RALI: Automatic weighting of text window distances

Bernard Brosseau-Villeneuve*#, Noriko Kando#, Jian-Yun Nie*
* Université de Montréal, Email: {brosseab, nie}@iro.umontreal.ca
# National Institute of Informatics, Email: {bbrosseau, kando}@nii.ac.jp

Abstract

Systems using text windows to model word contexts have mostly been using fixed-sized windows and uniform weights. The window size is often selected by trial and error to maximize task results. We propose a non-supervised method for selecting weights for each window distance, effectively removing the need to limit window sizes, by maximizing the mutual generation of two sets of samples of the same word. Experiments on Semeval Word Sense Disambiguation tasks showed considerable improvements.

1 Introduction

The meaning of a word can be defined by the words that accompany it in the text. This is the principle often used in previous studies on Word Sense Disambiguation (WSD) (Ide and Véronis, 1998; Navigli, 2009). In general, the accompanying words form a context vector of the target word, or a probability distribution of the context words. For example, under the unigram bag-of-word assumption, this means building $p(x|t) = \frac{count(x,t)}{\sum_{x'} count(x',t)}$, where $count(x,t)$ is the count of co-occurrences of word x with the target word t under a certain criterion. In most studies, x and t should co-occur within a window of up to k words or sentences. The bounds are usually selected as to maximize system performance. Occurrences inside the window usually weight the same without regard to their position. This is counterintuitive. Indeed, a word closer to the target word usually has a greater semantic constraint on the target word than a more distant word. Some studies have also proposed decaying factors to decrease the importance of more distant words in the context vector. However, the decaying functions are defined manually. It is unclear that the functions defined can capture the true impact of the context words on the target word. In this paper, we propose an unsupervised method to automatically learn the optimal weight of a word according to its distance to the target word. The general idea used to determine such weight is that, if we randomly determine two sets of texts containing the target word, the resulting probability distributions for its context words in the two sets should be similar. Therefore, the weights of context words at different distance are determined so as to maximize the mutual generation probabilities of two sets of samples. Experimentation on Semeval-2007 English and Semeval-2010 Japanese lexical sample task data shows that improvements can automatically be attained on simple Naive Bayes (NB) systems in comparison to the best manually selected fixed window system.

The remainder of this paper is organized as follows: example uses of text windows and related work are presented in Section 2. Our method is presented in Section 3. In Section 4 and 5, we show experimental results on English and Japanese WSD. We conclude in Section 6 with discussion and further possible extensions.

2 Uses of text windows

Modeling the distribution of words around one target word has many uses. For instance, the Xu&Croft co-occurrence-based stemmer (Xu and Croft, 1998) uses window co-occurrence statistics to calculate the best equivalence classes for a group of word forms. They suggest using windows of up to 100 words. Another example can be found in WSD systems, where a shorter window is preferred. In Semeval-2007, top performing systems on WSD tasks, such as NUS-ML (Cai et al., 2007), made use of bag-of-word features around the target word. In this case, they found that the best results can be achieved using a window size of 3.

Proceedings of the 5th International Workshop on Semantic Evaluation, ACL 2010, pages 375–378,
Uppsala, Sweden, 15-16 July 2010. ©2010 Association for Computational Linguistics

Both these systems limit the size of their windows for different purposes. The former aims to model the topic of the documents containing the word rather than the word's meaning. The latter limits the size because bag-of-word features further from the target word would not be sufficiently related to its meaning (Ide and Véronis, 1998). We see that because of sparsity issues, there is a compromise between taking few, highly related words, or taking several, lower quality words.

In most current systems, all words in a window are given equal weight, but we can easily understand that the occurrences of words should generally count less as they become farther; they form a long tail that we should use. Previous work proposed using non-linear functions of the distance to model the relation between two words. For instance, improvements can be obtained by using an exponential function (Gao et al., 2002). Yet, there is no evidence that the exponential – with its manually selected parameter – is the best function.

