Unsupervised learning of word sense disambiguation rules by estimating an optimum iteration number in the EM algorithm

Hiroyuki Shinnou
Department of Systems Engineering,
Ibaraki University
4-12-1 Nakanarusawa, Hitachi, Ibaraki
316-8511 JAPAN
shinnou@dse.ibaraki.ac.jp

Minoru Sasaki
Department of Computer and
Information Sciences,
Ibaraki University
4-12-1 Nakanarusawa, Hitachi, Ibaraki
316-8511 JAPAN
sasaki@cis.ibaraki.ac.jp

Abstract

In this paper, we improve an unsupervised learning method using the Expectation-Maximization (EM) algorithm proposed by Nigam et al. for text classification problems in order to apply it to word sense disambiguation (WSD) problems. The improved method stops the EM algorithm at the optimum iteration number. To estimate that number, we propose two methods. In experiments, we solved 50 noun WSD problems in the Japanese Dictionary Task in SENSEVAL2. The score of our method is a match for the best public score of this task. Furthermore, our methods were confirmed to be effective also for verb WSD problems.

1 Introduction

In this paper, we improve an unsupervised learning method using the Expectation-Maximization (EM) algorithm proposed by (Nigam et al., 2000) for text classification problems in order to apply it to word sense disambiguation (WSD) problems. The original method works well, but often causes worse classification for WSD. To avoid this, we propose two methods to estimate the optimum iteration number in the EM algorithm.

Many problems in natural language processing can be converted into classification problems, and be solved by an inductive learning method. This strategy has been very successful, but it has a serious problem in that an inductive learning method requires labeled data, which is expensive because it must be made manually. To overcome this problem, unsupervised learning methods using huge unlabeled data to boost the performance of rules learned by small labeled data have been proposed recently(Blum and Mitchell, 1998)(Yarowsky, 1995)(Park et al., 2000)(Li and Li, 2002). Among these methods, the method using the EM algorithm proposed by the paper(Nigam et al., 2000), which is referred to as the *EM method* in this paper, is the state of the art. However, the target of the EM method is text classification. It is hoped that this method can be applied to WSD, because WSD is the most important problem in natural language processing.

The EM method works well in text classification, but often causes worse classification in WSD. The EM method is expected to improve the accuracy of learned rules step by step in proportion to the iteration number in the EM algorithm. However, this rarely happens in practice, and in many cases, the accuracy falls after a certain iteration number in the EM algorithm. In the worst case, the accuracy of the rule learned through only labeled data is degraded by using unlabeled data. To overcome this problem, we estimate an optimum iteration number in the EM algorithm, and in actual learning, we stop the iteration of the EM algorithm at the estimated number. If the estimated number is 0, it means that the EM method is not used. To estimate the optimum iteration number, we propose two methods: one uses cross validation and the other uses two heuristics besides cross validation. In this paper, we refer to the former method as *CV-EM* and the latter method as *CV-EM2*.

In experiments, we solved 50 noun WSD problems in the Japanese Dictionary Task in SENSEVAL2(Kurohashi and Shirai, 2001). The original EM method failed to boost the precision (76.78%) of the rule learned through only labeled data. On the other hand, CV-EM and CV-EM2 boosted the precision to 77.88% and 78.56%. The score of CV-EM2 is a match for the best public score of this task. Furthermore, these methods were confirmed to be effective also for verb WSD problems.

2 WSD by Naive Bayes

In a classification problem, let $C = \{c_1, c_2, \cdots, c_m\}$ be a set of classes. An instance x is represented as a feature

list

$$x = (f_1, f_2, \cdots, f_n).$$

We can solve the classification problem by estimating the probability $P(c|x)$. Actually, the class c_x of x, is given by

$$c_x = arg \max_{c \in C} P(c|x).$$

Bayes theorem shows that

$$P(c|x) = \frac{P(c)P(x|c)}{P(x)}.$$

As a result, we get

$$c_x = arg \max_{c \in C} P(c)P(x|c).$$

In the above equation, $P(c)$ is estimated easily; the question is how to estimate $P(x|c)$. Naive Bayes models assume the following:

$$P(x|c) = \prod_{i=1}^{n} P(f_i|c). \quad (1)$$

The estimation of $P(f_i|c)$ is easy, so we can estimate $P(x|c)$(Mitchell, 1997). In order to use Naive Bayes effectively, we must select features that satisfy the equation 1 as much as possible. In text classification tasks, the appearance of each word corresponds to each feature.

