Word Sense Disambiguation Using Random Indexing

Márton Miháltz

MorphoLogic
Orbánhegyi út 5., Budapest, H-1126, Hungary
mihaltz@morphologic.hu

Abstract

This paper presents the results of an experiment to apply a novel semantic representational formalism called Random Indexing for the supervised word sense disambiguation of English words. Random Indexing uses high-dimensional sparse vectors with random patterns modeling neural activation patterns in the brain to represent linguistic information. The presented learning and disambiguating method was trained and tested using manually sense-tagged corpora available from Senseval. The results are evaluated and compared to previous works using the same corpora, and the possible lacks and weaknesses of Random Indexing are pointed out both in general, both for the purpose of word sense disambiguation.

Introduction

In this paper, we describe how we have used a technique called Random Indexing .for the representation of word meaning in the supervised word sense disambiguation (WSD) of English content words. For training and testing, we have used openly available sense-tagged corpora from Senseval[1] (Edmonds & Kilgariff, 2003).

The paper is organized as follows: in the next section, we present the theory behind Random Indexing, its recent applications and its adaptation for supervised WSD. We then present the results of applying this method for the disambiguation of two polysemous English nouns, *line* and *party*, utilizing several kinds of contextual information. This is done first by using the original concept of RI, and then by using features known to be useful in the WSD literature. In the next sections, we point out the possible problems with this method, and show the results of an experiment that underline the weaknesses of Random Indexing itself.

Random Indexing

Random Indexing (RI) is a vector-based semantic representation model comparable to such well-known formalisms as the Hyperspace Analogue to Language (HAL) (Lund, Burgess & Atchley, 1995) or Latent Semantic Analysis (LSA) (Landauer & Dumais, 1997). However, Random Indexing differs from the latter one in that it does not require the computationally and memory-intensive process of singular value decomposition (Sahlgren, 2001; Sahlgren, 2002). The idea behind Random Indexing is that the meaning of a given word is determined by its distribution, or the kinds of contexts it appears in. Its meaning can therefore be modeled by an appropriate union of its contexts, where contexts are obtained by looking up the word's occurrences in a large corpus.

In the first step of the algorithm, so-called *random labels* are assigned to word types in a corpus. These are high-dimensional sparse vectors, with a few randomly chosen coordinates turned to +1 or –1, while the rest are set to 0. Learning the meaning representation of a given word is accomplished by formulating context windows around its occurrences and adding up the label vectors of the words in the windows. This is repeated for all occurrences, and the representations of contexts for each occurrence of the target word in a corpus are added up. The context words in the window are weighted in each summation, with the weights reflecting distance from the focus word. Words further from the focus word have smaller weights than the closer ones, which serves as a rough model of the syntactic relationships the focus word participates in.

Sahlgren (2001) used a dimension of 1,800 for the vectors. Random labels were defined by turning 4 randomly selected coordinates to +1 and 4 others to –1, while the rest were set to 0. Training took place on a 10-million word balanced corpus of English by sliding a 3+3 word window (3 context words before and after each word) over each token. The word forms were stemmed by a morphological analyzer before training. The label vectors in the context windows were weighted with an exponential weight function.

This way, 1800-dimension semantic vector representations were formed for each corpus token, which include traces of all the narrow contexts the words appeared in in the corpus. Sahlgren (2001) used the acquired vector representations to automatically obtain semantically similar words for given terms. This was accomplished by retrieving words whose representation vectors were closest to the representation of the query word in the vectors space, where closeness between vectors was determined by the cosine similarity metric.

The representation method was evaluated with the standard TOEFL synonym test, where the computer has to pick the synonym of given words from a lists of possible choices. Random Indexing produced a best score of 68.1% with optimal parameter settings. LSA is known to produce 64.4%, while human (non-English) speakers average 64.5% (Sahlgren, 2001).

