Towards an optimal weighting of context words based on distance

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

Abstract

Word Sense Disambiguation (WSD) often relies on a context model or vector constructed from the words that co-occur with the target word within the same text windows. In most cases, a fixed-sized window is used, which is determined by trial and error. In addition, words within the same window are weighted uniformly regardless to their distance to the target word. Intuitively, it seems more reasonable to assign a stronger weight to context words closer to the target word. However, it is difficult to manually define the optimal weighting function based on distance. In this paper, we propose a unsupervised method for determining the optimal weights for context words according to their distance. The general idea is that the optimal weights should maximize the similarity of two context models of the target word generated from two random samples. This principle is applied to both English and Japanese. The context models using the resulting weights are used in WSD tasks on Semeval data. Our experimental results showed that substantial improvements in WSD accuracy can be obtained using the automatically defined weighting schema.

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-words 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 in an ad-hoc fashion to maximize system performance. Occurrences inside the window often weight the same without regard to their position. This is counterintuitive. Indeed, a word closer to the target word generally has a greater semantic constraint on the target word than a more distant word. It is however difficult to define the optimal weighting function manually. To get around this, some systems add positional features for very close words. In information retrieval, to model the strength of word relations, some studies have proposed non-uniform weighting methods of context words, which decrease the importance of more distant words in the context vector. However, the weighting functions are defined manually. It is unclear that these functions can best capture the 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 principle used to determine such weight is that, if we randomly determine two sets of windows containing the target word from the same corpus, the meaning – or mixture of meanings for polysemic words – of the target word in the two sets should be similar. As the context model – a probability distribution for the context words – determines the meaning of the target word, the context models generated from the two sets should also be similar. The weights of context words at different distance are therefore de-

Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010), pages 107–115,
Beijing, August 2010

termined so as to maximize the similarity of context models generated from the two sets of samples. In this paper, we propose a gradient descent method to find the optimal weights. We will see that the optimal weighting functions are different from those used in previous studies. Experimentation on Semeval-2007 English and Semeval-2010 Japanese lexical sample task data shows that improvements can be attained using the resulting weighting functions on simple Naïve Bayes (NB) systems in comparison to manually selected functions. This result validates the general principle we propose in this paper.

The remainder of this paper is organized as follows: typical uses of text windows and related work are presented in Section 2. Our method is presented in Section 3. In Section 4 to 6, we show experimental results on English and Japanese WSD. We conclude in Section 7 with discussion and further possible extensions.

2 Uses of text windows

Modeling the distribution of words around one target word, which we call context model, has many uses. For instance, one can use it to define a co-occurrence-based stemmer (Xu and Croft, 1998), which uses window co-occurrence statistics to calculate the best equivalence classes for a group of word forms. In the study of Xu and Croft, they suggest using windows of up to 100 words. Context models are also widely used in WSD. For example, top performing systems on English WSD tasks in Semeval-2007, such as NUS-ML (Cai et al., 2007), all made use of bag-of-words features around the target word. In this case, they found that the best results can be achieved using a window size of 3.

Both systems limit the size of their windows for different purposes. The former uses a large size in order to model the topic of the documents containing the word rather than the word's meaning. The latter would limit the size because bag-of-words features further from the target word would not be sufficiently related to its meaning (Ide and Véronis, 1998). We see that there is a compromise between taking fewer, highly related words, or taking more, lower quality words. However, there is no principled way to determine the optimal size of windows. The size is determined by trial and error.

A more questionable aspect in the above systems is that for bag-of-words features, all words in a window are given equal weights. This is counterintuitive. One can easily understand that a context word closer to the target word *generally* imposes a stronger constraint on the meaning of the latter, than a more distant context word. It is then reasonable to define a weighting function that decreases along with distance. Several studies in information retrieval (IR) have proposed such functions to model the strength of dependency between words. For instance, Gao et al. (2002) proposed an exponential decay function to capture the strength of dependency between words. This function turns out to work better than the uniform weighting in the IR experiments.

Song and Bruza (2003) used a fixed-size sliding window to determine word co-occurrences. This is equivalent to define a linear decay function for context words. The context vectors defined this way are used to estimate similarity between words. A use of the resulting similarity in query expansion in IR turned out to be successful (Bai et al., 2005).

