, then according to our judgment for the first 100 occurrences of *palm* in the BNC we get 87 correct predictions.

If we want to disambiguate the sense of *palm* given a new context, we can either add this context to our original matrix and redo the SVD. Alternatively, there is also the possibility of mapping the new context into the reduced space, as described in Manning & Schütze (1999:563).

5. Summary and conclusions

We have presented a method for fully automatic word sense induction and disambiguation from text. It relies on the semantic interpretation of a dimensionality reduced matrix of contexts. As this is ongoing work, there are many open questions: Can the results that have been presented here for a single example be generalized? Will the dimensions that SVD computes always agree with our intuitions? What is the meaning of the third and the following dimensions? What happens if several singular values are of similar size? What about ambiguous words with more than two senses?

Nevertheless, the simplicity of the approach is surprising, and SVD has the advantage that it is an analytical method that always finds optimal dimensions. If the hypothesis of this paper, namely that SVD dimensions agree with human intuitions, can be confirmed, then this could be useful for solving many problems in natural language processing. Other applications that we have started to explore include language identification in mixed language corpora, text categorization, and part-of-speech induction.

6. References

Dorow, B; Widdows, D. (2003): Discovering corpus-specific word senses. *EACL 2003*, Budapest, conference companion (research notes and demos), 79–82.

Gale, W.; Church, K.; Yarowsky, D. (1992). A method for disambiguating word senses in a large corpus. *Computers and the Humanities*, 26, 415–439.

Landauer, T. K.; Dumais, S. T. (1997). A solution to Plato's problem: the latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological Review*, 104(2), 211–240.

Manning, C.D.; Schütze, H. (1999): *Foundations of Statistical Natural Language Processing*. Cambridge, MA: MIT Press.

Neill, D. B. (2002). *Fully Automatic Word Sense Induction by Semantic Clustering*. Cambridge University, Master's Thesis, M.Phil. in Computer Speech.

Pantel, P.; Lin, D. (2002). Discovering word senses from text. In: *Proceedings of ACM SIGKDD*, Edmonton, 613–619.

Press, W.H.; Teukolsky, S.A.; Vetterling, W.T.; Flannery, B.P. (1992): *Numerical Recipes in C. The Art of Scientific Computing*, 2nd ed. Cambridge Univ. Press.

Rapp, R. (2002). The computation of word associations: comparing syntagmatic and paradigmatic approaches. *Proc. of 19th COLING*, Taipei, ROC, Vol. 2, 821–827.

Rapp, R. (2003). Word sense discovery based on sense descriptor dissimilarity. In: *Proceedings of the Ninth Machine Translation Summit*, New Orleans, 315–322.

Rapp, R. (2004). Mining text for word senses using independent component analysis. In: *Proc. of the SIAM Int. Conf. on Data Mining*, Lake Buena Vista, Florida.

Schütze, H. (1997). *Ambiguity Resolution in Language Learning: Computational and Cognitive Models*. Stanford: CSLI Publications.

Acknowledgements

I would like to thank Manfred Wettler and Raz Tamir for interesting discussions, Hinrich Schütze and Mike Berry for the SVD software, and the DFG for financially supporting this work.