Word Sense Disambiguation Using Co-Occurrence Statistics on Random Labels

Martin Hassel
KTH KOD
Royal Institute of Technology
SE-100 44 Stockholm
Sweden
xmartin@kth.se

Abstract

In this paper we present experiments using Random Indexing for "query expansion" in Word Sense Disambiguation. Random Indexing is an efficient, scalable and incremental latent semantic indexing method somewhat akin to LSA, and has in these experiments shown promising results on a small test set for Swedish with an accuracy up to 80% with relatively little training data. We also compare it to results obtained when applying a Naïve Bayes classifier to the same training and data sets, retrieving a maximum accuracy of 56%.

1 Introduction

A given word can have several *senses*. For example, the word "hot" can mean a high temperature, fiery, excited, eager, spicy or simply incredibly good-looking. A word sense is thus a given *meaning* of a word. While humans display an uncanny ability to select the appropriate meaning when hearing such words in context, natural language applications do seldom fare as well.

The automatic disambiguation of word senses has been an interest and concern since the earliest days of computer treatment of language in the 1950s. Word sense disambiguation (WSD) is an "intermediate task" (Wilks & Stevenson 96), which is not an end in itself, but rather is necessary at one level or another to accomplish many natural language processing tasks. It is obviously essential for language understanding applications, such as message understanding and man-machine communication; and is at least helpful for applications whose aim is not language understanding, e.g. machine translation, information retrieval and hypertext navigation, content and thematic analysis, grammatical analysis, speech processing, and text processing.

Since the senses being discriminated between all are realized with the same lexical sequence, disambiguation work traditionally involves matching the context of the instance of the word to be disambiguated with either information from an external knowledge source (knowledgedriven WSD), or information about the contexts of previously disambiguated instances of the word derived from corpora (data-driven or corpus-based WSD). Any of a variety of association methods is used to determine the best match between the current context and one of these sources of information, in order to assign a sense to each word occurrence (Ide & Véronis 98).

The context is often divided into microcontext and topical context. The microcontext generally means a context of a few words up to an entire sentence. Early findings (Kaplan 50) suggest that ± 2 word contexts are highly reliable, and that even ± 1 contexts are reliable in as much as 8 out of 10 cases. In the microcontext it is also recognized that the distance to the keyword, the collocations as well as the syntactic relations are significant for local word sense disambiguation. Topical context usually means a window of several sentences or more. While local context can account for most of the ambiguities, topical context often can improve the result (Lindén 05).

2 Word Spaces and Random Indexing

Word space models, most notably Latent Semantic Analysis/Indexing, enjoy considerable attention in current research on computational semantics. Since its introduction in 1990 it has more or less spawned an entire research field with a wide range of word space models as a result, and numerous publications reporting exceptional results in many different tasks, such as information retrieval, various semantic knowledge tests (for example TOEFL[1]), text categorization and also word sense disambiguation.

The general idea behind word space models is to use statistics on word distributions in order to generate a high-dimensional vector space.

[1] Test of English as a Foreign Language

In this vector space the words are represented by context vectors whose relative directions are assumed to indicate semantic similarity. The basis of this assumption is the *distributional hypothesis* (Harris 85), according to which words that occur in similar contexts also tend to have similar properties (meanings/functions). From this follows that if we repeatedly observe two words in the same (or very similar) contexts, then it's not too far fetched to assume that they also mean similar things.

In these experiments with word sense disambiguation we have used the Random Indexing (Kanerva *et al.* 00; Sahlgren 05) word space approach, which presents an efficient, scalable and inherently incremental alternative to standard word space methods. As an alternative to LSA-like models that first construct a huge co-occurrence matrix and then use a separate dimension reduction phase, Random Indexing instead accumulates context vectors on-the-fly based on the occurrence of words (tokens) in contexts, without a specific need of a separate dimension reduction phase. This technique can readily be used with any type of linguistic context and can be used to index using a more traditional bag-of-tokens approach as well as using a sliding context window capturing sequential relations between tokens. These tokens can be the word simply represented by its lexical string as well as its lemma, or more elaborate approaches utilizing tagging, chunking, parsing or other linguistic units can be employed.