In a more recent study, Lv and Zhai (2009) evaluated several kernel functions to determine the weights of context words according to distance, including Gaussian kernel, cosine kernel, and so on. As for the exponential and linear decaying functions, all these kernel functions have fixed shapes, which are determined manually.

Notice that the above functions have only been tested in IR experiments. It is not clear how these functions perform in WSD. More importantly, all the previous studies have investigated only a limited number of weighting functions for context words. Although some improvements using these functions have been observed in IR, it is not clear whether the functions can best capture the true impact of the context words on the meaning of the target word. Although the proposed functions comply with the general principle that closer words are more important than more distant words, no principled way has been proposed to determine the particular shape of the function for different languages and collections.

In this paper, we argue that there is indeed a hidden weighting function that best capture the impact of context words, but the function cannot be defined manually. Rather, the best function should be the one that emerges naturally from the data. Therefore, we propose an unsupervised method to discover such a function based on the following principle: the context models for a target word generated from two random samples should be similar. In the next section, we will define in detail how this principle is used.

3 Computing weights for distances

In this section, we present our method for choosing how much a word occurrence should count in the context model according to its distance to the target word. In this study, for simplicity, we assume that all word occurrences at a given distance count equally in the context model. That is, we ignore other features such as POS-tags, which are used in other studies on WSD.

Let $\mathcal{C}$ be a corpus, W a set of text windows for the target word w, $c_{W,i,x}$ 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 occurrence at distance i. Then,

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

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

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

The pseudo-count μ_W can be a constant, or can be found by using Newton's method, maximizing the log likelihood via leave-one-out estimation:

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

The general process, which we call automatic Dirichlet smoothing, is similar to that described in (Zhai and Lafferty, 2002).

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

- 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 similarity of the models obtained from the two sets, by minimizing their mutual cross entropy:

$$l(\alpha) = H(P_{ML,A}, P_{Dir,B}) + H(P_{ML,B}, P_{Dir,A}) \quad (3)$$

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 weighted by the Dirichlet parameter. We hypothesize that target words occur in limited contexts, and as we get farther from them, the possibilities become greater, resulting in sparse and less related counts. Since two different sets of the same word are essentially noisy samples of the same distribution, the weights maximizing their mutual generation probabilities should model this phenomenon.

One may wonder why we do not use a distribution similarity metric such as Kullback–Leibler (KL) divergence or Information Radius (IRad). The reason is that with enough word occurrences (big windows or enough samples), the most similar distributions are found with uniform weights, when all word counts are used. KL divergence is especially problematic as, since it requires smoothing, the weights will converge to the degenerate weights $\alpha = 0$, where only the identical smoothing counts remain. Entropy minimization is therefore needed in the objective function.

To determine the optimal weight of α_i, we propose a simple gradient descent minimizing (3) over α. The following are the necessary derivatives:

$$\frac{\partial l}{\partial \alpha_i} = \frac{\partial H(P_{ML,A}, P_{Dir,B})}{\partial \alpha_i} + \frac{\partial H(P_{ML,B}, P_{Dir,A})}{\partial \alpha_i}$$

$$\frac{\partial H\left(P_{ML,W}, P_{Dir,(T-W)}\right)}{\partial \alpha_i} =$$

$$-\sum_{x\in V}\left[\frac{\partial P_{ML,W}(x)}{\partial\alpha_i}\log P_{Dir,(T-W)}(x)+\frac{\partial P_{Dir,(T-W)}(x)}{\partial\alpha_i}\times\frac{P_{ML,W}(x)}{P_{Dir,(T-W)}(x)}\right]$$

$$\frac{\partial P_{ML,W}(x)}{\partial\alpha_i}=\frac{c_{W,i,x}-P_{ML,W}(x)c_{W,i}}{\sum_j\alpha_j c_{W,j}}$$

$$\frac{\partial P_{Dir,W}(x)}{\partial\alpha_i}=\frac{c_{W,i,x}-P_{Dir,W}(x)c_{W,i}}{\sum_j\alpha_j c_{W,j}+\mu_W}$$

We use stochastic gradient descent: one word is selected randomly, it's gradient is computed, a small gradient step is done and the process is repeated. A pseudo-code of the process can be found in Algorithm 1.

