Kernels Incorporating Word Positional Information in Natural Language Disambiguation Tasks

Tapio Pahikkala, Sampo Pyysalo, Filip Ginter, Jorma Boberg, Jouni Järvinen, Tapio Salakoski
Department of Information Technology,
University of Turku and Turku Centre for Computer Science (TUCS),
Lemminkäisenkatu 14 A,
20520 Turku, Finland
firstname.lastname@it.utu.fi

Abstract

In this paper, we introduce a new kernel function designed for the problem of word sense disambiguation. The presented kernel function employs two different types of positional information related to words present in the contexts of the words to be disambiguated. For each pair of words in two contexts, the proposed kernel takes into account both their distances from the ambiguous words and also the difference of their mutual positions. We apply the kernel to context-sensitive spelling correction with SVMs and show that it signifi cantly outperforms other considered kernels.

Introduction

Many natural language processing applications require accurate resolution of the various kinds of ambiguity present in natural language, giving rise to a class of disambiguation problems. In this paper, we focus on lexical disambiguation problems, where disambiguation is done at the level of words. A common example of such a problem is word sense disambiguation (WSD), where the task is to resolve the correct sense for an instance of a polysemous word, for example, the word *bank* where the ambiguity is between the senses "river bank" and "financial institution".

A lexical disambiguation problem closely related to WSD is context-sensitive spelling error correction, where the misspelling of the original word belongs to the language, such as, for example, *desert* misspelled as *dessert*. This mistake cannot be detected by standard lexicon-based checkers, since *dessert* belongs to the English lexicon. A set of similar words that belong to the lexicon and that are often confused with the other words in the set is called a *confusion set*. For example, {*piece*, *peace*} can be considered as a binary confusion set.

Lexical disambiguation problems other than WSD can also be used as alternatives for the purpose of evaluating WSD systems. For example, (Yarowsky 1994) studies the problem of restoring accents in Spanish and French texts as a substitute for the WSD problem. It is easy to cast context-sensitive spelling error correction as a WSD problem such that each word of the confusion set is considered as a "sense". A common motivation to use a substitute problem for the WSD is that data necessary for evaluating the methods can be obtained automatically for these problems, whereas data for the WSD problem have to be annotated by hand.

In order to disambiguate the sense of an ambiguous word, any WSD method has to incorporate the information about its context, that is, the words surrounding it in the text. A common way of representing the context is a bag-of-words (BoW), where no information of the positions of the words in the context is preserved. Positional information is directly incorporated by the ordered BoW model (Audibert 2004), where each word of the context is represented together with its position and thus two occurrences of one word at different positions in the context are considered different features. Previously, we introduced a weighted BoW approach, where the context words are weighted in such a way that the words closer to the ambiguous word receive higher values, motivated by the assumption that closer words are more relevant for disambiguation (Ginter *et al.* 2004).

We present here a new kernel function that generalizes the aforementioned BoW approaches. The proposed WSD kernel further incorporates a new type of positional information not present in the previously introduced BoW approaches. Apart from the weight associated with a context word's absolute distance from the ambiguous word itself, the proposed kernel also takes into account for two words in the compared contexts the mutual difference in the words' positions relative to the ambiguous word.

We evaluate the method using Support Vector Machines (SVMs), because they have been repeatedly shown to provide state-of-the-art performance in natural language disambiguation tasks. We use context-sensitive spelling error correction as the model problem and show a significant gain in performance compared to the weighted BoW and to the standard BoW approach.

Binary Classification with Support Vector Machines

We begin with a short description of SVMs. A more comprehensive introduction can be found, for example, in (Rifkin 2002; Vapnik 1998).

In a binary classification task, the training data is comprised of m labeled examples $(x_1, y_1), \ldots, (x_m, y_m)$, where $x_i \in X$ are training data points and $y_i \in \{-1, +1\}$ are the

corresponding class labels.

SVMs can be considered as a special case of the following regularization problem known as Tikhonov regularization:

$$\min_f \sum_i l(f(x_i), y_i) + \lambda \|f\|_k^2, \quad (1)$$

where i ranges from 1 to m, l is the loss function used by the learning machine, $f : X \to Y$ is a function which maps the input vectors $x \in X$ to the output labels $y \in Y$, $\lambda \in \mathbb{R}_+$ is a regularization parameter, and $\|\cdot\|_k$ is a norm in a Reproducing Kernel Hilbert Space defined by a positive definite kernel function k. The second term is called a regularizer. The loss function used by SVMs for binary classification problems is called linear soft margin loss or hinge loss and is defined as

$$l(f(x), y) = \max(1 - yf(x), 0).$$

By the Representer Theorem, the minimizer of (1) has the following form:

$$f(x) = \sum_i a_i k(x, x_i),$$

where $a_i \in \mathbb{R}$ and k is the kernel function associated with the Reproducing Kernel Hilbert Space mentioned above.