There are several advantages of using such a vector-space representation. First, the method allows for a simple and effective training procedure of word meanings, without having to estimate probabilities based on counts from large bodies of text. Second, it provides for a unified treatment of linguistic information, where the different words, contexts and senses are all represented in the same vector-space. And third, since the representation itself doesn't explicitly tell what the meanings of words are, but rather tells what other meanings they are related to (Sahlgren, 2002), the method seems suitable for the task of word sense disambiguation.

[1] http://www.senseval.org/

Word Sense Disambiguation with Random Indexing

We used Random Indexing to gain the vector-space representations of word meanings for the purpose of supervised word sense disambiguation. In our adaptation of Random Indexing, we used the algorithm to gain vector representations not of words, but of the different senses of polysemous words.

We used sense-tagged corpora available from the Senseval Project for training the meanings and testing the disambiguation of the two polysemous nouns *line* and *party*. The training data for *party* was produced by voluntary contributors on the Internet in the *Open Mind Word Expert* (OMWE) Project (Mihalcea & Chklovski, 2002). The sense inventory used for the semantic annotation of both corpora was WordNet (Miller et al, 1990). The *line* corpus describes only the 6 most frequently appearing senses of the noun *line* out of the many more possible in WordNet. The original OMWE *party* corpus covers 5 different senses for *party,* but we decided to use only the 4 most frequent, since the 5th sense had only 8 instances, which proved to be insufficient for training. The figures for the two corpora are depicted in Table 1.

		Number of instances	
Word	Senses	(total)	(most frequent sense)
line	6	4,146	2,217 (53%)
party	4	623	262 (42%)

Table 1: Figures for the two training corpora

The corpora were available part-of-speech tagged from Senseval, and we used our own morphological analyzer to derive the base forms (stems) of the corpus tokens.

For the representation of the vectors, we used a dimension of 1,800. The random label vectors were formed by setting 4 randomly selected coordinates to +1 and 4 others to –1. In our first experiment, we used a context window size of 3+3 (3 words preceding and following each instance of the focus word), and used the same exponentially decreasing weight function as Sahlgren (2001) (Fig. 1.)

[(0.25 0.5 1) 0 (1 0.5 0.25)]

Figure 1: Word weights used in the summation of the context windows

After training the representation of the different senses, the vectors were used for disambiguation. From each instance to be disambiguated, we formed a context vector the same way as during training, then compared this context vector to the vectors representing the different senses of the ambiguous word. The sense was returned that was most similar using the cosine similarity function.

Results and Evaluation

We used 10-fold (stratified) cross-validation on the training corpora with the above disambiguation method, and assigned recall (ratio of disambiguated items to all items) and precision (ratio of correctly disambiguated items to disambiguated items) scores to rate the performance (Table 2).

Word	Precision	Recall
line	50.38%	100%
party	43.05%	100%

Table 2: Disambiguation results using all words in 3+3-context window

The precision score for *line* did not reach the most-frequent-sense baseline, while for *party* it barely exceeded it. From the results of this preliminary experiment, it was obvious that Random Indexing in its original form was not suitable for our word sense disambiguation task. For this reason, we decided to experiment with more sophisticated types of contextual features than just the word stems in the narrow context windows.

Leacock, Miller & Chodorow (1998) used the same training corpus for the noun *line* with a Naive Bayes classifier for disambiguation. They used two different groups of features. The *global feature* consisted of the bag-of-words of all open-class words found in the whole context (the sentence containing the ambiguous word plus the sentences proceeding and following it). Using this feature captures the topical information associated with an instance. The three different *local features* are considered in the sentence containing the ambiguous word only: stems of open-class words in the 3+3 window, stems of closed-class words in the 2+2 window, and POS-tags in the 2+2-sized window surrounding the ambiguous word. These local features capture certain collocational and more syntactical properties of the instances. Using only the topical feature, Leacock, Miller & Chodorow (1998) obtained 78% precision, with only the three local features, 67% precision on the line corpus (about 40% of all instances was set aside for testing). By combining all the features, they reached a precision of 84%.