Algorithm 1 LearnWeight($\mathcal{C},\eta,\epsilon$)

$\alpha\leftarrow 1^k$
repeat
  $T\leftarrow${Get windows for next word}
  $(A,B)\leftarrow$RandomPartition(T)
  for W in A,B **do**
    $P_{ML,W}\leftarrow$MakeML(W,α)
    $\mu_W\leftarrow$ComputePseudoCount(W,$\mathcal{C}$)
    $P_{Dir,W}\leftarrow$MakeDir($P_{ML,W},\mu_W,\mathcal{C}$)
  end for
  $grad\leftarrow\ \nabla H(P_{ML,A},P_{Dir,B})\ +\ \nabla H(P_{ML,B},P_{Dir,A})$
  $\alpha\leftarrow\alpha-\eta\frac{grad}{\|grad\|}$
until $\exists\alpha_i<\epsilon$
return $\alpha/\max\{\alpha_i\}$

Now, as the objective function would eventually go towards putting nearly all weight on α_1, we hypothesize that the farthest distances should have a near-zero contribution, and determine the stop criterion as having one weight go under a small threshold. Alternatively, a control set of held out words can be used to observe the progress of the objective function or the gradient length. When more and more weight is put on the few closest positions, the objective function and gradient depends on less counts and will become less stable. This can be used as a stop criterion.

The above weight learning process is applied on an English collection and a Japanese collection with $\eta=\epsilon=0.001$, and $\mu=1000$. In the next sections, we will describe both resulting weighting functions in the context of WSD experiments.

4 Classifiers for supervised WSD tasks

Since we use the same systems for both English and Japanese experiments, we will briefly discuss the used classifiers in this section. In both tasks, the objective is to maximize WSD accuracy on held-out data, given that we have a set of training text passages containing a sense-annotated target word.

The first of our baselines, the *Most Frequent Sense* (MFS) system always selects the most frequent sense in the training set. It gives us a lower bound on system accuracies.

Naïve Bayes (NB) classifiers score classes using the Bayes formula under a feature independence assumption. Let w be the target word in a given window sample to be classified, the scoring formula for sense class S is:

$$Score(w,S)=P(S)P_{Tar}(w|S)^{\lambda_{Tar}}\times\prod_{x_i\in context(w)}P_{Con}(x_i|S)^{\lambda_{Con}\alpha_{dist(x_i)}}$$

where $dist(x_i)$ is the distance between the context word x_i and the target word w. The target word being an informative feature present in all samples, we use it in a target word language model P_{Tar}. The surrounding words are summed in the context model P_{Con} as shown in equation (1). As we can see with the presence of α in the equation, the scoring follows the same weighting scheme as we do when accumulating counts, since the samples to classify follow the same distribution as the training ones. Also, when a language model uses automatic Dirichlet smoothing, the impact of the features against the prior is controlled with the manual parameters λ_{Tar} or λ_{Con}. When a manual smoothing parameter is used, it also handles impact control. Our systems use the following weight functions:

Uniform: $\alpha_i=\mathbf{1}_{1\leq i\leq\delta}$, where δ is a window size and $\mathbf{1}$ the indicator function.

Linear: $\alpha_i=\max\{0,1-(i-1)\delta\}$, where δ is the decay rate.

Exponential: $\alpha_i = e^{-(i-1)\delta}$, where δ is the exponential parameter.

Learned: α_i is the weight learned as shown previously.

The parameters for NB systems are identical for all words of a task and were selected by exhaustive search, maximizing leave-one-out accuracy on the training set. For each language model, we tried Laplace, manual Dirichlet and automatic Dirichlet smoothing.

For the sake of comparison, also we provide a *Support Vector Machine* (SVM) classifier, which produces the best results in Semeval 2007. We used libSVM with a linear kernel, and regularization parameters were selected via grid search maximizing leave-one-out accuracy on the training set. We tested the following windows limits: all words in sample, current sentence, and various fixed window sizes. We used the same features as the NB systems, testing Boolean, raw count, log-of-counts and counts from weight functions representations. Although non-Boolean features had good leave-one-out precision on the training data, since SVM does not employ smoothing, only Boolean features kept good results on test data, so our SVM baseline uses Boolean features.

5 WSD experiments on Semeval-2007 English Lexical Sample

The Semeval workshop holds WSD tasks such as the English Lexical Sample (ELS) (Pradhan et al., 2007). The task is to maximize WSD accuracy on a selected set of polysemous words, 65 verbs and 35 nouns, for which passages were taken from the WSJ Tree corpus. Passages contain a couple of sentences around the target word, which is manually annotated with a sense taken from OntoNotes (Hovy et al., 2006). The sense inventory is quite coarse, with an average of 3.6 senses per word. Instances count are listed in Table 1.