Word Sense Disambiguation Kernels

In this section, we consider kernel functions especially designed for word sense disambiguation. Kernel functions are similarity measures of data points in the input space X, and they correspond to an inner product in a feature space H to which the input space data points are mapped. Formally, kernel functions are defined as

$$k(x, x') = \langle \Phi(x), \Phi(x') \rangle,$$

where $\Phi : X \to H$. The input space X can be any set.

Mercer's theorem states that if k is a symmetric positive definite function, it is a valid kernel. The following closure properties of kernels can be easily derived from the Mercer's theorem (Cristianini & Shawe-Taylor 2000). If a_1 and a_2 are positive real numbers and $k_1(x, x')$ and $k_2(x, x')$ are kernels, then $a_1 k_1(x, x') + a_2 k_2(x, x')$ and $k_1(x, x') k_2(x, x')$ are also kernels. Moreover, if ψ_1 is a real valued function and ψ_2 is an $\mathbb{R}^d$ valued function, where $d \in \mathbb{N}$, then $\psi_1(x)\psi_1(x')$ and $k(\psi_2(x), \psi_2(x'))$ are kernels. Furthermore (see, e.g. (Smola & Schölkopf 2004), where further references can be found), function $k(x - x')$ is a kernel if it has a positive Fourier transform.

In our experiments, we trained SVMs to disambiguate the sense of a word between two possible senses based on its context. The contexts are extracted from documents that contain occurrences of the ambiguous words. If a document contains several such words, one context can be extracted for each occurrence. The contexts as such are not used as elements of the input space X, but they are represented as follows.

Representation of Contexts

Let τ denote a word to be disambiguated and let $\overline{\tau} = (\tau_{-t}, \ldots, \tau_{-1}, \tau_1, \ldots, \tau_r)$ be the context of τ. The words preceding τ are $\tau_{-t}, \ldots, \tau_{-1}$ in the order they appear in the text, and correspondingly $\tau_1, \ldots, \tau_r$ are the words which follow τ in the text. The word τ itself does not belong to $\overline{\tau}$. For a word τ_p, the index p is referred to as its position. Note that the numbers t and r do not have to be equal, and there does not have to be any words preceding or following τ.

Incorporation of Word Positional Information

Previously, we developed an approach that applied the information of the positions of the words with respect to the word to be disambiguated (Ginter *et al.* 2004). In this paper, we will refer to it as weighted bag-of-words (WBoW) approach. The idea of the WBoW is that the words near τ are likely more important than further words, and therefore they are given higher weight.

WBoW vector space model of contexts can be formalized as follows. Let C be a set of all possible contexts and let $V = \{v_1, \ldots, v_n\}$ be an ordered set of all distinct words of the contexts of C. Let further $\mathrm{Pos}(v, \overline{\tau}) = \{p \mid v = \tau_p \in \overline{\tau}\}$ denote the set of the positions p in which a word $v \in V$ occurs in a context $\overline{\tau}$. The weight assigned for the word positions is a function $w : \mathbb{Z} - \{0\} \to \mathbb{R}_+$. Following (Ginter *et al.* 2004) we define the weighting function

$$w(p) = \frac{1}{|p|^\alpha} + \beta,$$

where $\alpha, \beta \geq 0$ are the parameters for the weighting. If $\alpha > 0$, the weighting function has a hyperbolic shape with highest values immediately around τ, and the bigger α is, the steeper the weight values grow towards τ. The parameter β is an offset of the values, whose role is to reduce the ratio between the weights of words that are near τ and the weights of words which are far from τ.