The construction of context vectors using Random Indexing is perhaps easiest described as a two-step operation (Sahlgren 05). First, each context (e.g. each document, paragraph, word etc) in the data is assigned a unique and randomly generated label. These labels can be viewed as sparse, high-dimensional, and ternary vectors. This means that their dimensionality (d) usually is chosen to be in the range of a couple of hundred up to several thousands, depending of the size and redundancy of the data, and that they consist of a very small number (usually about 1-2%) of randomly distributed +1s and -1s, with the rest of the elements of the vectors set to 0.

Next, the actual context vectors are produced by scanning through the text and each time a token w occurs in a context (e.g. in a document or paragraph, or within a sliding context window), that context's d-dimensional random label is added to the context vector for the token w. Thus, when using a sliding context window, all tokens that appear within the context window contribute (to some degree) with its random label to w's context vector. Words are in this way effectively represented by d-dimensional context vectors that are the sum of the random labels of the co-occurring words.

In practice the random labels are usually represented in more efficient ways than extremely sparse vectors and are generated on-the-fly during the context vector indexing whenever a never before seen token is detected in the context. When using a sliding context window it is also common to use some kind of distance weighting in order to give more weight to tokens closer in context.

3 The Task at Hand

The task chosen for these experiments concerns word sense disambiguation, in our case the construction of a computer program capable of discriminating three different senses of the Swedish word form "resa", one *noun* sense and two *verb* senses, exemplified in the following sentence:

1 Hon vill göra en *resa*. [noun]
She wants to make a *journey*.
2 Hon vill *resa* till USA. [verb1]
She wants to *travel* to USA.
3 Hon vill *resa* en staty. [verb2]
She wants to *raise* a statue.

Reflexive uses of the verb "resa" meaning "rise, stand up" are considered instances of the third sense.

Extending the principles behind the distributional hypothesis and Random Indexing to the field of word sense disambiguation we can, as well as assuming that different words in similar contexts mean similar things, also assume that the same word in different contexts likewise means different things. The hypothesis here is therefore that if we model the different senses by the co-occurrence of "concepts", here represented by context vectors produced by means of Random Indexing, then we should not only be able distinguish the different senses, but also to some extent overcome the problem of sparse data that

here would hamper a traditional Naïve Bayesian approach.

4 Data and Baselines

As "training data" for the Random Indexing step approximately 900.000 words from the Swedish Parole corpus (Gellerstam *et al.* 00) were used together with the approximately 90.000 words from the WSD training set and 20.000 words from the test set; both latter slices taken from the Stockholm-Umeå Corpus, SUC (Ejerhed *et al.* 92), of Swedish texts. The WSD training and test sets where sense tagged, each with one of the three different senses for the word in question, by hand by two persons and then compared, showing basically no conflicting tags to resolve. The tagging resulted in 108 training examples with the following distribution:

sense 1 : 45 instances
sense 2 : 43 instances
sense 3 : 20 instances

The testing data set was similarly annotated in order to be able to automate the scoring of the results. This resulted in 25 test instances with the following distribution:

sense 1 : 7 instances
sense 2 : 7 instances
sense 3 : 11 instances

At this point we can easily deduce two simple baselines to compare our results to. One oft-used baseline is to randomly choose one of the possible alternatives in each instance. Since we in each instance have exactly three senses to choose from this gives us a baseline of 33% correct.

Having tagged the WSD training data we can also inspect the frequency of each of the three senses, and note that sense 1 is by a margin the most common. A promising baseline would thus be to always assign sense 1 to each instance in the WSD test data. However, after tagging of the test data we can establish that this only gives us a baseline of 28%, which is less than random. We are of course aware of that this discrepancy in sense frequency probably is due to the relative smallness of our training and test sets, which increases the risk of unbalanced as well as sparse data.