3 Computing weights for distances

In this section, we present our method for choosing how much a word should count according to its distance to the target word. First, for some definitions, let $\mathcal{C}$ be a corpus, W a set of text windows, $c_{W,i,x}$ the count of occurrences of word x at distance i in W, $c_{W,i}$ the sum of these counts, and α_i the weight put on one word at distance i. Then,

$$P_{ML,W}(x) = \frac{\sum_i \alpha_i c_{W,i,x}}{\sum_i \alpha_i c_{W,i}} \tag{1}$$

is the maximum likelihood estimator for x. To counter the zero-probability problem, we apply Dirichlet smoothing with the collection language model as a prior:

$$P_{Dir,W}(x) = \frac{\sum_i \alpha_i c_{W,i,x} + \mu_W P(x|\mathcal{C})}{\sum_i \alpha_i c_{W,i} + \mu_W} \tag{2}$$

The pseudo-count μ_W is found by using Newton's method via leave-one-out estimation. We follow the procedure shown in (Zhai and Lafferty, 2002), but since occurrences have different weights, the log-likelihood is changed to

$$\mathcal{L}_{-1}(\mu|W,\mathcal{C}) = \tag{3}$$
$$\sum_i \sum_{x \in V} \alpha_i c_{W,i,x} \log \frac{\alpha_i c_{W,i,x} - \alpha_i + \mu P(x|\mathcal{C})}{\sum_j \alpha_j c_{W,j} - \alpha_i + \mu}$$

To find the best weights for our model we propose the following:

- Let T be the set of all windows containing the target word. We randomly split this set into two sets A and B.
- We want to find $\alpha^\star$ that maximizes the mutual generation of the two sets, by minimizing their cross-entropy:

$$l(\alpha) = H(P_{ML,A}, P_{Dir,B}) + H(P_{ML,B}, P_{Dir,A}) \tag{4}$$

In other words, we want α_i to represent how much an occurrence at distance i models the context better than the collection language model, whose counts are controlled by the Dirichlet pseudo-count. We hypothesize that target words occurs in limited contexts, and as we get farther from them, the possibilities become greater, resulting in sparse and less related counts.

3.1 Gradient descent

We propose a simple gradient descent minimizing (4) over α. For the following experiments, we used one single curve for all words in a task. We used the mini-batch type of gradient descent: the gradients of a fixed amount of target words are summed, a gradient step is done, and the proces is repeated while cycling the data. The starting state was with all α_i to one, the batch size of 50 and a learning rate of 1. We notice that as the algorithm progress, weights on close distances increase and the farthest decrease. As further distances contribute less and less, middle distances start to decay more and more, until at some point, all distances but the closest start to decrease, heading towards a degenerate solution. We therefore suggest using the observation of several consecutive decreases of all except α_1 as an end criterion. We used 10 consecutive steps for our experiments.

4 Experiments on Semeval-2007 English Lexical Sample

The Semeval workshop holds WSD tasks such as the English Lexical Sample (ELS) (Pradhan et al., 2007). It consists of a selected set of polysemous words, contained within passages where a sense taken from a sense inventory is manually annotated. The task is to create supervised classifiers maximizing accuracy on test data.

Since there are only 50 words and instances are few, we judged there was not enough data to compute weights. Instead, we used the AP Newswire corpus of the TREC collection (CD 1 & 2). Words were stemmed with the Porter stemmer and text windows were grouped for all words. For simplicity and efficiency, windows to the right and to the left were considered independent, and we only kept words with between 30 and 1000 windows. Also, only windows with a size of 100, which was considered big enough without any doubt, were kept. A stop list of the top 10 frequent words was used, but place holders were left in the windows to preserve the distances. Multiple consecutive stop words (ex: "of the") were merged, and the target word, being the same for all samples of a set, was ignored. This results in 32,650 sets containing 5,870,604 windows. In Figure 1, we can see the resulting weight curve.