Besides experimenting with the four different kinds of features proposed by Leacock, Miller & Chodorow (1998), we also decided to test different window sizes: 3+3 and 2+2 words for the local open-class words, and 2+2 and 1+1-word windows for the local close-class words and POS-tags. Finally, we also experimented with using a constant window weight function (no weighting depending on relative position) in addition to the original distance-dependent one.

We trained the Random Indexing representation of the four different kinds of features separately, by using separate label and context vectors for each feature. We then calculated precision and recall by 10-fold cross-validation on the two corpora separately for the features. Results are shown in Table 3 (with bold highlighting showing the best values in the different groups of results).

Feature	Global	Local open-class words				Local closed-class words			Part-of-speech tags		
Window	open-class	3+3		2+2		2+2		1+1	2+2		1+1
Weight fn..	words	decr.	const.	decr.	const.	decr.	const.	const.	decr.	const.	const.
line											
recall	100%	96%	96%	85%	85%	77%	77%	66%	100%	100%	99%
precision	**66,1%**	57,3%	58,7%	58,2%	**59,2%**	48,6%	49,3%	**52,9%**	34,2%	**39,2%**	27,1%
party											
recall	100%	100%	100%	93%	93%	90%	90%	68%	100%	100%	99%
precision	**56,5%**	47,0%	**49,9%**	48,2%	48,3%	50,7%	50,9%	**52,9%**	43,5%	**46,3%**	41,4%

Table 3: Precision and recall using different contextual features, window sizes and window weights on the two corpora

Best individual results for precision among the different features were reached by using the global feature, in both cases. Using this feature alone proved to be significantly better already than the original Random Indexing implementation.

The reason for the word *line* performing better with this feature might be the greater number (4,100 vs. 600) and greater detail (average 51 words vs. average 24 words per instance) of its training instances compared to the OMWE *party* corpus.

About the same difference in precision between the two different items is visible when looking at the local open-class word feature. This might also be explained by the difference in the size and elaboration of the two corpora. However, the local closed-class word and POS-tag features were just as good or better for the less-elaborate *party* corpus. The reason for this might be that the different senses of *party* might be discriminated better using syntactic kinds of features alone than the senses of *line.*

When looking at the different window weights for the local features, the constant weight function definitely obtains better results than the decreasing one. Window sizes show a clear picture for the local closed-class and POS-features: 1+1 and 2+2 sizes are better respectively for both words.

Evaluating the Stability of the Representation

In the last experiment, we wanted to evaluate the stability of the Random Indexing representation. We wanted to see how much the random factor in the generation of the label vectors affects precision of the disambiguation. To test this, we performed 10-fold cross validation on the two corpora using only the global features, repeated 10 times with new randomly generated label vectors every time. Table 4 shows the results of the different runs.

It can be seen from Table 4 that the results on the same dataset. using the same feature can be very different with different label vectors sets. A difference of up to 18% can occur between the best and the worst cases in 10 consecutive runs.

Run	*line*	*party*
1.	69.468%	59.016%
2.	69.468%	59.016%
3.	*71.207%*	*52.459%*
4.	70.144%	57.377%
5.	70.434%	*70.491%*
6.	70.917%	63.934%
7.	70.144%	63.934%
8.	70.531%	59.016%
9.	*68.695%*	59.016%
10.	70.628%	65.573%
Average:	70.164%	60.983%
Standard dev.:	0.761%	5.056%
Δ(Max, Min)	2.512%	**18.032%**

Table 4: Precision results for different runs of the RI WSD algorithm on the same datasets

Discussion

There are several different problems arising from adapting Random Indexing for the word sense disambiguation problem. The first, and most severe problem is the instability of the representation itself: two different runs using the same features and the same training/testing data may produce differences in precision scores of up to 18%. Such a high level of noise introduced by the random factor alone is unacceptable. Using a representation method that does not always provide optimal results makes the evaluation of results almost impossible.

A solution to this problem might be to interfere with the sole randomness of the label generation by enforcing certain constraints on the random coordinate selection that ensure optimal random label sets every time. A more complete understanding of the mathematical principles underlying Random Indexing should be necessary for this.