5 Naïve Bayes

In order to be able to judge how well the Random Indexing approach fares we opted to apply a Naïve Bayes classifier (Mitchell 97) to the same training and data sets for comparison.

We experimented with context windows for the classifier of zero and up to ten words before and/or after the target word "resa", in several different permutations. These permutations were run on lemmatized only as well as lemmatized and PoS-tagged data. We also tried not letting context windows cross sentence boundaries. Furthermore, all combinations of window sizes and data were run with neither normalization nor smoothing, with Lidstone smoothing (Lidstone 20) using several different lambda values, as well as giving less weight to words further away in context.

The best results were obtained with lemmatization only on a context window of ten words before and three words after the target word using distance weighting. Distance weighting was performed by applying a linearly decreasing weight, and the lambda value giving the best result was the Jeffreys-Perks value of 0.5 in added frequency. Also, sentence boundaries were not crossed. Using these settings we achieved a maximum accuracy of 56%.

6 Experimental Set-Up

The three data sets used in the experiments where first transformed into running text by stripping all tag data, and then lemmatized with the Granska tagger (Domeij *et al.* 00) in order to guarantee uniform lemmatization. These sequences of lemmas were then fed into the JavaSDM package (Hassel 04), which is a Java class package for working with Random Indexing that produces a context vector for each word (token) by adding up the random labels of the words in a distance weighted context window of desired size. Apart from a four-fold variation on the seed used, the relevant settings used in JavaSDM throughout these experiments where:

dimensionality = 1000
random_degree = 8
left_window_size = 4
right_window_size = 4
weighting_scheme = moj.ri.weighting.MangesWS
unary_labels = false
document_labels = false
granska = lemmatize

The seeds used for these particular experiments were 710225, 751128, 666 and 777. These seeds are internally in JavaSDM combined with the lexical string of the indexed token (here in the form of a words lemma) in order to guarantee reproducibility.

Two basic approaches where tried in these experiments. The first approach creates a context vector for each training example in a way similar to the way JavaSDM constructs context vectors for words. This means that we here have a distance weighted context window spanning backwards as well as in front of the target word adding up the context vectors for each word found in the context window. The second approach simply adds up all the context vectors, for all training examples, for each sense (no weighting) giving us a context vector per sense - a sense model.

Having these context vectors, and by identically constructing context vectors for each instance of "resa" in the test data, we can now compare each test instance against best matching sense model as well as best matching training example (which's sense we can assign to the test instance). This comparison can be done using any of a wide range of possible vector similarity measures, in our experiments we have used the *cosine* of the angles between the vectors. Using this measure the closest match for each test instance was chosen as the correct sense corresponding to each of the two approaches.

7 Results

The two approaches showcased a wide range in accuracy depending mainly on the size of the context window, spanning from 80% down to 40%, still beating the better baseline by an inch. When taking the mean accuracy over the four tested seeds for each variant, at each tested size of the context window, we can plot a graph to visualize different traits in the four variants.

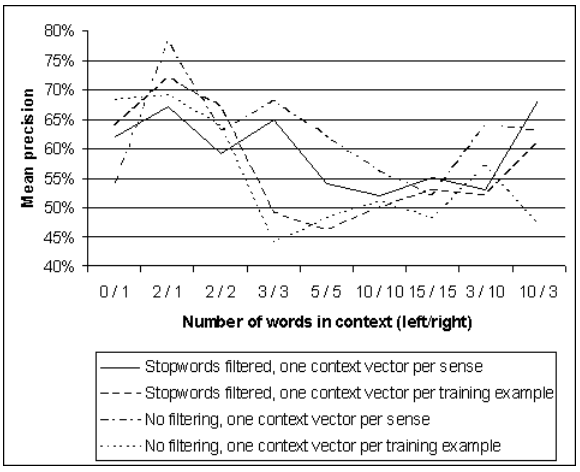

Figure 1: Mean precision over four different seeds for each variant.