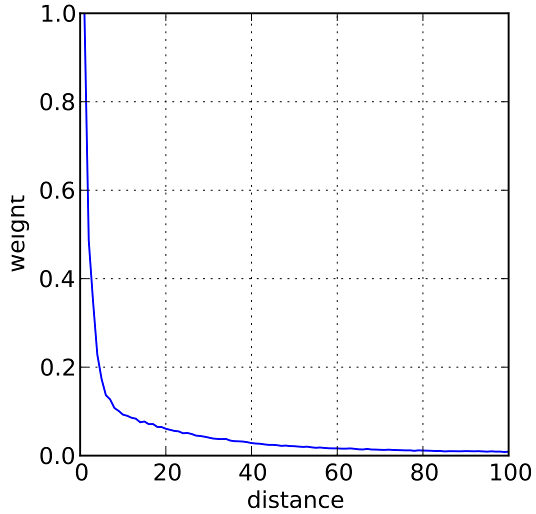

Figure 1: Weight curve for AP Newswire

Since the curve converges, words over the 100th distance were assigned the minimum weight found in the curve. From this we constructed NB models whose class priors used an absolute discounting of 0.5. The collection language model used the concatenation of the AP collection and the Semeval data. As the unstemmed target word is an important feature it was added to the models. It's weight was chosen to be 0.7 by maximizing accuracy on one-held-out cross-validation of the training data. The results are listed in Table 1.

System	Cross-Val (%)	Test set (%)
Prior only	78.66	77.76
Best uniform	85.48	83.28
RALI-2	88.23	86.45

Table 1: WSD accuracy on Semeval-2007 ELC

We used two baselines: most frequent sense (prior only), and the best uniform (except target word) fixed size window found from extensive search on the training data. The best settings were a window of size 4, with a weight of 4.4 on the target word and a Laplace smoothing of 2.9. The improvements seen using our system are substantial, beating most of the systems originally proposed for the task (Pradhan et al., 2007). Out of 15 systems, the best results had accuracies of 89.1*, 89.1*, 88.7, 86.9 and 86.4 (* indicates post-competition submissions). Notice that most were using Support Vector Machine (SVM) with bag-of-word features in a very small window, local collocations and POS tags. In our future work, we will investigate the applications of SVM with our new term weighting scheme.

5 Experiments on Semeval-2010 Japanese WSD

The Semeval-2010 Japanese WSD task (Okumura et al., 2010) consists of 50 polysemous words for which examples were taken from the BC-CWJ tagged corpus. It was manually segmented, tagged, and annotated with senses taken from the Iwanami Kokugo dictionary. The task is identical to the ELS of the previous experiment.

Since the data was again insufficient to compute curves, we used the Mainichi-2005 corpus of NTCIR-8. We tried to reproduce the same kind of segmentation as the training data by using the Chasen parser with UniDic. For the corpus and Semeval data, conjugations (setsuzoku-to, jodô-shi, etc.), particles (all jo-shi), symbols (blanks, kigô, etc.), and numbers were stripped. When a base-form reading was present (for verbs and adjectives), the token was replaced by the Kanjis (chinese characters) in the word writing concatenated with the base-form reading. This treatment is somewhat equivalent to the stemming+stop list of the ELS tasks. The resulting curve can be seen in Figure 2.

The NB models are the same as in the previous experiments. Target words were again added the same way as in the ELS task. The best fixed window model was found to have a window size of 1 with a target word weight of 0.6 and used manual Dirichlet smoothing with a pseudo-count of 110. We submited two systems with the following settings: RALI-1 used manual Dirichlet smoothing and 0.9 for the target word. RALI-2 used auto-