In this paper, we use following six attributes (`e1` to `e6`) for WSD. Suppose that the target word is w_i which is the i-th word in the sentence.

e1: the word w_{i-1}
e2: the word w_{i+1}
e3: two content words in front of w_i
e4: two content words behind w_i
e5: thesaurus ID number of e3
e6: thesaurus ID number of e4

For example, we make features from the following sentence [1] in which the target word is *'kiroku'*[2].

kako/saikou/wo/kiroku/suru/ta/.

Because the word to the left of the word *'kiroku'* is *'wo'*, we get `'e1=wo'`. In the same way, we get `'e2=suru'`. Content words to the left of the word *'kiroku'* are the word *'kako'* and the word *'saikou'*. We select two words from them in the order of proximity to the target word. Thus, we get `'e3=kako'` and `'e3=saikou'`. In the same way, we get `'e4=suru'` and `'e4=.'`. Note that the comma and the period are defined as a kind of content words in this paper. Next we look up the thesaurus ID of the word *'saikou'*, and find `3.1920_4` [3]. In our thesaurus, as shown in Figure 1, a higher number corresponds to a higher level meaning.

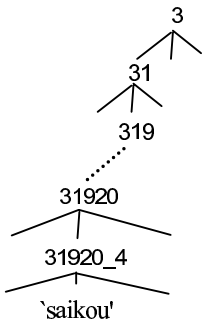

Figure 1: Japanese thesaurus: Bunrui-goi-hyou

In this paper, we use a four-digit number and a five-digit number of a thesaurus ID. As a result, for `'e3=saikou'`, we get `'e5=3192'` and `'e5=31920'`. In the same way, for `'e3=kako'`, we get `'e5=1164'` and `'e5=11642'`. Following this procedure, we should look up the thesaurus ID of the word *'suru'*. However, we do not look up the thesaurus ID for a word that consists of *hiragana* characters, because such words are too ambiguous, that is, they have too many thesaurus IDs. When a word has multiple thesaurus IDs, we create a feature for each ID.

As a result, we get following ten features from the above example sentence:

```
e1=wo, e2=suru, e3=saikou, e3=kako,
e4=suru, e4=., e5=3192, e5=31920,
e5=1164, e5=11642.
```

3 Unsupervised learning using EM algorithm

We can use the EM method if we use Naive Bayes for classification problems. In this paper, we show only key equations and the key algorithm of this method(Nigam et al., 2000).

Basically the method computes $P(f_i|c_j)$ where f_i is a feature and c_j is a class. This probability is given by[4]

$$P(f_i|c_j) = \frac{1 + \sum_{k=1}^{|D|} N(f_i, d_k)P(c_j|d_k)}{|F| + \sum_{m=1}^{|F|} \sum_{k=1}^{|D|} N(f_m, d_k)P(c_j|d_k)}. \quad (2)$$

[1] A sentence is segmented into words, and each word is transformed to its original form by morphological analysis.

[2] *'kiroku'* has at least two meanings: 'memo' and 'record'.

[3] In this paper we use the *bunrui-goi-hyou* as a Japanese thesaurus.

[4] This equation is smoothed by taking into account the frequency 0.

D: all data consisting of labeled data and unlabeled data
d_k: an element in D
F: the set of all features
f_m: an element in F
$N(f_i, d_k)$: the number of f_i in the instance d_k.

In our problem, $N(f_i, d_k)$ is 0 or 1, and almost all of them are 0. If d_k is labeled, $P(c_j|d_k)$ is 0 or 1. If d_k is unlabeled, $P(c_j|d_k)$ is initially 0, and is updated to an appropriate value step by step in proportion to the iteration of the EM algorithm.

By using equation 2, the following classifier is constructed:

$$P(c_j|d_i) = \frac{P(c_j)\prod_{f_n \in K_{d_i}} P(f_n|c_j)}{\sum_{r=1}^{|C|} P(c_r)\prod_{f_n \in K_{d_i}} P(f_n|c_r)}. \quad (3)$$

In this equation, K_{d_i} is the set of features in the instance d_i.

