Unknown Word Guessing and Part-of-Speech Tagging Using Support Vector Machines

Tetsuji Nakagawa, Taku Kudoh and Yuji Matsumoto
Graduate School of Information Science,
Nara Institute of Science and Technology
8916−5 Takayama, Ikoma, Nara 630−0101, Japan
{tetsu-na,taku-ku,matsu}@is.aist-nara.ac.jp

Abstract

The accuracy of part-of-speech (POS) tagging for unknown words is substantially lower than that for known words. Considering the high accuracy rate of up-to-date statistical POS taggers, unknown words account for a non-negligible portion of the errors. This paper describes POS prediction for unknown words using Support Vector Machines. We achieve high accuracy in POS tag prediction using substrings and surrounding context as the features. Furthermore, we integrate this method with a practical English POS tagger, and achieve accuracy of 97.1%, higher than conventional approaches.

1 Introduction

Part-of-speech (POS) tagging is fundamental in natural language processing. Many statistical POS taggers use text data which are manually annotated with POS tags as training data to obtain the statistical information or rules to perform POS tagging. However, in part-of-speech tagging, we frequently encounter words that do not exist in training data. Such unknown words are usually handled by an exceptional processing, because the statistical information or rules for those words are unknown. Many machine learning methods have been applied for part-of-speech tagging, such as the hidden Markov model (HMM) (Charniak et al., 1993), the transformation-based error-driven system (Brill, 1995), the decision tree (Schmid, 1994) and the maximum entropy model (Ratnaparkhi, 1996). Though these methods have good performance, the accuracy for unknown words is much lower than that for known words, and this is a non-negligible problem where training data is limited (Brants, 2000).

One known approach for unknown word guessing is to use suffixes or surrounding context of unknown words (Thede, 1998). Weischedel estimated conditional probability for an unknown word w given a tag t using the ending form of w, the existence of hyphenation and capitalization (Weischedel et al., 1993):

$$\begin{aligned} p(w|t) &= p(\textit{unknown word}|t) \cdot p(\textit{capital}|t) \\ &\quad \cdot p(\textit{ending/hyphenation}|t) \end{aligned}$$

Although this method has the merit in handling unknown words within the framework of probability theory, ending forms like "-ed" and "-ion" are selected heuristically, so applying this method to other languages is not straightforward. Brants used the linear interpolation of fixed length suffix model for unknown word handling in his part-of-speech tagger TnT (Brants, 2000). This method achieves relatively high accuracy and is reported to be effective for other languages (Džeroski et al., 2000). Cucerzan and Yarowsky proposed paradigmatic similarity measures and showed a good result for highly inflectional languages using a large amount of unannotated text (Cucerzan and Yarowsky, 2000). Other methods for unknown word guessing have been studied, such as the

rule-based method (Mikheev, 1997) and the decision tree-based method (Orphanos and Christodoulakis, 1999).

In this paper, we propose a method to predict POS tags of unknown English words as a post-processing of POS tagging using Support Vector Machines (SVMs). SVMs (Cortes and Vapnik, 1995; Vapnik, 1999) are a supervised machine learning algorithm for binary classification and known to have good generalization performance. SVMs can handle a large number of features and hardly overfit. Consequently, SVMs can be applied successfully to natural language processing applications (Joachims, 1998; Kudoh and Matsumoto, 2000). In this paper, we show how to apply SVMs to more general POS tagging as well as unknown word guessing, and report some experimental results.

The paper is organized as follows: We start by presenting Support Vector Machines in the next section. We then describe our method for unknown word guessing and POS tagging in sections 3 and 4. In section 5, we describe the results of some experiments.

2 Support Vector Machines

Support Vector Machines (SVMs) are supervised machine learning algorithm for binary classification on a feature vector space $\mathbf{x} \in \mathbf{R}^L$.

$$\mathbf{w} \cdot \mathbf{x} + b = 0, \qquad \mathbf{w} \in \mathbf{R}^L,\ b \in \mathbf{R}. \tag{1}$$

