Japanese word sense disambiguation using the simple Bayes and support vector machine methods

Masaki Murata, Masao Utiyama, Kiyotaka Uchimoto, Qing Ma, and **Hitoshi Isahara**
Communications Research Laboratory
2-2-2 Hikaridai, Seika-cho, Soraku-gun, Kyoto, 619-0289, Japan

Abstract

We submitted four systems to the Japanese dictionary-based lexical-sample task of SENSEVAL-2. They were i) the support vector machine method ii) the simple Bayes method, iii) a method combining the two, and iv) a method combining two kinds of each. The combined methods obtained the best precision among the submitted systems. After the contest, we tuned the parameter used in the simple Bayes method, and it obtained higher precision. An explanation of these systems used in Japanese word sense disambiguation was provided.

1 Introduction

We participated in the Japanese dictionary-based lexical-sample task of the SENSEVAL-2 contest. We used machine learning approaches and submitted four systems. After the contest, we tuned the parameter used in the simple Bayes method and carried out additional experiments. In this paper, we explain the systems and their experimental results.

2 Task Descriptions

The test data included 10,000 instances for evaluation. The RWC corpus (Shirai et al., 2001) was given as the training data. It was made from 3000 articles published in the Mainichi Newspaper. The nouns, verbs, and adjectives (the total number of which was about 150,000) were assigned sense tags defined on the basis of the Iwanami dictionary. The purpose of this task was to estimate the sense of a word by using its context.

3 Methods

Because the word sense assigned to each word is dependent on the word itself, estimations were conducted using machine learning methods for each word. That is, we constructed as many learning machines as there were individual words.

We used the simple Bayes and support vector machine methods as the machine learning method.[1] In this section, we explain each of the machine learning methods and then explain the method combining several of them.

3.1 Simple Bayes Method

This method estimates probability based on the Bayes theory. The category (i.e., the sense tag) with the highest probability is judged to be the desired one. This is a basic approach to the disambiguation of word sense. The probability of category a appearing in context b is defined as:

$$p(a|b) = \frac{p(a)}{p(b)} p(b|a) \quad (1)$$

$$\simeq \frac{\tilde{p}(a)}{p(b)} \prod_i \tilde{p}(f_i|a), \quad (2)$$

where context b is a set of features $f_j (\in F, 1 \leq j \leq k)$ that is defined in advance. $p(b)$ is the probability of context b, which is not calculated because it is a constant and is not dependent on category a. $\tilde{p}(a)$ and $\tilde{p}(f_i|a)$ are the probabilities estimated by using the training data and indicate the probability of the occurrence of category a in the examples of the training data and the probability of feature f_i occurring, given category a, respectively. When we use the maximum likelihood estimation to calculate $\tilde{p}(f_i|a)$, which often has a value of 0 and is therefore difficult to estimate the desired category, smoothing process is used. We used this

[1] We made preliminary experiments using various methods: the simple Bayes, the decision list, the maximum entropy, and the support vector machine. The results showed that the simple Bayes and support vector machine methods were better than the other two (Murata et al., 2001). We used these two methods in the contest.

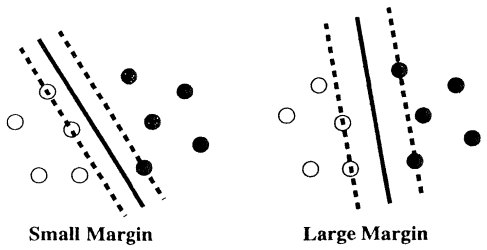

Figure 1: Maximizing the margin

equation for smoothing:

$$\tilde{p}(f_i|a) = \frac{freq(f_i, a) + \epsilon * freq(a)}{freq(a) + \epsilon * freq(a)}, \quad (3)$$

where $freq(f_i, a)$ is the number of events that have the feature f_i and whose category is a and $freq(a)$ is the number of events whose category is a. ϵ is a constant set by experimentation. In this study, we used 0.01 and 0.0001 as ϵ.[2]

3.2 Support Vector Machine Method

In this method, data consisting of two categories is classified by using a hyperplane to divide a space. When the two categories are, for example, positive and negative, enlarging the margin between the positive and negative examples in the training data (see Figure 1[3]) reduces the possibility of incorrectly choosing categories in test data. The hyperplane that maximizes the margin is thus determined, and classification is carried out using that hyperplane. Although the basics of this method are the same as those described above, in the extended versions of the method, the region between the margins through the training data can include a small number of examples, and the linearity of the hyperplane can be changed to a non-linearity by using kernel functions. The classification in the extended versions is equivalent to the classification using the following function (Equation (4)), and the two categories can be classified on the basis of whether the value output by the function is positive or negative (Cristianini and Shawe-Taylor, 2000; Kudoh, 2000):

$$f(\mathbf{x}) = sgn\left(\sum_{i=1}^{l} \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) + b\right) \quad (4)$$

$$b = \frac{max_{i,y_i=-1} b_i + min_{i,y_i=1} b_i}{2}$$

$$b_i = -\sum_{j=1}^{l} \alpha_j y_j K(\mathbf{x}_j, \mathbf{x}_i),$$

where $\mathbf{x}$ is the context (a set of features) of an input example, $\mathbf{x}_i$ indicates the context of a training datum, y_i ($i = 1, ..., l, y_i \in \{1, -1\}$) indicates its category, and the function sgn is

$$sgn(x) = \begin{array}{ll} 1 & (x \geq 0), \\ -1 & (otherwise). \end{array} \quad (5)$$