The second problem arises from using distinct kinds of features and treating them separately during training and testing. This problem surfaces when we try to compare our results to previous works using the same set of features and training corpora. The average 70% precision (from 10 different runs with different label sets) obtained by using the global (topical) feature is comparable to Leacock, Miller & Chodorow's (1998) result of 78% precision (especially considering that results could further improve by finding an optimal representation and tuning other parameters). However, there is no way to compare to their result of 67% precision

when using all the local features together, since there is no straightforward way to combine the different results obtained from disambiguating with the different features into one decision. A solution to this problem might be to construct a voting scheme for the different decisions, perhaps introducing weighting for the different features. Another solution would be to treat the different vectors representing the different features together in a higher-order space (perhaps as vector-tupples or matrices).

We could see that the window size and window weight function type parameter of the process does not always yield optimal results when using the values from the original implementation of Random Indexing. The third problem therefore is that there could be further parameters, which require tweaking for best performance. These might include the dimension of the vectors, the number of initial non-zero components in the random labels, and perhaps using other types of vector-space similarity functions.

A further way to improve the representation would be to find a way to represent relative position information in the vectors. Leacock, Miller & Chodorow (1998) used local features with regard to their relative position form the ambiguous word. Another open question is whether marking the part-of-speech of the corpus tokens in their lemmas (for example to distinguish the verbal and nominal occurences of the word *bark*) would improve the results. Sahlgren's (2001) results show a slight decrease in performance in the TOEFL test when using POS-tags, but for the different requirements of the WSD task the condition might be different. Finally, it might be also worth to consider a more sophisticated learning algorithm that offers more compositionality than just simply summing up the different vectors.

Conclusion

In this paper, we have looked at various ways to adapt the vector-space semantic representation technique of Random Indexing for supervised word sense disambiguation.

In its original implementation (using stems of all words in the 3+3 window surrounding the focus word, with exponentially decreasing weighting) was not suitable for the purpose. A modified version, in which different sorts of linguistic features were treated separately produced better results, with the confidence of the topical feature approaching that of a previous study using a Naive Bayes classifier. If the two major problems RI is facing—the instability of the representation and the present lack of a way to combine the different sources of information–could be overcome, this technique might prove to be an effective and elegant way for representing linguistic information in supervised WSD.

References

Edmonds. P., Kilgarriff. A. (2003). Introduction To The Special Issue On Evaluating Word Sense Disambiguation Systems. Journal of Natural Language Engineering. 8(4). 279—291.

Landauer. T. K., Dumais. S. T. (1997). A Solution to Plato's Problem: The Latent Semantic Analysis Theory of Acquisiton. Induction and Representation of Knowledge. Psychological Review. 104(2). 210—240.

Leacock. C., Miller. G. A., Chodorow, M. (1998). Using Corpus Statistics and WordNet Relations for Sense Identification. Computational Linguistics, Special-Issue-On-Word Sense Disambiguation.

Lund, K., Burgess, C., Atchley, R. (1995). Semantic and associative priming in high dimensional semantic space. In Proceedings of the 17th Annual Conference of the Cognitive Science Society. Hillsdale, NJ: Erlbaum.

Mihalcea, R., Chklovski T. (2002). Building a Sense Tagged Corpus with *Open Mind Word Expert*. In Proceedings of the ACL-02 Workshop on Word Sense Disambiguation: Recent Successes and Future Directions.

Miller, G. A., Beckwith, R., Fellbaum, C., Gross, D. & Miller, K. J. (1990). Introduction to WordNet: an on-line lexical database. International Journal of Lexicography 3 (4) 235 – 244.

Sahlgren. M. (2001). Vector-Based Semantic Analysis: Representing Word Meanings Based on Random Labels. Semantic Knowledge Acquisition and Categorisation Workshop. ESSLLI '01. Helsinki. Finland.

Sahlgren. M. (2002). Towards a Flexible Model of Word Meaning. Paper presented at the AAAI Spring Symposium 2002. March 25-27. Stanford University. Palo Alto. California. USA.