Extracting Word Sequence Correspondences with Support Vector Machines

Kengo SATO and **Hiroaki SAITO**
Department of Information and Computer Science
Keio University
3–14–1, Hiyoshi, Kohoku, Yokohama 223–8522, Japan
{satoken,hxs}@nak.ics.keio.ac.jp

Abstract

This paper proposes a learning and extracting method of word sequence correspondences from non-aligned parallel corpora with Support Vector Machines, which have high ability of the generalization, rarely cause over-fit for training samples and can learn dependencies of features by using a kernel function. Our method uses features for the translation model which use the translation dictionary, the number of words, part-of-speech, constituent words and neighbor words. Experiment results in which Japanese and English parallel corpora are used archived 81.1 % precision rate and 69.0 % recall rate of the extracted word sequence correspondences. This demonstrates that our method could reduce the cost for making translation dictionaries.

1 Introduction

Translation dictionaries used in multilingual natural language processing such as machine translation have been made manually, but a great deal of labor is required for this work and it is difficult to keep the description of the dictionaries consistent. Therefore, researches of extracting translation pairs from parallel corpora automatically become active recently (Gale and Church, 1991; Kaji and Aizono, 1996; Tanaka and Iwasaki, 1996; Kitamura and Matsumoto, 1996; Fung, 1997; Melamed, 1997; Sato and Nakanishi, 1998).

This paper proposes a learning and extracting method of bilingual word sequence correspondences from non-aligned parallel corpora with Support Vector Machines (SVMs) (Vapnik, 1999). SVMs are ones of large margin classifiers (Smola et al., 2000) which are based on the strategy where margins between separating boundary and vectors of which elements express the features of training samples is maximized. Therefore, SVMs have higer ability of the generalization than other learning models such as the decision trees and rarely cause over-fit for training samples. In addition, by using kernel functions, they can learn non-linear separating boundary and dependencies between the features. Therefore, SVMs have been recently used for the natural language processing such as text categorization (Joachims, 1998; Taira and Haruno, 1999), chunk identification (Kudo and Matsumoto, 2000b), dependency structure analysis (Kudo and Matsumoto, 2000a).

The method proposed in this paper does not require aligned parallel corpora which do not exist too many at present. Therefore, without limiting applicable domains, word sequence correspondences can been extracted.

2 Support Vector Machines

SVMs are binary classifiers which linearly separate d dimension vectors to two classes. Each vector represents the sample which has d features. It is distinguished whether given sample $\vec{x} = (x_1, x_2, \ldots, x_d)$ belongs to $\mathcal{X}_1$ or $\mathcal{X}_2$ by equation (1) :

$$f(\vec{x}) = \mathrm{sign}(g(\vec{x})) = \begin{cases} 1 & (\vec{x} \in \mathcal{X}_1) \\ -1 & (\vec{x} \in \mathcal{X}_2) \end{cases} \tag{1}$$

where $g(\vec{x})$ is the hyperplain which separates two classes in which $\vec{w}$ and b are decided by optimization.

$$g(\vec{x}) = \vec{w} \cdot \vec{x} + b \tag{2}$$

Let supervise signals for the training samples be expressed as

$$y_i = \begin{cases} 1 & (\vec{x}_i \in \mathcal{X}_1) \\ -1 & (\vec{x}_i \in \mathcal{X}_2) \end{cases}$$

where $\mathcal{X}_1$ is a set of positive samples and $\mathcal{X}_2$ is a set of negative samples.