$P(c_j)$ is computed by

$$P(c_j) = \frac{1 + \sum_{k=1}^{|D|} P(c_j|d_k)}{|C| + |D|}. \quad (4)$$

The EM algorithm computes $P(c_j|d_i)$ by using equation 3 (E-step). Next, by using equation 2, $P(f_i|c_j)$ is computed (M-step). By iterating E-step and M-step, $P(f_i|c_j)$ and $P(c_j|d_i)$ converge. In our experiment, when the difference between the current $P(f_i|c_j)$ and the updated $P(f_i|c_j)$ comes to less than $8 \cdot 10^{-6}$ or the iteration number reaches 10 times, we judge that the algorithm has converged.

4 Estimation of the optimum iteration number

In this paper, we propose two methods (CV-EM and CV-EM2) to estimate the optimum iteration number in the EM algorithm.

The CV-EM method is cross validation. First of all, we divide labeled data into three parts, one of which is used as test data and the others are used as new labeled data. By using this new labeled data and huge unlabeled data, we conduct the EM method. After each iteration in the EM algorithm, the learned rules at the time are evaluated by using test data. This experiment is conducted three times by changing the labeled data and test data. The precision of each iteration number is given by the mean of three experiments. The optimum iteration number is estimated to be the iteration number at which the highest precision is achieved.

The CV-EM2 method also uses cross validation, but estimates the optimum iteration number by ad-hoc mechanism.

First, we judge whether we can use the EM method without modification or not. To do this, we compare the precision at convergence with the precision of the iteration number 1. If the former is higher than the latter, we judge that we can use the EM method without modification. In this case, the optimum iteration number is estimated to be the converged number. On the other hand, if the former is not higher than the latter, we go to the second judgment, namely whether the EM method should be used or not. To judge this, we compare the two precisions of the iteration number 0 and 1. The iteration number 0 means that the EM method is not used. If the precision of the iteration number 0 is higher than the precision of the iteration number 1, we judge that the EM method should not be used. In this case, the optimum iteration number is estimated to be 0. Conversely, if the precision of the iteration number 1 is higher than the precision of the iteration number 0, we judge that the EM method should be used. In this case, the optimum iteration number is estimated to be the number obtained by CV-EM.

In the many cases, the CV-EM2 outputs the same number as the CV-EM. However, the basic idea is different. Roughly speaking, the CV-EM2 relies on two heuristics: (1) Basically we only have to judge whether the EM method can be used or not, because the EM algorithm improves or degrades the precision monotonically. (2) Whether the EM algorithm succeeds correlates closely with whether the precision is improved by the first iteration of the EM algorithm. Therefore, we estimate the optimum iteration number by comparing three precisions, the precision of the iteration number 0, 1 and at convergence.

The figure 2 shows a typical case that the CV-EM2 differs from the CV-EM. In the cross validation, the precision is degraded by the first iteration of the EM algorithm, and then it is improved by iteration, and the maximum precision is achieved at the k-th iteration, but the precision converges to the lower point than the precision of the iteration number 1. In this case, the CV-EM gives k as the estimation, but the CV-EM2 gives 0.

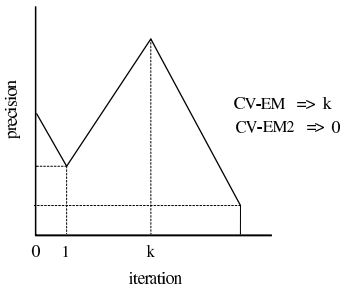

Figure 2: Typical difference between CV-EM and CV-EM2

5 Experiments

To confirm the effectiveness of our methods, we tested with 50 nouns of the Japanese Dictionary Task in SENSEVAL2(Kurohashi and Shirai, 2001).

The Japanese Dictionary Task is a standard WSD problem. As the evaluation words, 50 noun words and 50 verb words are provided. These words are selected so as to balance the difficulty of WSD. The number of labeled instances for nouns is 177.4 on average, and for verbs is 172.7 on average. The number of test instances for each evaluation word is 100, so the number of test instances of noun and verb evaluation words is 5,000 respectively. However, unlabeled data are not provided. Note that we cannot use simple raw texts including the target word, because we must use the same dictionary and part of speech set as labeled data. Therefore, we use Mainichi newspaper articles for 1995 with word segmentations provided by RWC. This data is the origin of labeled data. As a result, we gathered 7585.5 and 6571.9 unlabeled instances for per noun and per verb evaluation word on average, respectively.