	Train	Test	Total
Verb	8988	2292	11280
Noun	13293	2559	15852
Total	22281	4851	

Table 1: Number of instances in the ELS data

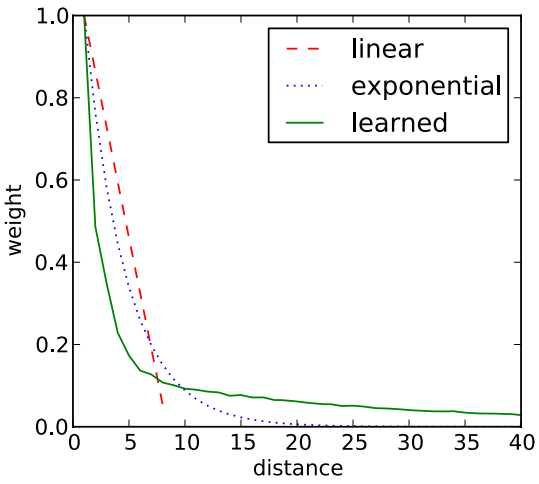

Figure 1: Weight curve for AP88-90

Since there are only 100 target words and instances are limited in the Semeval collection, we do not have sufficient samples to estimate the optimal weights for context words. Therefore, we used the AP88-90 corpus of the TREC collection (CD 1 & 2) in our training process. The AP collection contains 242,918 documents. Since our classifiers use word stems, the collection was also stemmed with the Porter stemmer and sets of windows were built for all word stems. To get near-uniform counts in all distances, only full windows with a size of 100, which was considered big enough without any doubt, were kept. In order to get more samples, windows to the right and to the left were separated. For each target word, we used 1000 windows. A stoplist 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 stem, being the same for all samples of a set, was ignored in the construction of context models. The AP collection results in 32,650 target words containing 5,870,604 windows. The training process described in Section 3 is used to determine the best weights of context words. Figure 1 shows the first 40 elements of the resulting weighting function curve.

As we can see, the curve is neither exponential, linear, or any of the forms used by Lv and Zhai. Its form is rather similar to $x^{-\delta}$, or rather $\log^{-1}(\delta + x)$ minus some constant. The decrease

System	Cross-Val (%)	Test set (%)
MFS	78.66	77.76
Uniform NB	86.04	84.52
SVM	85.53	85.03
Linear NB	86.89	85.71
Exp. NB	87.80	86.23
Learned NB	88.46	86.70

Table 2: WSD accuracy on Semeval-2007 ELC

rate is initially very high and then reduces as it becomes closer to zero. This long tail is not present in any of the previously suggested functions. The large difference between the above optimal weighting function and the functions used in previous studies would indicate that the latter are suboptimal. Also, as we can see, the relation between context words and the target word is mostly gone after a few words. This would motivate the commonly used very small windows when using a uniform weights, since using a bigger window would further widen the gap between the used weight and the optimal ones.

Now for the system settings, the context words were processed the same way as the external corpus. The target word was used without stemming but had the case stripped. The NB systems used the concatenation of the AP collection and the Semeval data for the collection language model. This is motivated by the fact that the Semeval data is not balanced: it contains only a small number of passages containing the target words. This makes words related to them unusually frequent. The class priors used an absolute discounting of 0.5 on class counts. *Uniform NB* uses a window of size 4, a Laplace smoothing of 0.65 on P_{Tar} and an automatic Dirichlet with $\lambda_{Con} = 0.7$ on P_{Con}. *Linear NB* has $\delta = 0.135$, uses a Laplace smoothing of 0.85 on P_{Tar} and an automatic Dirichlet with $\lambda_{Con} = 0.985$ on P_{Con}. *Exp NB* has $\delta = 0.27$, uses a Laplace smoothing of 2.8 on P_{Tar} and an automatic Dirichlet with $\lambda_{Con} = 1.01$ on P_{Con}. The *SVM* system uses a window of size 3. Our system, *Learned NB* uses a Laplace smoothing of 1.075 on P_{Tar}, and an automatic Dirichlet with $\lambda_{Con} = 1.025$ on P_{Con}. The results on WSD are listed in Table 2. WSD accuracy is measured by the proportion of correctly disambiguated words among all the word samples. The cross-validation is performed on the training data with leave-one-out and is shown as a hint of the capacity of the models. A randomization test comparing *Exponential NB* and *Learned NB* gives a p-value of 0.0508, which is quite good considering the extensive trials used to select the exponential parameter in comparison to a single curve computed from a different corpus. This performance is comparable to the current state of the art. It outperforms most of the systems participating in the task (Pradhan et al., 2007). Out of 14 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 previous systems used SVM with additional features such as local collocations, positional word features and POS tags. Our approach only uses bag-of-words in a Naïve Bayes classifier. Therefore, the performance of our method is sub-optimal. With additional features and better classification methods, we can expect that better performance can be obtained. In future work, we will investigate the applications of SVM with our new term weighting scheme, together with additional types of features.