If the training samples can be separated linearly, there could exist two or more pairs of $\vec{w}$ and b that

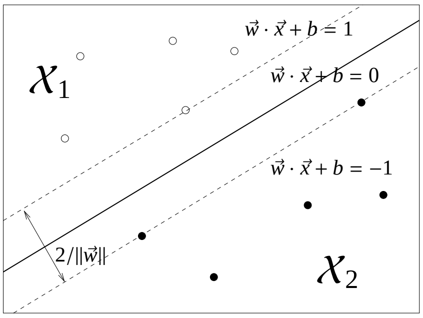
$\vec{w}\cdot\vec{x}+b=1$
$\vec{w}\cdot\vec{x}+b=0$
$\vec{w}\cdot\vec{x}+b=-1$
$\mathcal{X}_1$
$\mathcal{X}_2$
$2/||\vec{w}||$

Figure 1: A separating hyperplain

satisfy equation (1). Therefore, give the following constraints :

$$\forall i, \quad y_i(\vec{w}\cdot\vec{x_i}+b)-1\geq 0 \tag{3}$$

Figure 1 shows that the hyperplain which separates the samples. In this figure, solid line shows separating hyperplain $\vec{w}\cdot\vec{x}+b=0$ and two dotted lines show hyperplains expressed by $\vec{w}\cdot\vec{x}+b=\pm 1$. The constraints (3) mean that any vectors must not exist inside two dotted lines. The vectors on dotted lines are called support vectors and the distance between dotted lines is called a margin, which equals to $2/||\vec{w}||$.

The learning algorithm for SVMs could optimize $\vec{w}$ and b which maximize the margin $2/||\vec{w}||$ or minimize $||\vec{w}||^2/2$ subject to constraints (3). According to Lagrange's theory, the optimization problem is transformed to minimizing the Lagrangian L :

$$L=\frac{1}{2}||\vec{w}||^2+\sum_{i=1}^{n}\lambda_i\left(y_i(\vec{w}\cdot\vec{x_i}+b-1)\right) \tag{4}$$

where $\lambda_i\geq 0$ $(i=1,\ldots,n)$ are the Lagrange multipliers. By differentiating with respect to $\vec{w}$ and b, the following relations are obtained,

$$\frac{\partial L}{\partial\vec{w}}=\vec{w}-\sum_{i=1}^{n}\lambda_i y_i\vec{x}=0 \tag{5}$$

$$\frac{\partial L}{\partial b}=\sum_{i=1}^{n}\lambda_i y_i=0 \tag{6}$$

and substituting equations (5) (6) into equation (4) to obtain

$$D=-\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\lambda_i\lambda_j y_i y_j\vec{x_i}\cdot\vec{x_j}+\sum_{i=1}^{n}\lambda_i \tag{7}$$

Consequently, the optimization problem is transformed to maximizing the object function D subject to $\sum_{i=1}^{n}\lambda_i y_i=0$ and $\lambda_i\geq 0$. For the optimal parameters $\lambda^*=\arg\max_\lambda D$, each training sample $\vec{x_i}$ where $\lambda_i^*>0$ is corresponding to support vector.

$\vec{w}$ can be obtained from equation (5) and b can be obtained from

$$b=y_i-\vec{w}\cdot\vec{x_i}$$

where $\vec{x_i}$ is an arbitrary support vector. From equation (2) (5), the optimal hyperplain can be expressed as the following equation with optimal parameters λ^* :

$$g(\vec{x})=\sum_{i=1}^{n}\lambda_i^* y_i\vec{x_i}\cdot\vec{x}+b \tag{8}$$

The training samples could be allowed in some degree to enter the inside of the margin by changing equation (3) to :

$$\forall i, \quad y_i(\vec{w}\cdot\vec{x_i}+b)-1+\xi_i\geq 0 \tag{9}$$

where $\xi_i\geq 0$ are called slack variables. At this time, the maximal margin problem is enhanced as minimizing $||\vec{w}||^2/2+C\sum_{i=1}^{n}\xi_i$, where C expresses the weight of errors. As a result, the problem is to maximize the object function D subject to $\sum_{i=1}^{n}\lambda_i y_i=0$ and $0\leq\lambda_i\leq C$.