Table 1 shows the results of experiments for noun evaluation words. In this table, `NB` means Naive Bayes, `EM` the EM method, and `ideal` the EM method stopping at the ideal iteration number. Note that the precision is computed by mixed-gained scoring(Kurohashi and Shirai, 2001) which gives partial points in some cases.

The precision of Naive Bayes which learns through only labeled data was 76.58%. The EM method failed to boost it, and degraded it to 73.56%. On the other hand, by using CV-EM the precision was boosted to 77.88%. Furthermore, CV-EM2 boosted it to 78.56%. This score is a match for the best public score of this task. As successful results in this task, two researches are reported. One used Naive Bayes with various attributes, and achieved 78.22% precision(Murata et al., 2001). Another used Adaboost of decision trees, and achieved 78.47% precision(Nakano and Hirai, 2002). Our score is higher than these scores [5]. Furthermore, their methods used syntactic analysis, but our methods do not need it.

In the same way, we performed experiments for verb evaluation words. Table 2 shows the results. In the experiment, Naive Bayes achieved 78.16% precision. The EM method boosted it to 78.74%. Furthermore, CV-EM and CV-EM2 boosted it to 79.22% and 79.26% respectively. CV-EM2 is marginally higher than CV-EM.

Table 1: Results of experiments (Noun)

Word	NB (%)	EM (%)	CV-EM (%)	CV-EM2 (%)	ideal (%)
aida	81.0	80.0	82.0	82.0	82.0
atama	60.0	64.0	60.0	64.0	66.0
ippan	88.0	86.0	89.0	89.0	89.0
ippou	82.0	88.0	88.0	88.0	89.0
ima	90.0	90.0	90.0	90.0	90.0
imi	45.0	53.0	53.0	53.0	53.0
utagai	100.0	95.0	98.0	98.0	100.0
otoko	92.0	89.0	92.0	92.0	92.0
kaihatsu	62.0	63.0	62.0	62.0	63.0
kaku_n	71.0	77.0	71.0	77.0	81.0
kankei	85.0	90.0	90.0	90.0	90.0
kimochi	65.0	65.0	65.0	65.0	66.0
kiroku	74.0	71.0	73.0	73.0	77.0
gijutsu	96.0	92.0	96.0	96.0	96.0
genzai	97.0	09.0	98.0	98.0	98.0
koushou	100.0	88.0	100.0	100.0	100.0
kokunai	46.0	58.0	46.0	46.0	58.0
kotoba	45.0	40.0	40.0	40.0	45.0
kodomo	67.0	73.0	72.0	72.0	73.0
gogo	77.0	65.0	86.0	86.0	86.0
shijo	77.0	55.0	77.0	77.0	77.0
shimin	67.0	63.0	67.0	67.0	67.0
shakai	82.0	83.0	83.0	83.0	83.0
shonen	92.0	90.0	90.0	90.0	92.0
jikan	54.0	15.0	54.0	54.0	54.0
jigyou	69.0	70.0	69.0	69.0	71.0
jidai	72.0	77.0	77.0	77.0	78.0
jibun	100.0	100.0	100.0	100.0	100.0
joho	77.0	64.0	77.0	77.0	77.0
sugata	55.0	63.0	61.0	61.0	63.0
seishin	65.0	66.0	66.0	66.0	66.0
taishou	98.0	98.0	98.0	98.0	98.0
daihyou	85.0	95.0	96.0	96.0	98.0
chikaku	74.0	87.0	87.0	87.0	87.0
chihou	70.0	72.0	70.0	70.0	72.0
chushin	98.0	98.0	98.0	98.0	98.0
te	47.0	48.0	47.0	48.0	48.0
teido	100.0	100.0	100.0	100.0	100.0
denwa	84.0	65.0	83.0	83.0	85.0
doujitsu	81.0	51.0	57.0	81.0	81.0
hana	99.0	97.0	99.0	99.0	99.0
hantai	97.0	97.0	97.0	97.0	97.0
baai	82.0	91.0	91.0	91.0	92.0
mae	86.0	91.0	92.0	91.0	92.0
minkan	100.0	100.0	100.0	100.0	100.0
musume	88.0	88.0	88.0	88.0	88.0
mune	71.0	77.0	77.0	77.0	79.0
me	18.0	17.0	18.0	18.0	18.0
mono	31.0	27.0	27.0	27.0	31.0
mondai	97.0	97.0	97.0	97.0	97.0
average	76.78	73.56	77.88	78.56	79.64

[5]The best score for the total of noun words and verb words is reported to be 79.33% in (Murata et al., 2001).