6 WSD experiments on Semeval-2010 Japanese Lexical Sample

The Semeval-2010 Japanese WSD task (Okumura et al., 2010) consists of 50 polysemous words for which examples were taken from the BCCWJ corpus (Maekawa, 2008). It was manually segmented, POS-tagged, and annotated with senses taken from the Iwanami Kokugo dictionary. The selected words have 50 samples for both the training and test set. The task is identical to the ELS of the previous experiment.

Since the data was again insufficient to compute the optimal weighting curve, 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, which nevertheless results in different word segments as the training data. 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

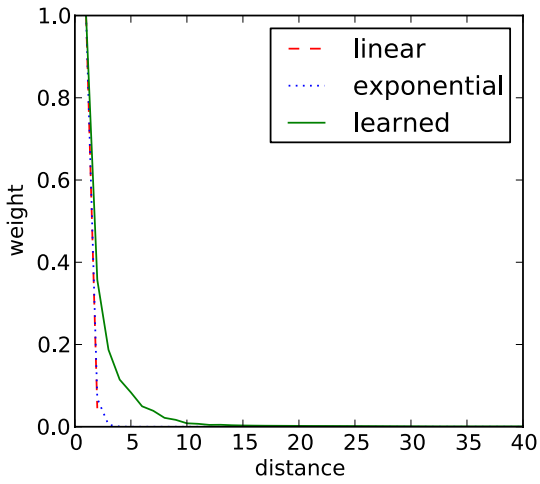

Figure 2: Weight curve for Mainichi 2005

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.

As we can see, the general form of the curve is similar to that of the English collection, but is steeper. This suggests that the meaning of Japanese words can be determined using only the closest context words. Words further than a few positions away have very small impact on the target word. This 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, mostly found at the end of a sentence, can be far apart from their subject, and vice versa. The window distance is therefore less useful to capture the relatedness in Japanese than in English since Japanese has more non-local dependencies.

The Semeval Japanese test data being part of a balanced corpus, untagged occurrences of the target words are plenty, so we can benefit from using the collection-level counts for smoothing. *Uniform NB* uses a window of size 1, manual Dirichlet smoothing of 4 for P_{Tar} and 90 for the P_{Con}. *Linear NB* has $\delta = 0.955$, uses a manual Dirichlet smoothing of 6.25 on P_{Tar} and manual Dirichlet

System	Cross-Val (%)	Test set (%)
MFS	75.23	68.96
SVM	82.55	74.92
Uniform NB	82.47	76.16
Linear NB	82.63	76.48
Exp. NB	82.68	76.44
Learned NB	82.67	76.52

Table 3: WSD accuracy on Semeval-2010 JWSD

smoothing with $\lambda_{Con} = 65$ on P_{Con}. *Exp NB* has $\delta = 2.675$, uses a manual Dirichlet smoothing of 6.5 on P_{Tar} and a manual Dirichlet of 70 on P_{Con}. The *SVM* system uses a window size of 1 and Boolean features. *Learned NB* used a manual Dirichlet smoothing of 4 for P_{Tar} and automatic Dirichlet smoothing with $\lambda_{Con} = 0.6$ for P_{Con}. We believe this smoothing is beneficial only on this system because it uses more words (the long tail), that makes the estimation of the pseudo-count more accurate. Results on WSD are listed in Table 3. As we can see, the difference between the NB models is less substantial than for English. This may be due to differences in the segmentation parameters of our external corpus: we used the human-checked segmentation found in the Semeval data for classification, but used a parser to segment our external corpus for weight learning. We are positive that the Chasen parser with the UniDic dictionary was used to create the initial segmentation in the Semeval data, but there may be differences in versions and the initial segmentation results were further modified manually.