Let ϕ now be the function which maps contexts to WBoW vectors:

$$\Phi : C \to \mathbb{R}^n, \overline{\tau} \mapsto (\phi_1(\overline{\tau}), \ldots, \phi_n(\overline{\tau})),$$

where

$$\phi_h(\overline{\tau}) = \sum_p w(p)$$

and p ranges through all values in $\mathrm{Pos}(v_h, \overline{\tau})$. The inner product

$$\langle \Phi(\overline{\tau}), \Phi(\overline{\kappa}) \rangle = \sum_h \phi_h(\overline{\tau}) \phi_h(\overline{\kappa}), \quad (2)$$

where h ranges all numbers from 1 to n, of two contexts $\Phi(\overline{\tau})$ and $\Phi(\overline{\kappa})$ can be used as a kernel function by SVMs and other kernel-based methods. Note that the setting $\alpha = \beta = 0$ corresponds to the ordinary BoW approach. We will refer to the kernel (2) as a WBoW kernel.

Incorporation of Mutual Word Positional Differences

The WBoW approach described above uses only the positional information of each word occurrence separately. The kernel function (2), however, depends on two contexts at the same time and some mutual positional information of the

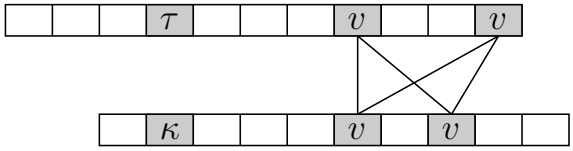

Figure 1: Consider a word v which has two occurrences in both contexts $\overline{\tau}$ and $\overline{\kappa}$. Each of the four lines between the occurrences correspond to one term in the sum (4).

word occurrences could also be incorporated into it. For example, if both two contexts have an occurrence of a certain word at similar positions, the kernel function could favor this at the expense of two contexts having the word occurrences at very different positions, for instance, far before and far after the word to be disambiguated.

Consider two instances τ and κ of the word to be disambiguated and let $\overline{\tau}$ and $\overline{\kappa}$ denote their contexts. In the WBoW approach, all mappings $\phi_h(\overline{\tau})$ are weighted sums of occurrences of the word v_h in the context $\overline{\tau}$. If we consider only a single word v_h and the corresponding mapping ϕ_h, the product of the representations of the contexts $\overline{\tau}$ and $\overline{\kappa}$ is

$$\phi_h(\overline{\tau})\phi_h(\overline{\kappa}) = \left(\sum_p w(p)\right) \cdot \left(\sum_q w(q)\right) \quad (3)$$

$$= \sum_{p,q} w(p)w(q), \quad (4)$$

where p ranges all values in $\mathrm{Pos}(v_h, \overline{\tau})$ and q ranges over $\mathrm{Pos}(v_h, \overline{\kappa})$. Hence, the sum is over all combinations of the occurrences of the word v_h in the two contexts. The sum (4) is illustrated in Figure 1.

The terms of the sum (4) are products of values of the weighting of two positions. We can generalize this kind of situation by using a function depending on the positions p and q:

$$g(p,q) : (\mathbb{Z} - \{0\}) \times (\mathbb{Z} - \{0\}) \to \mathbb{R}_+.$$

If we use $g(p,q)$ in the place of $w(p)w(q)$ in (4), we get

$$\phi_h(\overline{\tau})\phi_h(\overline{\kappa}) = \sum_{p,q} g(p,q). \quad (5)$$

If the same is done for every word in V, the result is a new kernel function

$$k(\overline{\tau}, \overline{\kappa}) = \langle \Phi(\overline{\tau}), \Phi(\overline{\kappa}) \rangle \quad (6)$$

$$= \sum_h \sum_{p,q} g(p,q), \quad (7)$$

As we mentioned earlier, the sum (4) depends only on the distance from the word to be disambiguated. In the following we take into the account also the difference of mutual positions in two contexts. This idea is illustrated in Figure 2.

One such function is $e^{-(p-q)^2}$, for example. However, this function does not take into account the distances of the words from the word to be disambiguated, but they can be

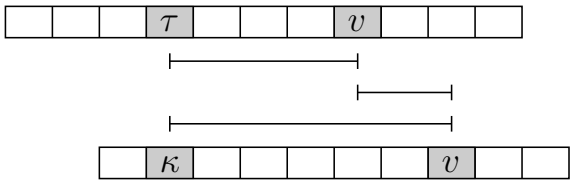

Figure 2: A word v with an occurrence in contexts $\overline{\tau}$ and $\overline{\kappa}$. There are three line segments in between the contexts. The top line is the distance of the occurrence of the word v from τ in the upper context and the bottom line is the same for $\overline{\kappa}$. The middle line is the absolute difference of the two words positions.

captured by a kernel function $e^{-p^2}e^{-q^2} = e^{-(p^2+q^2)}$. A natural way to combine these information sources is to multiply $e^{-(p-q)^2}$ by $e^{-(p^2+q^2)}$.