Table 2: Results of experiments (Verb)

Word	NB (%)	EM (%)	CV-EM (%)	CV-EM2 (%)	ideal (%)
ataeru	71.0	78.0	78.0	78.0	78.0
iu	94.0	94.0	94.0	94.0	94.0
ukeru	59.0	64.0	59.0	64.0	64.0
uttaeru	84.0	87.0	87.0	87.0	88.0
umareru	69.0	83.0	82.0	83.0	83.0
egaku	58.0	56.0	56.0	56.0	58.0
omou	90.0	89.0	89.0	89.0	90.0
kau	83.0	83.0	83.0	83.0	83.0
kakaru	58.0	57.0	58.0	58.0	58.0
kaku_v	72.0	66.0	72.0	72.0	72.0
kawaru	92.0	92.0	92.0	92.0	92.0
kangaeru	99.0	99.0	99.0	99.0	99.0
kiku	56.0	55.0	55.0	55.0	56.0
kimaru	96.0	96.0	96.0	96.0	96.0
kimeru	93.0	93.0	93.0	93.0	93.0
kuru	84.0	85.0	86.0	85.0	86.0
kuwaeru	89.0	89.0	89.0	89.0	89.0
koeru	78.0	82.0	85.0	82.0	88.0
shiru	97.0	97.0	97.0	97.0	97.0
susumu	49.0	50.0	50.0	50.0	50.0
susumeru	97.0	95.0	97.0	97.0	97.0
dasu	35.0	29.0	35.0	35.0	36.0
chigau	100.0	100.0	100.0	100.0	100.0
tsukau	97.0	97.0	97.0	97.0	97.0
tsukuru	69.0	75.0	78.0	75.0	78.0
tsutaeru	75.0	76.0	76.0	76.0	76.0
dekiru	81.0	81.0	81.0	81.0	81.0
deru	59.0	64.0	64.0	64.0	64.0
tou	69.0	79.0	79.0	79.0	79.0
toru	32.0	34.0	32.0	34.0	37.0
nerau	99.0	99.0	99.0	99.0	99.0
nokosu	79.0	79.0	79.0	79.0	79.0
noru	54.0	54.0	54.0	54.0	54.0
hairu	36.0	36.0	36.0	36.0	36.0
hakaru	92.0	92.0	92.0	92.0	92.0
hanasu	100.0	87.0	100.0	100.0	100.0
hiraku	86.0	94.0	94.0	94.0	94.0
fukumu	99.0	99.0	99.0	99.0	99.0
matsu	52.0	50.0	51.0	51.0	52.0
matomeru	79.0	80.0	80.0	80.0	80.0
mamoru	79.0	71.0	70.0	71.0	79.0
miseru	98.0	98.0	98.0	98.0	98.0
mitomeru	89.0	89.0	89.0	89.0	89.0
miru	73.0	71.0	73.0	73.0	73.0
mukaeru	89.0	89.0	89.0	89.0	89.0
motsu	57.0	62.0	57.0	57.0	62.0
motomeru	87.0	87.0	87.0	87.0	87.0
yomu	88.0	88.0	88.0	88.0	88.0
yoru	97.0	97.0	97.0	97.0	97.0
wakaru	90.0	90.0	90.0	90.0	90.0
average	78.16	78.74	79.22	79.26	79.92

6 Discussion

6.1 Cause of failure of the EM method

Why does the EM method often fail to boost the performance? One reason may be the difference among class distributions of labeled data L, unlabeled data U and test data T. Practically L, U and T are the same because they consist of random samples from all data. However, there are differences among them.