Suppose the hyperplane (1) separates the training data, $\{(\mathbf{x}_i, y_i) \,|\, \mathbf{x}_i \in \mathbf{R}^L, y_i \in \{\pm 1\}, 1 \leq i \leq l\}$, into two classes such that

$$y_i \cdot (\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1. \tag{2}$$

While several of such separating hyperplanes exist (Figure 1, left), SVMs find the optimal hyperplane that maximizes the margin (the distance between the hyperplane and the nearest points) (Figure 1, right). Such a hyperplane is known to have the minimum expected test error and can be solved by quadratic programming. Given a test example $\mathbf{x}$, its label y is decided by the sign of

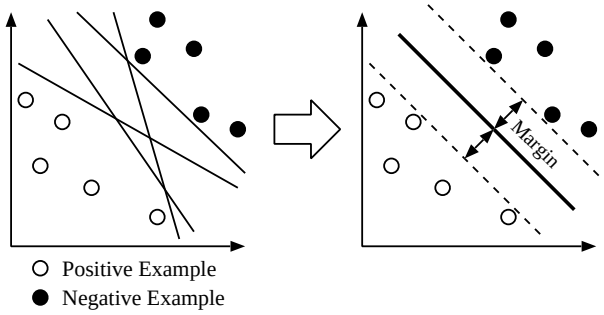

Figure 1: Maximize the Margin

discriminant function $f(\mathbf{x})$ as follows:

$$f(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x} + b, \tag{3}$$

$$y = \mathrm{sgn}(f(\mathbf{x})). \tag{4}$$

For linearly non-separable cases, feature vectors are mapped into a higher dimensional space by a nonlinear function $\mathbf{\Phi}(\mathbf{x})$ and linearly separated there. In SVMs' formula, since all data points appear as a form of inner product, we only need the inner product of two points in the higher dimensional space. Those values are calculated in $\mathbf{R}^L$ without mapping to the higher dimensional space by the following function $K(\mathbf{x}_i, \mathbf{x}_j)$ called a kernel function,

$$\mathbf{\Phi}(\mathbf{x}_i) \cdot \mathbf{\Phi}(\mathbf{x}_j) = K(\mathbf{x}_i, \mathbf{x}_j). \tag{5}$$

In this paper, we use the following polynomial kernel function,

$$K(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i \cdot \mathbf{x}_j + 1)^d. \tag{6}$$

This function virtually maps the original input space into a higher dimensional space where all combinations of up to d features are taken into consideration.

Since SVMs are binary classifiers, we must extend them to multi-class classifiers to predict $k > 2$ POS tags. Among several methods of multi-class classification for SVMs (Weston and Watkins, 1999), we employ the one-versus-rest approach. In training, k classifiers $f_i(\mathbf{x})\,(1 \leq i \leq k)$ are created to classify the class i from all other classes,

$$\begin{cases} f_i(\mathbf{x}) \geq +1 & \mathbf{x} \text{ belongs to the class } i, \\ f_i(\mathbf{x}) \leq -1 & \text{otherwise.} \end{cases} \tag{7}$$

Table 1: Example of Features for Unknown Word Guessing

POS Context	t_{-1} =TO, t_{-2} =VBD, t_{+1} =CD, t_{+2} =NNS
Word Context	w_{-1} =to, w_{-2} =returned, w_{+1} =two, w_{+2} =days
Substrings	^g, ^gr, ^gre, ^gree, e$, le$, lle$, ille$, $\langle Capital \rangle$

Given a test example $\mathbf{x}$, its class c is determined by the classifier that gives the largest discriminating function value,

$$c = \operatorname*{argmax}_i f_i(\mathbf{x}). \quad (8)$$

3 Predicting POS Tags of Unknown Words

In order to predict the POS tag of an unknown word, the following features are used:

(1) POS context: The POS tags of the two words on both sides of the unknown word.

(2) Word context: The lexical forms of the two words on both sides of the unknown word.

(3) Substrings: Prefixes and suffixes of up to four characters of the unknown word and the existence of numerals, capital letters and hyphens in the unknown word.

For the following example sentence,

```
... she/PRP returned/VBD to/TO
Greenville/(Unknown Word) two/CD
days/NNS before/IN ...
```

the features for the unknown word "Greenville" are shown in Table 1[1]. These features are almost same as those used by Ratnaparkhi (Ratnaparkhi, 1996), but combination of POS tags is not used because polynomial kernel can automatically consider them.

[1] "^" and "$" mean the beginning and the end of the word respectively.

SVM classifiers are created for each POS tag using all words in the training data. Then POS tags of unknown words are predicted using those classifiers.