In our experiments, we use the function

$$g(p,q) = e^{-\alpha(p^2+q^2)-\theta(p-q)^2} + \beta, \quad (8)$$

where $\alpha \geq 0$ is a parameter that controls distances, $\theta \geq 0$ is a parameter that controls the mutual positions, and β is an offset parameter. Note that the function we obtain by combining (7) and (8) is a valid kernel function due to the closure properties we presented above.

If $\alpha > 0$, the level-curves of $g(p,q)$ are ellipses whose major axes are always parallel with the line $p = q$. The major axes of the ellipses are determined by α and the ratio of the minor axis and the major axis is $\frac{\alpha}{\alpha+2\theta}$. Thus, the ellipses are circles if θ is zero.

We may now present the following observations concerning the function (8) (see Figure 3 for illustrations).

- For $\alpha = \theta = \beta = 0$, we have a constant function 1. This corresponds the standard BoW kernel.
- For $\theta = 0$, only the term $e^{-\alpha(p^2+q^2)}$ has importance. This information is very similar to the information provided by WBoW approach.
- For $\alpha = 0$, solely the mutual positions of the context words are taken into account, not their distances from the ambiguous words.
- For $\theta \to \infty$ and $\alpha = 0$, we have a close resemblance to the ordered BoW kernel, where for each context word the exact position is important.
- For $\theta \to \infty$, we have again similarities to the ordered BoW kernel. However, since $\alpha \neq 0$, just the words close to the ambiguous words have importance.

A disadvantage of this approach is that the number of terms in the sum (5) is $|\mathrm{Pos}(v, \overline{\tau})| \times |\mathrm{Pos}(v, \overline{\kappa})|$. This number can be quite high if both contexts have several occurrences of some word v. In the WBoW approach, (3) can be computed by only $|\mathrm{Pos}(v, \overline{\tau})| + |\mathrm{Pos}(v, \overline{\kappa})| - 2$ additions. Moreover, the sums in (3) need only be calculated once. For computational efficiency when evaluating the new kernel, in each context we consider only the closest occurrence to the word to be disambiguated. With this simplification, the sum

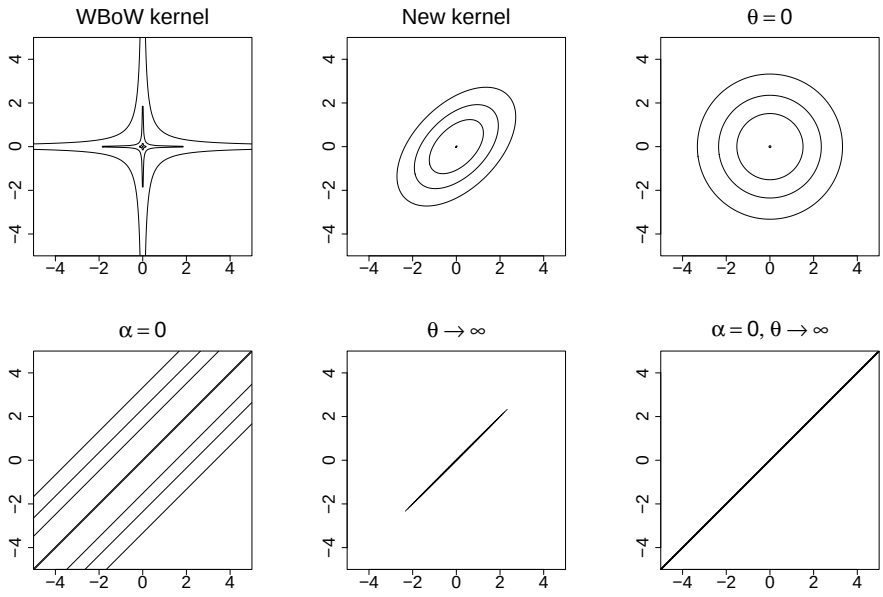

Figure 3: Schematic level curves of $g(p, q)$-values of WBoW kernel and the new kernel with different values for the parameters α and θ. The parameter β is set to zero. The axes are word positions p and q in contexts $\overline{\tau}$ and $\overline{\kappa}$, respectively.