Intuitively, learning by combining labeled data and unlabeled data is regarded as learning from the distribution of $L+U$. It is expected that the EM method is effective if $d = d(L,T) - d(L+U,T) > 0$, and is counterproductive if $d < 0$, in which $d(\cdot,\cdot)$ means the distance of two distributions.

To confirm the above expectation, we conduct an experiment by using Kullback-Leibler divergence as $d(\cdot,\cdot)$. The distribution of $L+U$ can be obtained from Equation 4 when the EM algorithm converges. The result of the experiment is shown in Table 3.

Table 3: Effects of the distribution of meanings

	$d>0$	$d<0$
improvement	6	7
deterioration	2	8

The columns of the table are divided into positive ($d>0$) and negative ($d<0$). Positive means that $L+U$ gets close to T and negative means that $L+U$ goes away from T. The rows of the table are divided into improvement of precision and deterioration of precision. In this paper, improvement of precision is when the precision is improved by over 5%, and deterioration of precision is when the precision is degraded by over 5%.

This result indicates that there is a weak correlation between whether $L+U$ gets close to T or goes away from T and whether the EM method is effective or not, but we cannot conclude they are completely dependent. However, the evaluation word *'genzai'* whose precision falls most by the EM method is precisely the above case. The d for this word is the smallest, –0.30, among all evaluation words. Further investigation of the causes of failure of the EM method is our future work.

6.2 Effectiveness of estimation of CV-EM2

CV-EM2 achieved ideal estimation for 29 of 50 evaluation words, that is 58%. Furthermore, for 15 of the other 21 evaluation words, the difference between the precision through our method and that through ideal estimation did not exceed 2%. Therefore, estimation of CV-EM2 is mostly effective.

The words *'kokunai'* and *'kotoba'* are typical cases where estimation fails. The difference between the precision of CV-EM2 and that through ideal estimation exceeded 5%. The failure of estimation for these two words reduced the whole precision.

Figure 3 compares the precision for cross validation and that for actual evaluation for the word *'kokunai'*. In the same way, Figure 3 shows the case of the word *'kotoba'*. In these figures, the x-axis shows the iteration number of the EM algorithm. To clarify the change of precision, the initial precision is set to 0, and the y-axis shows the difference (%) between the actual and initial precision.

In the case of *'kokunai'*, the precision got worse in cross validation, but the precision got better in the actual evaluation. This means that cross validation is use-

less, so it is difficult to estimate an optimum iteration number in the EM algorithm. However, such cases are rare. In the experiment, this case arises for only this word *'kokunai'*. Consider next the case of *'kotoba'*. In cross validation, the precision improved in the first iteration of the EM algorithm, but got worse step by step thereafter. On the other hand, in the actual evaluation, the precision got worse even in the first iteration of the EM algorithm. The difference of these results in the first iteration of the EM algorithm causes our estimation to fail. In future, we must improve our method by further investigation of these words.

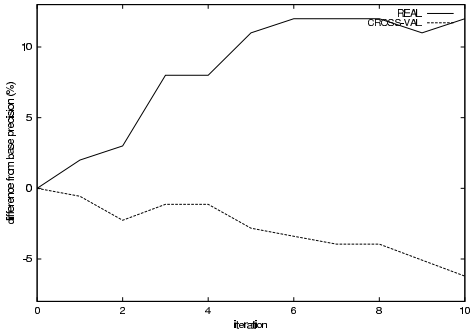

Figure 3: Comparison between cross validation and actual evaluation (*'kokunai'*)

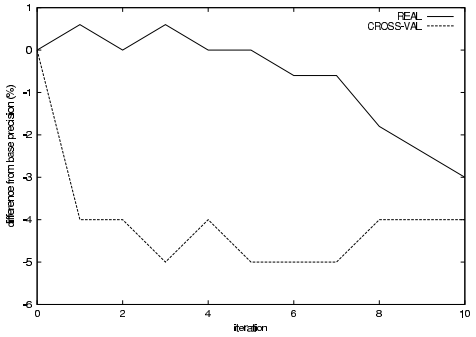

Figure 4: Comparison between cross validation and actual evaluation (*'kotoba'*)

6.3 Comparison of CV-EM and CV-EM2

CV-EM2 is slightly superior to CV-EM. In the evaluation word *'doujitu'*, there is a remarkable difference between the two methods.