For the training samples which cannot be separated linearly, they might be separated linearly in higher dimension by mapping them using a nonlinear function:

$$\phi : R^d\mapsto R^{d'}$$

A linear separating in $R^{d'}$ for $\phi(\vec{x})$ is same as a nonlinear separating in R^d for $\vec{x}$. Let ϕ satisfy

$$K(\vec{x},\vec{x'})=\phi(\vec{x})\cdot\phi(\vec{x'}) \tag{10}$$

where $K(\vec{x},\vec{x'})$ is called kernel function. As a result, the object function is rewritten to

$$D=-\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\lambda_i\lambda_j y_i y_j K(\vec{x_i},\vec{x_j})+\sum_{i=1}^{n}\lambda_i \tag{11}$$

and the optimal hyperplain is rewritten to

$$g(\vec{x})=\sum_{i=1}^{n}\lambda_i^* y_i K(\vec{x_i},\vec{x})+b \tag{12}$$

Note that ϕ does not appear in equation (11) (12). Therefore, we need not calculate ϕ in higher dimension.

The well-known kernel functions are the polynomial kernel function (13) and the Gaussian kernel function (14).

$$K(\vec{x}, \vec{x'}) = (\vec{x} \cdot \vec{x'} + 1)^p \tag{13}$$

$$K(\vec{x}, \vec{x'}) = \exp\left(-\frac{||\vec{x} - \vec{x'}||^2}{2\delta^2}\right) \tag{14}$$

A non-linear separating using one of these kernel functions is corresponding to separating with consideration of the dependencies between the features in R^d.

3 Extracting Word Sequence Correspondences with SVMs

3.1 Outline

The method proposed in this paper can obtain word sequence correspondences (translation pairs) in the parallel corpora which include Japanese and English sentences. It consists of the following three steps:

1. Make training samples which include positive samples as translation pairs and negative samples as non-translation pairs from the training corpora manually, and learn a translation model from these with SVMs.
2. Make a set of candidates of translation pairs which are pairs of phrases obtained by parsing both Japanese sentences and English sentences.
3. Extract translation pairs from the candidates by inputting them to the translation model made in step 1.

3.2 Features for the Translation Model

To apply SVMs for extracting translation pairs, the candidates of the translation pairs must be converted into feature vectors. In our method, they are composed of the following features:

1. Features which use an existing translation dictionary.
   (a) Bilingual word pairs in the translation dictionary which are included in the candidates of the translation pairs.
   (b) Bilingual word pairs in the translation dictionary which are co-occurred in the context in which the candidates appear.
2. Features which use the number of words.
   (a) The number of words in Japanese phrases.
   (b) The number of words in English phrases.
3. Features which use the part-of-speech.
   (a) The ratios of appearance of noun, verb, adjective and adverb in Japanese phrases.
   (b) The ratios of appearance of noun, verb, adjective and adverb in English phrases.
4. Features which use constituent words.
   (a) Constituent words in Japanese phrases.
   (b) Constituent words in English phrases.
5. Features which use neighbor words.
   (a) Neighbor words which appear in Japanese phrases just before or after.
   (b) Neighbor words which appear in English phrases just before or after.

Two types of the features which use an existing translation dictionary are used because the improvement of accuracy can be expected by effectively using existing knowledge in the features. For features (1a), words included in a candidate of the translation pair are looked up with the translation dictionary and the bilingual word pairs in the candidate become features. They are based on the idea that a translation pair would include many bilingual word pairs. Each bilingual word pair included in the dictionary is allocated to the dimension of the feature vectors. If a bilingual word pair appears in the candidate of translation pair, the value of the corresponding dimension of the vector is set to 1, and otherwise it is set to 0. For features (1b), all pairs of words which co-occurred with a candidate of the translation pair are looked up with the translation dictionary and the bilingual word pairs in the dictionary become features. They are based on the idea that the context of the words which appear in neighborhood looks like each other for the translation pairs although expressed in the two different languages (Kaji and Aizono, 1996). The candidates are converted into the feature vectors just like (1a).