Another reason for the results could be that the systems use almost the same weights: *Uniform NB* and *SVM* both used windows of size 1, and the Japanese curve is steeper than the English one, making the context model account to almost only immediately adjacent words. So, even if our context model contains more context words at larger distances, their weights are very low. This makes all context model quite similar. Nevertheless, we still observe some gain in WSD accuracy. These results show that the curves work as expected even in different languages. However, the weighting curve is strongly language-dependent. It could also be collection-dependent – we will investigate

this aspect in the future, using different collections.

7 Conclusions

The definition of context vector and context model is critical in WSD. In previous studies in IR, decaying weight along with distance within a text window have been proposed. However, the decaying functions are defined manually. Although some of the functions produced better results than the uniform weighting, there is no evidence showing that these functions best capture the impact of the context words on the meaning of the target word. This paper proposed an unsupervised method for finding optimal weights for context words according to their distance to the target word. The general idea was to find the weights that best fit the data, in such a way that the context models for the same target word generated from two random windows samples become similar. It is the first time that this general principle is used for this purpose. Our experiments on WSD in English and Japanese suggest the validity of the principle.

In this paper, we limited context models to bag-of-words features, excluding additional features such as POS-tags. Despite this simple type of feature and the use of a simple Naïve Bayes classifier, the WSD accuracy we obtained can rival the other state-of-the-art systems with more sophisticated features and classification algorithms. This result indicates that a crucial aspect in WSD is the definition of an appropriate context model, and our weighting method can generate more reasonable weights of context words than using a predefined decaying function.

Our experiments also showed that the optimal weighting function is language-dependent. We obtained two different functions for English and Japanese, although their general shapes are similar. In fact, the optimal weighting function reflects the linguistic properties: as dependent words in Japanese can be further away from the target word due to its linguistic structure, the optimal weighting quickly decays, meaning that we can rely less on distant context words. This also shows a limitation of this study: distance is not the sole criterion to determine the impact of a context word. Other factors, such as POS-tag and syntactic dependency, can play an important role in the context model. These additional factors are complementary to the distance criterion and our approach can be extended to include such additional features. This extension is part of our future work.

Another limitation of straight window distance is that all words introduce the same distance, regardless of their nature. In our experiments, to make the distance a more sensible metric, we merged consecutive stop words in one placeholder token. The idea behind this it that some words, such as stop words, should introduce less distance than others. On the opposite, we can easily understand that tokens such as commas, full stops, parentheses and paragraph should introduce a bigger distance than regular words. We could therefore use a *congruence* score for a word, an indicator showing on average how much what comes before is similar to what comes after the word.

Also, we have combined our weighting schema with NB classifier. Other classifiers such as SVM could lead to better results. The utilization of our new weighting schema with SVM is another future work.

Finally, the weights computed with our method has been used in WSD tasks. The weights could be seen as the expected strength of relation between two words in a document according to their distance. The consideration of word relationships in documents and queries is one of the endeavors in current research in IR. The new weighting schema could be easily integrated with a dependency model in IR. We plan to perform such integration in the future.

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

Bai, Jing, Dawei Song, Peter Bruza, Jian-Yun Nie, and Guihong Cao. 2005. Query expansion using term relationships in language models for information retrieval. In *CIKM '05 Proceedings*, pages 688–695, New York, NY, USA. ACM.

Cai, Jun Fu, 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.

Cheung, Percy and Pascale Fung. 2004. Translation disambiguation in mixed language queries. *Machine Translation*, 18(4):251–273.

Gao, Jianfeng, 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.

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

Lv, Yuanhua and ChengXiang Zhai. 2009. Positional language models for information retrieval. In *SIGIR '09 Proceedings*, pages 299–306, New York, NY, USA. ACM.

Maekawa, Kikuo. 2008. Compilation of the balanced corpus of contemporary written japanese in the kotonoha initiative (invited paper). In *ISUC '08 Proceedings*, pages 169–172, Washington, DC, USA. IEEE Computer Society.

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

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

Pradhan, Sameer S., 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.

Song, D. and P. D. Bruza. 2003. Towards context sensitive information inference. *Journal of the American Society for Information Science and Technology*, 54(4):321–334.

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

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