Figure 5 shows the change of the precision for *'doujitsu'* in cross validation, and Figure 6 shows that in actual evaluation.

The precision goes up in cross validation, but goes down largely in actual evaluation. In CV-EM, the best point is selected in cross validation, that is 3. On the other hand, CV-EM2 estimates 0 by using the relation of three precisions: the initial precision, the precision for the iteration 1 and the precision at convergence.

Let's count the number of words for which CV-EM2 is better or worse than CV-EM. For one word *'mae'* in nouns and three words *'kuru'*, *'koeru'* and *'tukuru'* in verbs, CV-EM was superior to CV-EM2. On the other hand, for four words *'atama'*, *'kaku_n'*, *'te'* and *'doujitsu'* in nouns and four words *'ukeru'*, *'umareru'*, *'toru'* and *'mamortu'* in verbs, CV-EM2 was better to CV-EM. These numbers show that our method is somewhat superior to CV-EM.

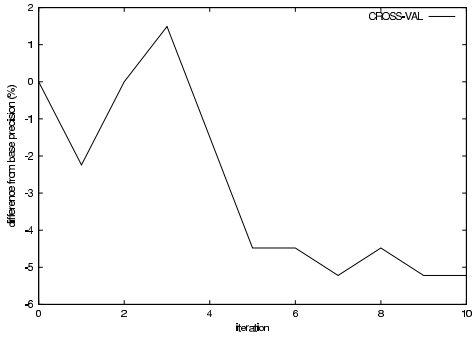

Figure 5: Cross validation in *'doujitsu'*

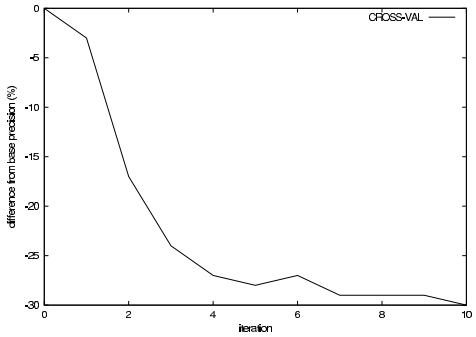

Figure 6: Actual evaluation in *'doujitsu'*

6.4 Unsupervised learning for verb WSD

In the experiments, CV-EM and CV-EM2 improved the EM method for both noun words and verb words. The effectiveness of these methods was large for noun words, but was small for verb words. We believe that the cause of this difference is the difficulty of unsupervised learning for verb WSD. In ideal estimation, the precision for noun words was boosted from 76.78% to 79.64% by the EM method, that is 1.037 times. On the other hand, the precision for verb words was boosted from 78.16% to 79.92%

by the EM method, that is 1.022 times. This shows that the EM method does not work so well for verb words.

We consider that feature independence plays a key role in unsupervised learning. Suppose the instance x consists of two features f_1 and f_2. When class c_x of x is judged from feature f_1, the probability $P(c_x|f_2)$ is tuned to be larger. The question is whether it is actually right or not to increase $P(c_x|f_2)$. If it is right, unsupervised learning works well, but if it is not, unsupervised learning fails. Intuitively, feature independence warrants increasing $P(c_x|f_2)$. In noun WSD, the left context of the target word corresponds to the words modifying the target word, and the right context of the target word corresponds to the verb word whose case slot can have the target word. Both the left context and right context can judge the meaning of the target word by itself, and are independent. Left context and right context act as independent features. On the other hand, we cannot find such an opportune interpretation for the features of verbs (Shinnou, 2002). Therefore, the EM method is not so effective for verb words.

Naive Bayes assumes the independence of features, too. However, this assumption is not so rigid in practice. We believe that the improvement by the EM method for verb words depends on the robustness of Naive Bayes. In our experiments, the EM method for noun words failed to boost the precision. We think that the cause is the imbalance of labeled data, unlabeled data and test data. We should investigate this in a future study.

6.5 Related works

Co-training(Blum and Mitchell, 1998) is a powerful unsupervised learning method. In Co-training, if we can find two independent feature sets for the target problem, any supervised learning method can be used. Furthermore, it is reported that Co-training is superior to the EM method if complete independent feature sets can be used(Nigam and Ghani, 2000). However, Co-training requires consistency besides independence for two feature sets. This condition makes it difficult to apply Co-training to multiclass classification problems. On the other hand, the EM method requires Naive Bayes to be used as the supervised learning method, but can be applied to multiclass classification problems without any modification. Therefore, the EM method is more practical than Co-training.