Features (2a) (2b) are based on the idea that there is a correlation in the number of constituent words of the phrases of both languages in the translation pair. The number of constituent words of each language is used for the feature vector.

Features (3a) (3b) are based on the idea that there is a correlation in the ratio of content words (noun, verb, adjective and adverb) which appear in the phrases of both languages in a translation pair. The ratios of the numbers of noun, verb, adjective and adverb to the number of words of the phrases of each language are used for the feature vector.

For features (4a) (4b), each content word (noun, verb, adjective and adverb) is allocated to the dimension of the feature vectors for each language. If a word appears in the candidate of translation pair, the value of the corresponding dimension of the vector is set to 1, and otherwise it is set to 0.

For features (5a) (5b), each content words (noun, verb, adjective and adverb) is allocated to the dimension of the feature vectors for each language. If a word appears in the candidate of translation pair just before or after, the value of the corresponding dimension of the vector is set to 1, and otherwise it is set to 0.

3.3 Learning the Translation Model

Training samples which include positive samples as the translation pairs and negative samples as the non-translation pairs are made from the training corpora manually, and are converted into the feature vectors by the method described in section 3.2. For supervise signals y_i, each positive sample is assigned to $+1$ and each negative sample is assigned to -1. The translation model is learned from them by SVMs described in section 2. As a result, the optimal parameters $\lambda*$ for SVMs are obtained.

3.4 Making the Candidate of the Translation Pairs

A set of candidates of translation pairs is made from the combinations of phrases which are obtained by parsing both Japanese and English sentences. How to make the combinations does not require sentence alignments between both languages. Because the set grows too big for all the combinations, the phrases used for the combinations are limited in upper bound of the number of constituent words and only noun phrases and verb phrases.

3.5 Extracting the Translation Pairs

The candidates of the translation pairs are converted into the feature vectors with the method described in section 3.2. By inputting them to equation (8) with the optimal parameters $\lambda*$ obtained in section 3.3, $+1$ or -1 could be obtained as the output for each vector. If the output is $+1$, the candidate corresponding to the input vector is the translation pair, otherwise it is not the translation pair.

4 Experiments

To confirm the effectiveness of the method described in section 3, we did the experiments where the English Business Letter Example Collection published from Nihon Keizai Shimbun Inc. are used as parallel corpora, which include Japanese and English sentences which are examples of business letters, and are marked up at translation pairs.

As both training and test corpora, 1,000 sentences were used. The translation pairs which are already marked up in the corpora were corrected to the form described in section 3.4 to be used as the positive samples. Japanese sentences were parsed by KNP [1] and English sentences were parsed by Apple Pie Parser [2]. The negative samples of the same number as the positive samples were randomly chosen from combinations of phrases which were made by parsing and of which the numbers of constituent words were below 8 words. As a result, 2,000 samples (1,000 positives and 1,000 negatives) for both training and test were prepared.

The obtained samples must be converted into the feature vectors by the method described in section 3.2. For features (1a) (1b), 94,511 bilingual word pairs included in EDICT [3] were prepared. For features (4a) (4b) (5a) (5b), 1,009 Japanese words and 890 English words which appeared in the training corpora above 3 times were used. Therefore, the number of dimensions for the feature vectors was $94,511\times2+1\times2+4\times2+1,009+890+1,009+890 = 192,830$.

SVM^{light} [4] was used for the learner and the classifier of SVMs. For the kernel function, the squared polynomial kernel ($p = 2$ in equation (13)) was used, and the error weight C was set to 0.01.

The translation model was learned by the training samples and the translation pairs were extracted from the test samples by the method described in section 3.