Each α_i ($i = 1, 2...$) is fixed as the value of α_i that maximizes the value of $L(\alpha)$ in Equation (6) under the conditions set by Equations (7) and (8).

$$L(\alpha) = \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{l} \alpha_i \alpha_j y_i y_j K(\mathbf{x_i}, \mathbf{x_j}) \quad (6)$$

$$0 \leq \alpha_i \leq C \ (i = 1, ..., l) \quad (7)$$

$$\sum_{i=1}^{l} \alpha_i y_i = 0 \quad (8)$$

Function K is called a kernel function and various functions are used as kernel functions. We have used the following polynomial function exclusively.

$$K(\mathbf{x}, \mathbf{y}) = (\mathbf{x} \cdot \mathbf{y} + 1)^d \quad (9)$$

C and d are constants set by experimentation. For all of the experiments reported in this paper, C was fixed as 1 and d was fixed as 2.

A set of $\mathbf{x}_i$ that satisfies $\alpha_i > 0$ is called a support vector (SV_s)[4]. The summation portion of Equation (4) was calculated using only the examples that were support vectors.

Support vector machine methods are capable of handling data consisting of two categories. In general, data consisting of more than two categories is handled by using the pair-wise method (Kudoh and Matsumoto, 2000).

In this method, for data consisting of N categories, pairs of two different categories (N(N-1)/2 pairs) are constructed. The better cate-

[2] In the SENSEVAL-2 contest, we used 0.01 as ϵ. After the contest, we tested several values (0.1 to 0.00000001) as ϵ. We confirmed that $\epsilon = 0.0001$ produced the best results using 10-fold cross validation in the training data.

[3] In the figure, the white and black circles indicate positive and negative examples, respectively. The solid line indicates the hyperplane that divides the space, and the broken lines indicate the planes that mark the margins.

[4] In Figure 1, the circles in the broken lines indicate support vectors.

gory is determined by using a 2-category classifier (in this paper, a support vector machine[5] was used as the 2-category classifier), and the correct category is finally determined by "voting" on the N(N-1)/2 pairs that result from analysis using the 2-category classifier.

The support vector machine method is, in fact, performed by combining the support vector machine and pair-wise methods described above.

3.3 Combined Method

Our combined method changed the used machine-learning method for each word. The used method for each word was the best one for the word in the 10-fold cross validation[6] on the training data among the given methods for combination.

We used the following three kinds of combinations.

- Combined method 1
  a combination of the simple Bayes and support vector machine methods
- Combined method 2
  a combination of two kinds of the simple Bayes method and two kinds of the support vector machine method
  (Here, "the two kinds" indicate an instance where all features were used and where the syntactic feature alone were not).[7]
- Combined method 3
  a combination of two kinds of the simple Bayes method
  (Here, "the two kinds" indicate instance where $\epsilon = 0.0001$ and another where $\epsilon = 0.01$).

4 Features (information used in classification)

In this paper, the following are defined as features.

- **Features based on strings**
  - strings in the analyzed morpheme
  - strings of 1 to 3-grams just before the analyzed morpheme
  - strings of 1 to 3-grams just after the analyzed morpheme
- **Features based on the morphological information given by the RWC tags**
  - the part of speech (POS), the minor POS, and the more minor POS of the analyzed morpheme [8]
  - the previous morpheme, its 5-digit category number, its 3-digit category number, its POS, its minor POS, and its more minor POS[9]
  - the next morpheme, its 5-digit category number, its 3-digit category number, its POS, its minor POS, and its more minor POS
- **Features based on the morphological information given by JUMAN**
  The corpus was analyzed using the Japanese morphological analyzer, JUMAN (Kurohashi and Nagao, 1998), and the results were used as features.
  - the POS, the minor POS, and the more minor POS of the analyzed morpheme, which were determined from the results of JUMAN.
  - the previous morpheme, its 5-digit category number, its 3-digit category number, its POS, its minor POS, and its more minor POS
  - the next morpheme, its 5-digit category number, its 3-digit category number, its POS, its minor POS, and its more minor POS
- **Features based on syntactic information**
  The corpus was analyzed using the Japanese syntactic analyzer KNP (Kurohashi, 1998), and the results were used as features.
  - the *bunsetsu*,[10] including the analyzed morpheme information on whether or not

[5] We used Kudoh's TinySVM software (Kudoh, 2000) as the support vector machine.

[6] In the 10-fold cross validation, we first divide the training data into ten parts. The answers of the instances in each part are estimated by using the instances in the remaining nine parts as the training data. We then use all the results in the ten parts for evaluation.

[7] We used a case where the syntactic feature alone was not used because it obtained a higher precision than when all the features had been used in our preliminary experiments.