Yarowsky proposed the unsupervised learning method for WSD(Yarowsky, 1995). His method is reported to be a special case of Co-training(Blum and Mitchell, 1998). As two independent feature sets, one is the context surrounding the target word and the other is the heuristic of 'one sense per discourse'. However, it is unknown how valid this heuristic is for granularity of meanings of our evaluation words. Furthermore, this method needs documents in which the target word appears multiple times, as unlabeled data. Therefore, it is not so easy to gather unlabeled data. On the other hand, the EM method does not have such problem because it uses sentences including the target word as unlabeled data.

6.6 Future works

We have three future works. First, we must raise the precision for verb words, which may be impossible unless we use other features, so we need to investigate other features. Second, we must improve the estimation method of the optimum iteration number in the EM algorithm. The difference between the precision through our estimation and that through the ideal estimation is large. We can improve the accuracy by improving the estimation method. Finally, we will investigate the reason for the failure of the EM method, which may be the key to unsupervised learning.

7 Conclusions

In this paper, we improved the EM method proposed by Nigam et al. for text classification problems in order to apply it to WSD problems. To avoid some failures in the original EM method, we proposed two methods to estimate the optimum iteration number in the EM algorithm. In experiments, we tested with 50 noun WSD problems in the Japanese Dictionary Task in SENSEVAL2. Our two methods greatly improved the original EM method. Especially, the score of noun evaluation words was equivalent to the best public score of this task. Furthermore, our methods were also effective for verb WSD problems. In future, we will tackle three works: (1) To find other effective features for unsupervised learning of verb WSD, (2) To improve the estimation method of the optimum iteration number in the EM algorithm, and (3) To investigate the reason for the failure of the EM method.

References

Avrim Blum and Tom Mitchell. 1998. Combining Labeled and Unlabeled Data with Co-Training. In *11th Annual Conference on Computational Learning Theory (COLT-98)*, pages 92–100.

Sadao Kurohashi and Kiyoaki Shirai. 2001. SENSEVAL-2 Japanese Tasks (in Japansese). In *Technical Report of IEICE*, NLC-36–48, pages 1–8.

Cong Li and Hang Li. 2002. Word Translation Disambiguation Using Bilingual Bootstrapping. In *40th Annual Meeting of the Association for Computational Linguistics (ACL-02)*, pages 343–351.

Tom Mitchell. 1997. *Machine Learning*. McGraw-Hill Companies.

Masaki Murata, Masao Utiyama, Kiyotaka Uchimoto, Qing Ma, and Hitoshi Isahara. 2001. CRL at Japanese dictionary-based task of SENSEVAL-2 (in Japanese). In *Technical Report of IEICE*, NLC-36–48, pages 31–38.

Keigo Nakano and Yuuzou Hirai. 2002. AdaBoost wo motiita gogi no aimaisei kaisyou (in Japanese). In *8th Annual Meeting of the Association for Natural Language Processing*, pages 659–662.

Kamal Nigam and Rayid Ghani. 2000. Analyzing the effectiveness and applicability of co-training. In *9th International Conference on Information and Knowledge Management*, pages 86–93.

Kamal Nigam, Andrew McCallum, Sebastian Thrun, and Tom Mitchell. 2000. Text Classification from Labeled and Unlabeled Documents using EM. *Machine Learning*, 39(2/3):103–134.

Seong-Bae Park, Byoung-Tak Zhang, and Yung Taek Kim. 2000. Word sense disambiguation by learning from unlabeled data. In *38th Annual Meeting of the Association for Computational Linguistics (ACL-00)*, pages 547–554.

Hiroyuki Shinnou. 2002. Learning of word sense disambiguation rules by Co-training, checking co-occurrence of features. In *3rd international conference on Language resources and evaluation (LREC-2002)*, pages 1380–1384.

David Yarowsky. 1995. Unsupervised Word Sense Disambiguation Rivaling Supervised Methods. In *33th Annual Meeting of the Association for Computational Linguistics (ACL-95)*, pages 189–196.