4 Part-of-Speech Tagging

In unknown word guessing, the POS tag of an unknown word is predicted using the POS context, the word context and the substrings. This method can be extended to more general POS tagging by predicting the POS tags of all words in a given sentence. Differing from unknown word guessing as a post-processing of POS tagging, the POS tags for succeeding words are usually not known during POS tagging. Therefore, two methods for this task are tested as described in the following subsections.

4.1 Using Only the Preceding POS Tags

The first method uses only the POS tags of the preceding words. As features, (1) the POS tags of the two preceding words, (2) two preceding and succeeding words, and (3) prefixes and suffixes of up to four characters, the existence of numerals, capital letters and hyphens are used.

In probabilistic models such as HMM, the generative probabilities of all sequences are considered and the most likely path is selected by the Viterbi algorithm. Since the SVMs do not produce probabilities, we employ a deterministic method for POS tag prediction of the guessing word by referring to the features shown above. This method has the merit of having a small computational cost, but it has the demerit of not using the information of the succeeding POS tags.

A tag dictionary which provides the lists of POS tags for known words (i.e., that appeared in training data) is used. This dictionary was also used by Ratnaparkhi to reduce the number of possible POS tags for known words (Ratnaparkhi, 1996). For unknown words, all possible POS tags are taken as the candidates.

This method requires no exceptional processings to handle unknown words.

Table 2: Test Data for POS Tagging

Training Tokens	Known Words/Unknown Words	Percentage of Unknown Word
1,000	153,492/131,316	46.1%
10,000	218,197/ 66,611	23.4%
100,000	261,786/ 23,022	8.1%
1,000,000	278,535/ 6,273	2.2%

4.2 Using the Preceding and Succeeding POS Tags

The second method uses the POS tag information on both sides of the guessing word. Same features as shown in Table 1 are used.

In general, the POS tags of the succeeding words are unknown. Roth and Zelenko address the POS tagging with no unknown words and used a dictionary with the most frequent POS tag for each word (Roth and Zelenko, 1998). They used that POS tag for the succeeding words. They report that about 2% of accuracy decrease is caused by incorrectly attached POS tags by their method. We use a similar two pass method without using a dictionary. In the first pass, all POS tags are predicted without using the POS tag information of succeeding words (i.e., using the same features as section 4.1). In the second pass, POS tagging is performed using the POS tags predicted in the first pass for the succeeding context (i.e., using the same features as section 3). This method has the advantage of handling known and unknown words in the same way.

The tag dictionary is also used in this method.

5 Evaluation

Experiments for unknown word guessing and POS tagging are performed using the Penn Treebank WSJ corpus having 50 POS tags. Four training data sets were constructed by randomly selecting approximately 1,000, 10,000, 100,000 and 1,000,000 tokens.

Test data for unknown word guessing consists of words that do not appear in the training data. The POS tags on both sides of the unknown word were tagged by TnT.

Test data for POS tagging consists of about 285,000 tokens differing from the training data. The number of known/unknown words and the percentage of unknown word in the test data are shown in Table 2.

The accuracies are compared with TnT which is based on a second order Markov model.

5.1 Unknown Word Guessing

The accuracy of the unknown word guessing is shown in Table 3 together with the degree of polynomial kernel used for the experiments. Our method has higher accuracy compared to TnT for every training data set.

Accuracies with various settings are shown in Table 4. The first column shows the cases when the correct POS context on both sides of the guessing words is given. From the second to fourth columns, some features are deleted so as to see the contribution of the features to the accuracy. The decrease of accuracy caused by the errors in POS tagging by TnT is about 1%. Information from substrings (prefixes, suffixes and the existence of numerals, capital letters and hyphens) plays the most important role in predicting POS tags. On the other hand, the words themselves have much less contribution while the POS context have moderate contribution to the final accuracy.

In general, features that rarely appear in the training data are statistically unreliable, and often decrease the performance of the system. Ratnaparkhi ignored features that appeared less than 10 times in training data (Ratnaparkhi, 1996). We examined the behavior when reducing the sparse features. Table 5 shows the result for 10,000 training tokens. Ignoring the features that appeared only once, the accuracy is a bit improved.