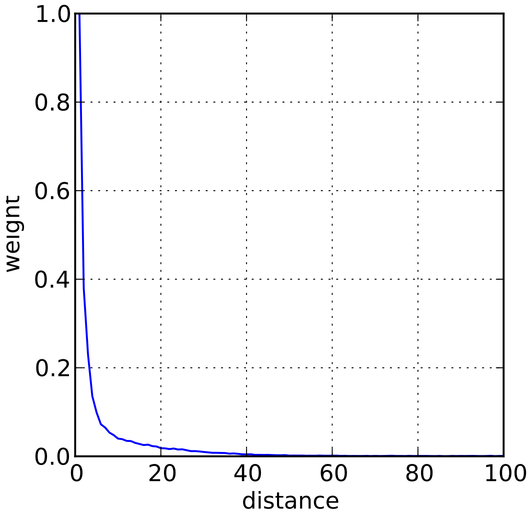

Figure 2: Weight curve for Mainichi Shinbun 2005

matic Dirichlet smoothing and 1.7 for the target word weight. Results are listed in Table 2.

System	Cross-Val (%)	Test set (%)
prior only	75.23	68.96
Best uniform	82.29	76.12
RALI-1	82.77	75.92
RALI-2	83.05	76.36

Table 2: WSD accuracy on Semeval-2010 JWSD

As we can see, the results are not significantly different from the best uniform model. This may be due to differences in the segmentation parameters of our external corpus. Another reason could be that the systems use almost the same weights: the best fixed window had size 1, and the Japanese curve is steeper than the English one.

This steeper curve can be explained by the grammatical structure of the Japanese language. While English can be considered a Subject-Verb-Complement language, Japanese is considered Subject-Complement-Verb. Verbs are mostly found at the end of the sentence, far from their subject, and vice versa. The window distance is therefore less useful in Japanese than in English since it has more non-local dependencies. These results show that the curves work as expected even in different languages.

6 Conclusions

This paper proposed an unsupervised method for finding weights for counts in text windows according to their distance to the target word. Results from the Semeval-2007 English lexical sample showed a substantial improvement in precision. Yet, as we have seen with the Japanese task, window distance is not always a good indicator of word relatedness. Fortunately, we can easily imagine extensions to the current scheme that bins word counts by factors other than word distance. For instance, we could also bin counts by parsing tree distance, sentence distance or POS-tags.

Acknowledgments

The authors would like to thank Florian Boudin and Satoko Fujisawa for helpful comments on this work. This work is partially supported by Japanese MEXT Grant-in-Aid for Scientific Research on Info-plosion (#21013046) and the Japanese MEXT Research Student Scholarship program.

References

Jun Fu Cai, Wee Sun Lee, and Yee Whye Teh. 2007. Nus-ml: improving word sense disambiguation using topic features. In *SemEval '07 Proceedings*, pages 249–252, Morristown, NJ, USA. Association for Computational Linguistics.

Jianfeng Gao, Ming Zhou, Jian-Yun Nie, Hongzhao He, and Weijun Chen. 2002. Resolving query translation ambiguity using a decaying co-occurrence model and syntactic dependence relations. In *SIGIR '02 Proceedings*, pages 183–190, New York, NY, USA. ACM.

Nancy Ide and Jean Véronis. 1998. Introduction to the special issue on word sense disambiguation: the state of the art. *Comput. Linguist.*, 24(1):2–40.

Roberto Navigli. 2009. Word sense disambiguation: A survey. *ACM Comput. Surv.*, 41(2):1–69.

Manabu Okumura, Kiyoaki Shirai, Kanako Komiya, and Hikaru Yokono. 2010. Semeval-2010 task: Japanese wsd. In *SemEval '10 Proceedings*. Association for Computational Linguistics.

Sameer S. Pradhan, Edward Loper, Dmitriy Dligach, and Martha Palmer. 2007. Semeval-2007 task 17: English lexical sample, srl and all words. In *SemEval '07 Proceedings*, pages 87–92, Morristown, NJ, USA. Association for Computational Linguistics.

Jinxi Xu and W. Bruce Croft. 1998. Corpus-based stemming using cooccurrence of word variants. *ACM Trans. Inf. Syst.*, 16(1):61–81.

ChengXiang Zhai and John Lafferty. 2002. Two-stage language models for information retrieval. In *SIGIR '02 Proceedings*, pages 49–56, New York, NY, USA. ACM.