(5) has only one term and the kernel can be computed as efficiently as the WBoW kernel. In the experiments we show that even with this simplification, the proposed kernel outperforms the WBoW kernel, which uses all context words.

We also normalize the data in the feature space using the so called normalized kernel (see e.g. (Graf, Smola, & Borer 2003))

$$\tilde{k}(x, x') = \frac{k(x, x')}{\sqrt{k(x, x)k(x', x')}}.$$

Empirical Validation

To validate the performance of the proposed kernel empirically, we consider the context-sensitive spelling error correction problem as a WSD problem. We extract the datasets from the Reuters News corpus (Rose, Stevenson, & Whitehead 2002) and consider the seven largest binary confusion sets among the sets used by (Golding & Roth 1999) in their context-sensitive spelling error correction experiments. We stem all documents using the Porter stemmer (Porter 1980).

We measure the performance of the classifiers using the area under the ROC curve (AUC) (Fawcett 2003). The ROC curve is a relation between the true-positive and the false-positive rates at various classification thresholds. Unlike other popular measures such as accuracy and precision-recall analysis, the AUC measure is invariant with respect to the prior class probabilities. AUC corresponds to the probability that given a randomly chosen positive example and a randomly chosen negative example, the classifier will correctly determine which is which.

Parameter Selection

In all experiments, we first selected parameter values by performing 10-fold cross-validation on a parameter estimation dataset of 1000 documents with various parameter combinations. Only one example per document is used, and these are selected so that the probability for each instance to be chosen is equal in the set of all contexts of all documents. We searched for the optimal parameter value combinations using coarse grid searches similar to that suggested by (Keerthi & Lin 2003). The regularization parameter λ in (1) was also separately optimized for each parameter combination.

We compared four different approaches: The standard BoW kernel, the WBoW kernel, the new kernel with $\theta = 0$ and the new kernel with unrestricted parameter values. The performance of the BoW kernel is known to depend on the context size and tends to degrade if the contexts are too large. Hence, we introduce a context span parameter s. For fixed s, we take always the largest context $\overline{\tau} = (\tau_{-t}, \ldots, \tau_{-1}, \tau_1, \ldots, \tau_r)$, so that $t \leq s$ and $r \leq s$. Note that if there exists s words preceding and following the word to be disambiguated, then $t = r = s$, and otherwise $t < s$ or $r < s$. For the WBoW kernel, we estimated the best combination of the weighting parameters α and β, and the context span parameter s. Based on the results of initial experiments, we use the following parameter selection scheme for the new kernel: we first consider the case $\theta = 0$, and find the best combination for s, α and β. Then these values for s and β are used for the new kernel in general case, and only the values for α and θ are optimized. For the standard BoW kernel, we search the optimal context span s with values $1, 2, 3, \ldots, 256$ and the full text. For other kernels we only tested the values $1, 4, 16, 64, 256$ and the full text.