[1] http://www-lab25.kuee.kyoto-u.ac.jp/nl-resource/knp.html

[2] http://www.cs.nyu.edu/cs/projects/proteus/app/

[3] http://www.csse.monash.edu.au/~jwb/edict.html

[4] http://svmlight.joachims.org/

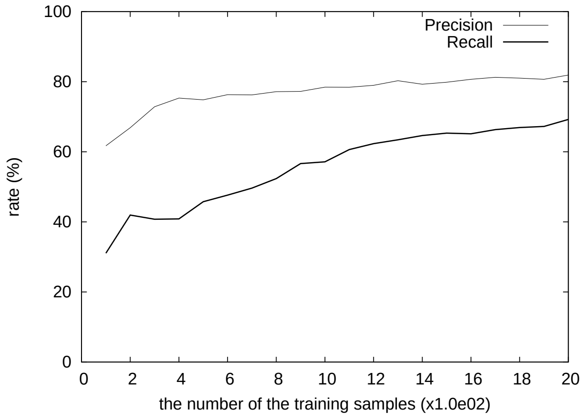

Figure 2: Transition in the precision rate and the recall rate when the number of the training samples are increased

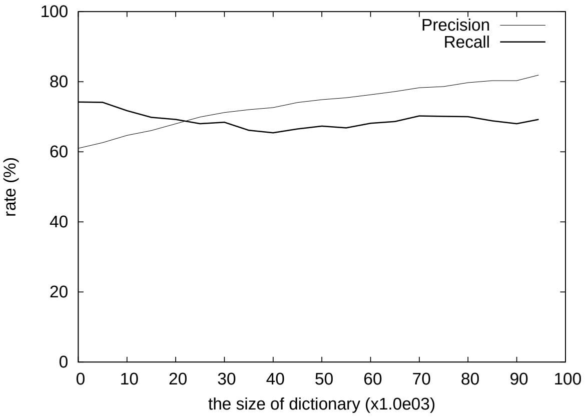

Figure 3: Transition in the precision rate and the recall rate when the number of the bilingual word pairs in the translation dictionary are increased

Table 1 shows the precision rate and the recall rate of the extracted translation pairs, and table 2 shows examples of the extracted translation pairs.

Table 1: Precision and recall rate

Outputs	Corrects	Precision	Recall
851	690	81.1 %	69.0 %

5 Discussion

Figure 2 shows the transition in the precision rate and the recall rate when the number of the training samples are increased from 100 to 2,000 by every 100 samples. The recall rate rose according to the number of the training samples, and reaching the level-off in the precision rate since 1,300. Therefore, it suggests that the recall rate can be improved without lowering the precision rate too much by increasing the number of the training samples.

Figure 3 shows that the transition in the precision rate and the recall rate when the number of the bilingual word pairs in the translation dictionary are increased from 0 to 90,000 by every 5,000 pairs. The precision rate rose almost linearly according to the number of the pairs, and reaching the level-off in the recall rate since 30,000. Therefore, it suggests that the precision rate can be improved without lowering the recall rate too much by increasing the number of the bilingual word pairs in the translation dictionary.

Table 3 shows the precision rate and the recall rate when each kind of features described in section 3.2 was removed. The values in parentheses in the columns of the precision rate and the recall rate are differences with the values when all the features are used. The fall of the precision rate when the features which use the translation dictionary (1a) (1b) were removed and the fall of the recall rate when the features which use the number of words (2a) (2b) were removed were especially large.

It is clear that feature (1a) (1b) could restrict the translation model most strongly in all features. Therefore, if feature (1a) (1b) were removed, it causes a good translation model not to be able to be learned only by the features of the remainder because of the weak constraints, wrong outputs increased, and the precision rate has fallen.

Only features (2a) (2b) surely appear in all samples although some other features appeared in the training samples may not appear in the test samples. So, in the test samples, the importance of features (2a) (2b) are increased on the coverage of the samples relatively. Therefore, if features (2a) (2b) were removed, it causes the recall rate to fall because of the low coverage of the samples.