In figure 1 we can clearly see a distinct difference between using one context vector per sense or one per training instance in narrow to mid-sized context windows as well as a difference between using stopwords in wide context windows mainly spanning before or after the word being classified. All four variants display a varying demand on a backward-looking context and peak at a context of two words before and one word after. The interesting part is that while using one context vector per training example proves to be particularly unfavourable in mid-sized contexts, this is not the case when using one context vector per sense. This pattern repeats itself regardless if we use stopword filtering or not. On the other hand, while stopword filtering seems to be the way to go when using a context window mainly spanning before the instance being disambiguated, this is clearly not the case in a mainly forward-looking context. However, in doing these observations we must be aware that the three context vectors that are per sense also include, or rather represent, the same amount of information as all the combined training example context vectors per sense. Taking this into account, it is also not so surprising that the main contender is one of the sense model approaches. As we can see in figure 1 the best results were obtained using a context window spanning two words before and one word after the instance of "resa" being classified.

Because of an inherent property of the d-dimensional vectors representing the random labels making them nearly orthogonal, we can

approximate orthogonality simply by choosing random directions in the high-dimensional space. This means that if we collect the context vectors we produce with Random Indexing in a matrix, this matrix will be an approximation of the standard co-occurrence matrix in the sense that their corresponding rows are similar or dissimilar to the same degree. In this way, we can achieve the same dimensional reduction as is done in LSA by the use of SVD: transforming the original co-occurrence counts into a much smaller and denser representation (Sahlgren 05). A key factor in proving the theory to hold in practice is thus the stability in the results over different random projections, here represented by the four different seeds.

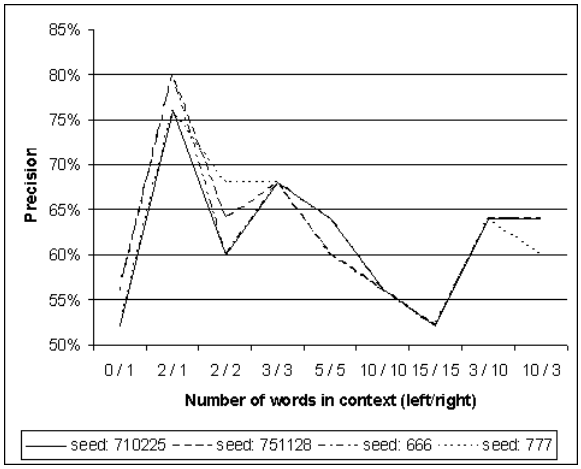

Figure 2: Variation in precision over the four seeds for the best combination, *No stopword filtering, one context vector per sense.*

As we can see in figure 2 the four seeds to a great extent plot against the same lines, with two seeds reaching a maximum of 80% followed closely by the other two at 76%. This can be compared to the above mentioned WSD experiments using a Naïve Bayes classifier on the same training and test sets that reached a top performance of 56%, a result the Random Indexing approach beats hands down. As in the case with the Naïve Bayes approach, words closer in context tend to weigh more in discriminating the different senses of "resa". Other words and their respective senses can of course, depending on differing syntactic and lexical "constraints", display other patterns. This also, naturally, applies to stopword filtering. Using one context vector per sense, rather than one per training example, seems to be a generally good idea since this generates more information dense context vectors.

8 Conclusions and Future Work

We have applied a word co-occurrence based method called Random Indexing to word sense disambiguation for Swedish, modeling the different senses by the co-occurrence of "concepts". The Random Indexing method faired well compared to a standard Naïve Bayes package reaching a maximum accuracy of 80%. As in the case with Naïve Bayes approach, words closer in context tend to weigh more in discriminating the different senses of "resa". The most favourable context window size proved to be two words before and one word after the word being disambiguated, indicating a local ambiguity. Also, stopword filtering proved to remove important syntactic clues in such narrow contexts. One possible explanation for preferring a short context window in this case could be that (mainly) sense 1 and 2 share the same domain, travelling. A wider context window will result in a likewise higher degree of shared co-occurring words. For other ambiguous words, different distance relations may however be more efficient.