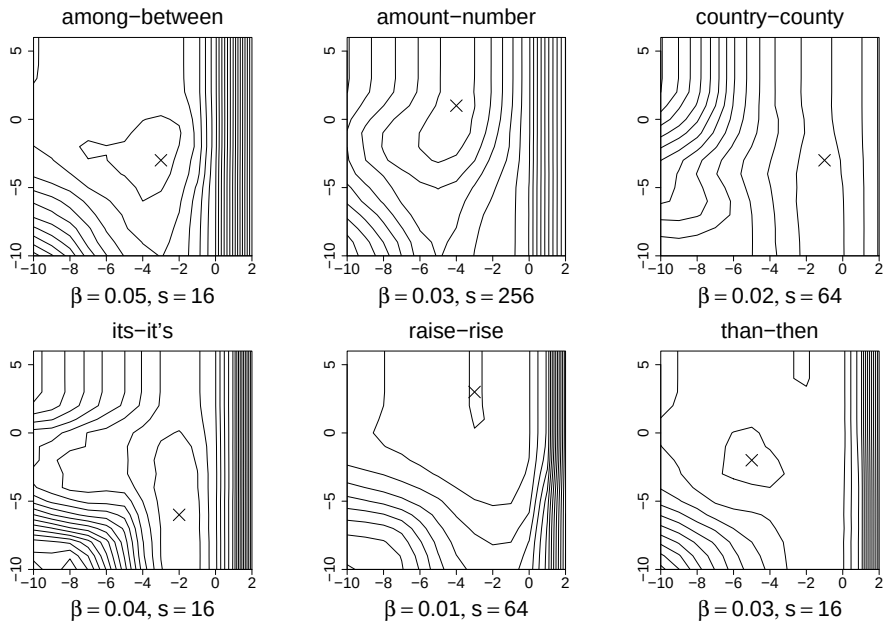

Figure 4: Optimal β and s parameters for different confusion sets for the new kernel and contour plots of performance with different values of α and θ. The x- and y- axes represent α and θ, respectively, both on a logarithmic scale. The optimal value combination is marked by a cross. The confusion set {*I, me*} is not shown as it favored a context span $s = 1$, where α has no effect. For {*I, me*}, values $\alpha = 0$, $\theta = 1$ and $\beta = 0$ were selected.

	BoW	WBoW		
Confusion set	s	s	α	β
among, between	4	64	1	0.0
amount, number	5	256	1	0.04
country, county	109	256	1.5	0.04
I, me	1	1	2	0.2
its, it's	1	16	1.5	0.2
raise, rise	1	256	1.5	0.02
than, then	2	4	1.5	0.04

Table 1: Optimal parameters values for BoW and WBoW.

The chosen parameter combinations for BoW and WBoW are presented in Table 1. The selected s parameter values suggest that the WBoW approach prefers much longer contexts. This agrees with our previous findings using a similar weighting scheme (Ginter *et al.* 2004).

In Figure 4, the optimal β and s parameters for the new kernel are given and the effect of the α and θ parameters on performance is illustrated. Similarly as for the WBoW kernel, the selected s values indicate that much longer contexts are preferred than for the BoW kernel. The α, θ-parameter spaces are relatively regular for all confusion sets, but there are significant differences in their shapes. For example, while the confusion sets {*amount, number*} and {*raise, rise*} favour relatively high values of θ, for the confusion set {*country, county*} there is little difference in performance for different values of θ given a good choice of α, whereas the confusion set {*its, it's*} favours small values of θ. This suggests that the mutual positions of context words may be more important for the first two confusion sets, and that effectively ignoring mutual word positions is best for the confusion set {*its, it's*}.

Validation Results

To validate the performance of the methods with the selected parameters and to test for statistical significance for each confusion set we use the robust 5×2-cv test (Alpaydin 1999) on separate validation sets of 20000 documents that were not used in parameter selection. The test avoids the problem of dependence between folds in N-fold cross-validation schemes and results in a more realistic estimate than, for example, the t-test. We extracted 1000 training examples from the validation training set of 10000 documents in the same way as we formed the parameter estimation set. All possible examples were extracted and tested from the validation testing set of 10000 documents.

The results of the final validation are presented in Table 2. Although the BoW kernel baseline is relatively high (average 93.6% AUC), all the other three methods significantly outperform BoW on most confusion sets, with a notable average difference of 4.1% for the new kernel. Thus, we were able to reduce 64% of errors on average. There is little difference in performance between the WBoW kernel and the new kernel with $\theta = 0$ (0.4% average in favour of the new kernel), supporting the analysis that this special case of the new kernel captures similar information as the WBoW kernel. This further indicates that the new kernel can perform competitively despite the computational simplification of only considering the closest word of each type.

	BoW	WBoW		New kernel, $\theta = 0$			New kernel			
Confusion set	AUC	AUC	Δ_1	AUC	Δ_1	Δ_2	AUC	Δ_1	Δ_2	Δ_3
among, between	90.95	93.03	**2.1**	93.43	**2.5**	**0.4**	94.93	**4.0**	**1.9**	**1.5**
amount, number	87.99	93.04	**5.0**	93.16	**5.2**	0.1	94.62	**6.6**	**1.6**	**1.5**
country, county	96.88	98.77	**1.9**	99.03	**2.1**	0.3	99.03	**2.2**	0.3	0.0
I, me	95.87	95.72	-0.1	95.72	-0.1	0.0	99.10	**3.2**	**3.4**	**3.4**
its, it's	95.97	97.63	1.7	97.97	**2.0**	**0.3**	98.17	**2.2**	**0.5**	0.2
raise, rise	89.96	94.28	**4.3**	95.11	**5.1**	**0.8**	98.54	**8.6**	**4.3**	**3.4**
than, then	97.38	98.33	**0.9**	98.91	**1.5**	**0.6**	99.15	**1.8**	**0.8**	0.2
Average	93.57	95.83	**2.3**	96.19	**2.6**	**0.4**	97.65	**4.1**	**1.8**	**1.5**

Table 2: Validation results. Δ_1 values give difference to the BoW kernel, Δ_2 to the WBoW kernel and Δ_3 to the new kernel with $\theta = 0$. Statistically significant differences are typed in bold.