6 Related Works

With difference from our method, there have been researches which are based on the assumption of the sentence alignments for parallel corpora (Gale and Church, 1991; Kitamura and Matsumoto, 1996; Melamed, 1997). (Gale and Church, 1991) has used the ϕ^2 statistics as the correspondence level of the word pairs and has showed that it was more effective than the mutual information. (Kitamura and Matsumoto, 1996) has used the Dice coefficient (Kay and Röschesen, 1993) which was weighted by the logarithm of the frequency of the word pair as the

Table 2: Examples of translation pairs extracted by our method

Japanese	English
特別企画委員会の委員長	chairman of a special program committee
から正式に引退し	officially retired as
公式のお別れのご挨拶を申し上げたく	would like to say an official farewell
30 年にわたる私の経験	my thirty years of experience
ゴルフの腕を磨ける	sharpen up on my golf

Table 3: Precision rate and recall rate when each kind of features is removed

Feature	Num.	Outputs	Corrects	Precision (%)	Recall (%)
(1a)	94,511	891	686	77.0 (−4.1)	68.6 (−0.4)
(1b)	94,511	1,058	719	68.0 (−13.1)	71.9 (+2.9)
(1)	189,022	1,237	756	61.1 (−20.0)	75.6 (+6.6)
(2a)	1	742	611	82.3 (+1.3)	61.1 (−7.9)
(2b)	1	755	600	79.5 (−1.6)	60.0 (−9.0)
(2)	2	489	404	82.6 (+1.5)	40.4 (−28.6)
(3a)	4	846	685	81.0 (−0.1)	68.5 (−0.5)
(3b)	4	834	660	79.1 (−1.9)	66.0 (−3.0)
(3)	8	840	661	78.7 (−2.4)	66.1 (−2.9)
(4a)	1,009	814	668	82.1 (+1.0)	66.8 (−2.2)
(4b)	890	855	698	81.6 (+0.6)	69.8 (+0.8)
(4)	1,899	838	689	82.2 (+1.1)	68.9 (−0.1)
(5a)	1,009	844	683	80.9 (−0.2)	68.3 (−0.7)
(5b)	890	851	688	80.8 (−0.3)	68.8 (−0.2)
(5)	1,899	845	682	80.7 (−0.4)	68.2 (−0.8)
All features	192,830	851	690	81.1	69.0

correspondence level of the word pairs. (Melamed, 1997) has proposed the Competitive Linking Algorithm for linking the word pairs and a method which calculates the optimized correspondence level of the word pairs by hill climbing.

These methods could archive high accuracy because of the assumption of the sentence alignments for parallel corpora, but they have the problem with narrow applicable domains because there are not too many parallel corpora with sentence alignments at present. However, because our method does not require sentence alignments, it can be applied for wider applicable domains.

Like our method, researches which are not based on the assumption of the sentence alignments for parallel corpora have been done (Kaji and Aizono, 1996; Tanaka and Iwasaki, 1996; Fung, 1997). They are based on the idea that the context of the words which appear in neighborhood looks like each other for the translation pairs although expressed in two different languages. (Kaji and Aizono, 1996) has proposed the correspondence level calculated by the size of intersection between co-occurrence sets with the word included in an existing translation dictionary. (Tanaka and Iwasaki, 1996) has proposed a method for obtaining the bilingual word pairs by optimizing the matrix of the translation probabilities so that the distance of the matrices of the probabilities of co-occurrences of words which appeared in each language might become small. (Fung, 1997) has calculated the vectors in which the weighted mutual information between the word in the corpora and the word included in an existing translation dictionary was an element, and has used these inner products as the correspondence level of word pairs.

There is a common point between these method and ours on the idea that the context of the words which appear in neighborhood looks like each other for the translation pairs because features (1b) are based on the same idea. However, since our method caught extracting the translation pairs as the approach of the statistical machine learning, it could be expected to improve the performance by adding new features to the translation model. In addition, if learning the translation model for the training samples is done once with our method, the model need not be learned again for new samples although it needs the positive and negative samples for the training data. However, the methods introduced above must learn a new model again for new corpora.