Apart from the obvious studies on more words and their corresponding senses, there is also a need for studying how different parameter settings affect the quality of the sense models. One obvious example is of course the dimensionality d, another such property is the size of the context window during the Random Indexing phase, i.e. when building the initial context vector for each word/token. Throughout these experiments we used for the Random Indexing phase a sliding context window spanning four words (lemmas) before and four after the current token. An interesting thought is how it would affect the results in figure 1 if we also vary the size of the context window in this phase.

9 Acknowledgement

We would like to express gratitude to Botond Pakucs of the Centre for Speech Technology at KTH who participated in the experiments involving the Naïve Bayes classifier, and who also took part in the initial sense annotation.

References

(Domeij *et al.* 00) Rickard Domeij, Ola Knutsson, Johan Carlberger, and Viggo Kann. Granska - An efficient hybrid system for Swedish grammar checking. In *Proceedings of NoDaLiDa'99 - 12th Nordic Conference on Computational Linguistics*, 2000.

(Ejerhed *et al.* 92) Eva Ejerhed, Gunnel Källgren, Ola Wennstedt, and Magnus Åström. *SUC - The Stockholm-Umeå Corpus*, version 1.0 (suc 1.0). CD-ROM produced by the Dept of Linguistics, University of Stockholm and the Dept of Linguistics, University of Umeå. ISBN 91-7191-348-3, 1992.

(Gellerstam *et al.* 00) Martin Gellerstam, Yvonne Cederholm, and Torgny Rasmark. The bank of swedish. In *In the proceedings of Second International Conference on Language Resources and Evaluation. LREC-2000*, pages 329–333, Athens, Greece, 2000.

(Harris 85) Zelig Harris. *Distributional Structure*. Oxford University Press, New York, 1985.

(Hassel 04) Martin Hassel. JavaSDM - A Java package for working with Random Indexing and Granska, 2004. http://www.nada.kth.se/~xmartin/java/JavaSDM/.

(Ide & Véronis 98) Nancy Ide and Jean Véronis. Word Sense Disambiguation: The State of the Art. *Computational Linguistics*, 24(1), 1998.

(Kanerva *et al.* 00) Pentti Kanerva, Jan Kristoferson, and Anders Holst. Random Indexing of text samples for Latent Semantic Analysis. In L.R. Gleitman and A.K. Josh, editors, *Proceedings 22nd Annual Conference of the Cognitive Science Society*, Pennsylvania, August 2000.

(Kaplan 50) Abraham Kaplan. An experimental study of ambiguity and context. *Santa Monica: The RAND Corporation. Repr. in Mechanical Translation 2 (1955)*, pages 39–46, 1950.

(Lidstone 20) George James Lidstone. *Note on the general case of the Bayes-Laplace formula for inductive or a posteriori probabilities. Transactions of the Faculty of Actuaries*, volume 8. Brown University Press, Providence R.I., 1920.

(Lindén 05) Krister Lindén. *Word Sense Discovery and Disambiguation*. PhD dissertation, University of Helsinki, Department of General Linguistics, June 2005.

(Mitchell 97) Tom Mitchell. *Machine Learning*. McGraw-Hill, 1997.

(Sahlgren 05) Magnus Sahlgren. An Introduction to Random Indexing. In *Methods and Applications of Semantic Indexing Workshop at the 7th International Conference on Terminology and Knowledge Engineering, TKE 2005)*, Copenhagen, Denmark, August 16 2005.

(Wilks & Stevenson 96) Yorick Wilks and Mark Stevenson. The grammar of sense: Is word sense tagging much more than part-of-speech tagging? Technical report, University of Sheffield, Sheffield, UK, 1996.