The new kernel with unrestricted parameters significantly outperforms all other considered methods on most confusion sets. The difference is especially notable – more than 3% better than the performance of any other considered method – for the confusion sets {*raise*, *rise*} and {*I*, *me*}, both of which favour relatively high values of the θ parameter. The results indicate that there are different preferences in emphasis on relative context word positioning and that the new kernel can beneficially capture intermediate choices between strictly ordered and unordered alternatives.

Discussion and Conclusions

In this paper, we have shown that the performance of SVMs can be improved by incorporating the information on the positions of the words in the context of the ambiguous word into a kernel function. A straightforward way to do this is to adopt the WBoW approach and use a linear kernel over WBoW vectors. We refined this approach further by incorporating also for each pair of words in two contexts the difference of their mutual positions and showed that this information significantly improves the performance of SVMs.

As future work, we may consider one sense per discourse hypothesis (see e.g. (Yarowsky 1995)) that can be used to improve the simultaneous disambiguation of word senses in the same document. Moreover, the performance of the SVM with the new kernel might be further improved, for example, by using collocations in order to capture the local syntax around the term to be disambiguated. However, the proposed kernel uses the local information, and therefore it already captures the information represented by collocations to some extent.

Acknowledgements

We wish to thank the anonymous reviewers for their insightful suggestions. This work was supported by TEKES, the Finnish National Technology Agency.

References

Alpaydin, E. 1999. Combined 5×2 cv F-test for comparing supervised classifi cation learning algorithms. *Neural Computation* 11(8):1885–1892.

Audibert, L. 2004. Word sense disambiguation criteria: a systematic study. In *Proceedings of 20th International Conference on Computational Linguistics*, 910–916. Geneva, Switzerland: Association for Computational Linguistics.

Cristianini, N., and Shawe-Taylor, J. 2000. *An Introduction to Support Vector Machines*. Cambridge University Press.

Fawcett, T. 2003. Roc graphs: Notes and practical considerations for data mining researchers. Technical Report HPL-2003-4, HP Labs, Palo Alto, Ca.

Ginter, F.; Boberg, J.; Järvinen, J.; and Salakoski, T. 2004. New techniques for disambiguation in natural language and their application to biological text. *Journal of Machine Learning Research* 5:605–621.

Golding, A. R., and Roth, D. 1999. A winnow-based approach to context-sensitive spelling correction. *Machine Learning* 34:107–130.

Graf, A.; Smola, A.; and Borer, S. 2003. Classifi cation in a normalized feature space using support vector machines. *IEEE Transactions on Neural Networks* 14(3):597–605.

Keerthi, S. S., and Lin, C.-J. 2003. Asymptotic behaviors of support vector machines with gaussian kernel. *Neural Computation* 15:1667–1689.

Porter, M. F. 1980. An algorithm for suffi x stripping. *Program* 14:130–137.

Rifkin, R. 2002. *Everything Old Is New Again: A Fresh Look at Historical Approaches in Machine Learning.* Ph.D. Dissertation, MIT.

Rose, T. G.; Stevenson, M.; and Whitehead, M. 2002. The Reuters Corpus Volume 1: From yesterday's news to tomorrow's language resources. In Rodriguez, M. G., and Araujo, C. P. S., eds., *Proceedings of the Third International Conference on Language Resources and Evaluation.* ELRA, Paris.

Smola, A. J., and Schölkopf, B. 2004. A tutorial on support vector regression. *Statistics and Computing* 14(3):199–222.

Vapnik, V. 1998. *Statistical Learning Theory*. New York: Wiley.

Yarowsky, D. 1994. Decision lists for lexical ambiguity resolution: Application to accent restoration in spanish and french. In *Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics*, 88–95. Association for Computational Linguistics.

Yarowsky, D. 1995. Unsupervised word sense disambiguation rivaling supervised methods. In *Proceedings of the 33rd conference on Association for Computational Linguistics*, 189–196. Association for Computational Linguistics.