Table 3: Performance of Unknown Word Guessing vs. Size of Training Tokens

Training Tokens	SVMs	d	TnT
1,000	69.8%	1	69.4%
10,000	82.3%	2	81.5%
100,000	86.7%	2	83.3%
1,000,000	87.1%	2	84.2%

Table 4: Performance of Unknown Word Guessing for Different Sets of Features

Training Tokens	Correct POS	No POS	No Word	No Substrings
1,000	71.1%	64.2%	68.7%	33.7%
10,000	82.9%	75.8%	80.2%	37.1%
100,000	87.5%	80.1%	85.2%	33.8%
1,000,000	88.5%	80.8%	86.0%	30.0%

Table 5: Performance of Unknown Word Guessing along Reduction of Features: Features occurring less than or equal to "cutoff" are not taken into consideration.

Cutoff	0	1	10	100
Accuracy	82.3%	82.5%	81.7%	74.8%
Number of Features	18,936	7,683	1,314	201

Table 6: Performance of Unknown Word Guessing for Different Kernel Functions

Training Tokens	Degree of Polynomial Kernel			
	1	2	3	4
1,000	69.8%	69.8%	61.2%	34.8%
10,000	82.0%	82.3%	80.3%	79.5%
100,000	85.0%	86.7%	86.0%	84.3%
1,000,000	—	87.1%	—	—

Table 7: Performance of POS tagging (for Known Word/for Unknown Word)

Training Tokens	SVMs				TnT
	Preceding POS	d	Preceding & Succeeding POS	d	
1,000	83.4%(96.3/68.3)	1	83.9%(96.4/69.3)	1	83.8%(96.0/69.4)
10,000	92.1%(95.5/81.2)	2	92.5%(95.7/82.2)	2	92.3%(95.7/81.5)
100,000	95.6%(96.5/85.7)	2	95.9%(96.7/86.7)	2	95.4%(96.4/83.3)
1,000,000	97.0%(97.3/86.3)	2	97.1%(97.3/86.9)	2	96.6%(96.9/84.2)

Table 8: Performance of POS tagging for Different Sets of Features (for Known Word/for Unknown Word)

Training Tokens	No POS	No Word	No Substrings
1,000	81.3%(96.0/64.2)	83.2%(96.2/68.0)	65.2%(96.2/29.0)
10,000	90.3%(94.7/75.8)	91.9%(95.5/79.9)	80.7%(94.8/34.4)
100,000	93.9%(95.1/80.1)	95.4%(96.4/85.0)	82.4%(87.0/30.7)
1,000,000	95.0%(95.3/80.8)	96.7%(96.9/85.7)	74.4%(75.6/24.1)

However, even if a large number of features are used without a cutoff, SVMs hardly overfit and keep good generalization performance.

The performance at the different degree of polynomial kernel is shown in Table 6. The best degree seems to be 1 or 2 for this task, and the best degree tends to increase when the training data increases.

5.2 Part-of-Speech Tagging

The accuracies of POS tagging are shown in Table 7. The table shows two cases: one refers only to the preceding POS tags, and the other refers to both preceding and succeeding POS tags with the two pass method. The results show that the performance is comparable to TnT in the first case and better in the second case. Between the first case and the second case, the accuracy for known words are almost equal, but the accuracy of the first case for unknown words is lower than that of the second case.

Accuracies measured by deleting each feature are shown in Table 8. The contribution of each feature has the same tendency as the case of the unknown word guessing in section 5.1. The biggest difference of features between our method and the TnT is the use of word context. Although using a lot of features such as word context is difficult in Markov model, it is easy in SVMs as seen in section 5.1. For small training data, the accuracies of the case without the word context are less than that of TnT. This means that one reason for better performance of our method is the use of word context.

6 Conclusion and Future Work

In this paper, we applied SVMs to unknown word guessing and showed that they perform quite well using context and substring information. Furthermore, extending the method to POS tagging, the resulting tagger achieves higher accuracy than the state-of-the-art HMM-based tagger. Comparing to other machine learning algorithms, SVMs have the advantage of considering the combinations of features automatically by introducing a kernel function and seldom over-fit with a large set of features. Our methods do not depend on particular characteristics of English, therefore, our methods can be applied to other languages such as German and French. However, for languages like Japanese and Chinese, it is difficult to apply our methods straightforwardly because words are not separated by spaces in those languages.