(Sato and Nakanishi, 1998) has proposed a method for learning a probabilistic translation model with Maximum Entropy (ME) modeling which was the same approach of the statistical machine learning as SVMs, in which co-occurrence information and morphological information were used as features and has archived 58.25 % accuracy with 4,119 features. ME modeling might be similar to SVMs on using features for learning a model, but feature selection for ME modeling is more difficult because ME modeling is easier to cause over-fit for training samples than SVMs. In addition, ME modeling cannot learn dependencies between features, but SVMs can learn them automatically using a kernel function. Therefore, SVMs could learn more complex and effective model than ME modeling.

7 Conclusion

In this paper, we proposed a learning and extracting method of bilingual word sequence correspondences from non-aligned parallel corpora with SVMs. Our method used features for the translation model which use the translation dictionary, the number of words, the part-of-speech, constituent words and neighbor words. Experiment results in which Japanese and English parallel corpora are used archived 81.1 % precision rate and 69.0 % recall rate of the extracted translation pairs. This demonstrates that our method could reduce the cost for making translation dictionaries.

Acknowledgments

We would like to thank Nihon Keizai Shimbun Inc. for giving us the research application permission of the English Business Letter Example Collection.

References

Pascale Fung. 1997. Finding terminology translation from non-parallel corpora. In *Proceeding of the 5th Workshop on Very Large Corpora*, pages 192–202.

William A. Gale and Kenneth W. Church. 1991. Identifying word correspondances in parallel texts. In *Proceedings of the 2nd Speech and Natural Language Workshop*, pages 152–157.

Thorsten Joachims. 1998. Text categorization with support vector machines: Learning with many relevant features. In *the 10th European Conference on Machine Learning*, pages 137–142.

Hiroyuki Kaji and Toshiko Aizono. 1996. Extracting word correspondences from bilingual corpora based on word co-occurrence information. In *Proceedings of the 16th International Conference on Computational Linguistics*, pages 23–28.

Martin Kay and Martin Röschesen. 1993. Text-translation alignment. *Computational Linguistics*, 19(1):121–142.

Mihoko Kitamura and Yuji Matsumoto. 1996. Automatic extraction of word sequence correspondences in parallel corpora. In *Proceeding of the 4th Workshop on Very Large Corpora*, pages 78–89.

Taku Kudo and Yuji Matsumoto. 2000a. Japanese dependency structure analysis based on support vector machines. In *Proceedings of the 2000 Joint SIGDAT Conference on Emprical Methods in Natural Language Processing and Very Large Corpora*, pages 18–25, Hong Kong, October.

Taku Kudo and Yuji Matsumoto. 2000b. Use of support vector learning for chunk identification. In *Proceedings of the 4th Conference on Computational Natural Language Learning and the 2nd Learning Language in Logic Workshop*, pages 142–144, Lisbon, September.

I. Dan Melamed. 1997. A word-to-word model of translation equivalence. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics*, pages 490–497.

Kengo Sato and Masakazu Nakanishi. 1998. Maximum entropy model learning of the translation rules. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and the 17th International Conference on Computational Linguistics*, pages 1171–1175, August.

Alexander J. Smola, Peter J. Bartlett, bernha Schölkopf, and Dale Schuurmans, editors. 2000. *Advances in Large Margin Classifiers*. MIT Press.

Hirotoshi Taira and Masahiko Haruno. 1999. Feature selection in svm text categorization. In *Proceedings of the 16th National Conference of the American Associtation of Artificial Intelligence*, pages 480–486, Florida, July.

Kumiko Tanaka and Hideya Iwasaki. 1996. Extraction of lexical translatins from non-aligned corpora. In *Proceedings of the 16th International Conference on Computational Linguistics*, pages 580–585.

Vladimir Naumovich Vapnik. 1999. *The Nature of Statistical Learning Theory (Statistics for Engineering and Information Seience)*. Springer-Verlag Telos, 2nd edition, December.