[8] The POS, the minor POS, and the more minor POS of a morpheme are the items in the third, fourth, and fifth fields of the RWC corpus, respectively.

[9] A Japanese thesaurus, the *Bunrui Goi Hyou* dictionary (NLRI, 1964), was used to determine the category number of each morpheme. This thesaurus is of the 'is-a' hierarchical type, in which each word has *a category number*, which is a 10-digit number that indicates seven levels of an 'is-a' hierarchy. The top five levels are expressed by the first five digits, the sixth level is expressed by the next two digits, and the final level is expressed by the final three digits.

[10] *Bunsetsu* is a Japanese grammatical term. A bunsetsu is similar to a phrase in English, but is a slightly smaller component. *Eki-de* "at the station" is a bunsetsu, and *sono*, which corresponds to "the" or "its," is also a bunsetsu. A bunsetsu is, roughly, a unit of items that refers to entities.

Table 1: Experimental results

Method	Precision
Baseline method	0.726
Support vector machine (CRL1)	0.783
Simple Bayes method, $\epsilon = 0.01$ (CRL2)	0.778
Simple Bayes method, $\epsilon = 0.0001$	0.790
Combined method 1 (CRL3)	0.786
Combined method 2 (CRL4)	0.786
Combined method 3	0.793
The best method in the contest	0.786

the bunsetsu was a noun phrase, the POS of the bunsetsu's particle, the minor POS of the particle, and the more minor POS of the particle

- the main word that the bunsetsu modifies, including the analyzed morpheme and its 5-digit category number, 3-digit category number, POS, minor POS, and more minor POS
- the main words of the modifiers of the bunsetsu including the analyzed morpheme and their 5-digit category numbers, 3-digit category numbers, POSs, minor POSs, and more minor POSs (In this case, the information on the particle, such as *ga* or *o*, was used as well).

- **Features of all words co-occurring in the same sentence**
  The corpus was analyzed using the Japanese morphological analyzer JUMAN (Kurohashi and Nagao, 1998), and lists of the results were used as features.
  - each morphology in the same sentence, its 5-digit category number, and its 3-digit category number

- **Features of the UDC code in a document**
  In the RWC corpus, each document has a universal decimal code (UDC), indicating its category.
  - the first digit, the first two-digits, and the first three-digits of the UDC in the document

5 Experiments

We submitted the four systems (CRL1 to CRL4), the support vector machine method, the simple Bayes method ($\epsilon = 0.01$), Combined method 1, and Combined method 2. After the contest, we carried out the experiments using the simple Bayes ($\epsilon = 0.0001$) and Combined method 3. Their experimental results are shown in Table 1. "Baseline method" selected the category that most frequently occurred in the training data as the answer. "The best method in the contest" was the best among all the systems submitted to the contest, which was CRL4 (0.786483). The precisions shown in the table are the mixed-grained scores calculated by software "scorer2", which was given by the committees of SENSEVAL-2. (In our systems, all the instances were attempted, so the recall rate was equal to its precision rate.)

We found the following items from the results.

- All the methods produced higher precision than the baseline method.
- Among the four submitted systems (CRL1 to CRL4), Combined method 2 was the best.
- The simple Bayes method using $\epsilon = 0.0001$ and Combined method 3 (the combination of the two simple Bayes methods) obtained higher precision. This indicates that the simple Bayes method was effective.

6 Conclusion

Our methods combining the simple Bayes and support vector machine methods obtained the best precision among all the submitted systems. After the contest, we tuned the parameter used in the simple Bayes method using the 10-fold cross validation in the training data, and it obtained higher precision. The best method was the combination of the two simple Bayes, whose precision was 0.793.

References

Nello Cristianini and John Shawe-Taylor. 2000. *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods.* Cambridge University Press.

Taku Kudoh and Yuji Matsumoto. 2000. Use of support vector learning for chunk identification. *CoNLL-2000.*

Taku Kudoh. 2000. TinySVM: Support Vector Machines. http://cl.aist-nara.ac.jp/ taku-ku// software/TinySVM/ index.html.

Sadao Kurohashi and Makoto Nagao, 1998. *Japanese Morphological Analysis System JUMAN version 3.5.* Department of Informatics, Kyoto University. (in Japanese).

Sadao Kurohashi, 1998. *Japanese Dependency/Case Structure Analyzer KNP version 2.0b6.* Department of Informatics, Kyoto University. (in Japanese).

Masaki Murata, Masao Utiyama, Kiyotaka Uchimoto, Qing Ma, and Hitoshi Isahara. 2001. Experiments on word sense disambiguation using several machine-learning methods. In *IEICE-WGNLC2001-2.* (in Japanese).

NLRI. 1964. *Bunrui Goi Hyou.* Shuuei Publishing.

Kiyoaki Shirai, Wakako Kashino, Minako Hashimoto, Takenobu Tokunaga, Eiichi Arita, Hitoshi Isahara, Shiho Ogino, Ryuichi Kobune, HIronobu Takahashi, Katashi Nagao, Kôiti Hasida, and Masaki Murata. 2001. Text database with word sense tags defined by Iwanami Japanese dictionary. *Information Processing Society of Japan, WGNL 141-19.* (in Japanese).