One problem of our methods is computational cost. It took about 16.5 hours for training with 100,000 tokens and 4 hours for testing with 285,000 tokens in POS tagging using POS tags on both sides on an Alpha 21164A 500MHz processor. Although SVMs have good properties and performance, their computational cost is large. It is difficult to train for a large amount of training data, and testing time increases in more complex models.

Another point to be improved is the search algorithm for POS tagging. In this paper, a deterministic method is used as a search algorithm. This method does not consider the overall likelihood of a whole sentence and uses only local information compared to probabilistic models. The accuracy may be improved by incorporating some beam search scheme. Furthermore, our method outputs only the best answer and cannot output the second or third best answer. There is a way to translate the outputs of SVMs as probabilities (Platt, 1999), which may be applied directly to remedy this problem.

References

T. Brants. 2000. TnT — A Statistical Part-of-Speech Tagger. In *Proceedings of the 6th Applied NLP Conference(ANLP-2000)*, pages 224–231.

E. Brill. 1995. Transformation-Based Error-Driven Learning and Natural Language Processing: A Case Study in Part-of-Speech Tagging. *Computational Linguistics*, 21(4), pages 543–565.

E. Charniak, C. Hendrickson, N. Jacobson and M. Perkowitz. 1993. Equations for Part-of-Speech Tagging. In *Proceedings of*

the Eleventh National Conference on Artificial Intelligence(AAAI-93), pages 784–789.

C. Cortes and V. Vapnik. 1995. Support Vector Networks *Machine Learning*, 20, pages 273–297.

S. Cucerzan and D. Yarowsky. 2000. Language Independent, Minimally Supervised Induction of Lexical Probabilities. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics(ACL-2000)*, pages 270–277.

S. Džeroski, T. Erjavec and J. Zavrel. 2000. Morphosyntactic Tagging of Slovene: Evaluating Taggers and Tagsets. In *Proceedings of the Second International Conference on Language Resources and Evaluation(LREC-2000)*, pages 1099–1104.

T. Joachims. 1998. Text Categorization with Support Vector Machines: Learning with Many Relevant Features. In *Proceedings of the 10th European Conference on Machine Learning(ECML-98)*, pages 137–142.

T. Kudoh and Y. Matsumoto. 2000. Use of Support Vector Learning for Chunk Identification In *Proceedings of the Fourth Conference on Computational Natural Language Learning(CoNLL-2000)*, pages 142–144.

A. Mikheev. 1997. Automatic Rule Induction for Unknown-Word Guessing. *Computational Linguistics*, 23(3), pages 405–423.

G. Orphanos and D. Christodoulakis. 1999. POS Disambiguation and Unknown Word Guessing with Decision Trees. In *Proceedings of the Ninth Conference of the European Chapter of the Association for Computational Linguistics(EACL-99)*, pages 134–141.

J. Platt. 1999. Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods. *Advances in Large Margin Classifiers*. MIT Press.

A. Ratnaparkhi. 1996. A Maximum Entropy Model for Part-of-Speech Tagging. In *Proceedings of Conference on Empirical Methods in Natural Language Processing(EMNLP-1)*, pages 133–142.

D. Roth and D. Zelenko. 1998. Part of Speech Tagging Using a Network of Linear Separators. In *Proceedings of the joint 17th International Conference on Computational Linguistics and 36th Annual Meeting of the Association for Computational Linguistics(ACL/COLING-98)*, pages 1136–1142.

H. Schmid. 1994. Probabilistic Part-of-Speech Tagging Using Decision Trees. In *Proceedings of the International Conference on New Methods in Language Processing(NeMLaP-1)*, pages 44–49.

S. Thede. 1998. Predicting Part-of-Speech Information about Unknown Words using Statistical Methods. In *Proceedings of the joint 17th International Conference on Computational Linguistics and 36th Annual Meeting of the Association for Computational Linguistics(ACL/COLING-98)*, pages 1505–1507.

V. Vapnik. 1999. *The Nature of Statistical Learning Theory*. Springer.

R. Weischedel, M. Meteer, R. Schwartz, L. Ramshaw and J. Palmucci. 1993. Coping with Ambiguity and Unknown Words through Probabilistic Models. *Computational Linguistics*, 19(2), pages 359–382.

J. Weston and C. Watkins. 1999. Support Vector Machines for Multi-Class Pattern Recognition. In *Proceedings of the Seventh European Symposium On Artificial Neural Networks(ESANN-99)*.