Selective Sampling of Effective Example Sentence Sets for Word Sense Disambiguation

FUJII Atsushi, INUI Kentaro, TOKUNAGA Takenobu and TANAKA Hozumi

Department of Computer Science
Tokyo Institute of Technology
2-12-1 Oookayama Meguroku Tokyo 152, JAPAN
{fujii,inui,take,tanaka}@cs.titech.ac.jp

Abstract

This paper proposes an efficient example selection method for example-based word sense disambiguation systems. To construct a practical size database, a considerable overhead for manual sense disambiguation is required. Our method is characterized by the reliance on the notion of the training utility: the degree to which each example is informative for future example selection when used for the training of the system. The system progressively collects examples by selecting those with greatest utility. The paper reports the effectivity of our method through experiments on about one thousand sentences. Compared to experiments with random example selection, our method reduced the overhead without the degeneration of the performance of the system.

1 Introduction

Word sense disambiguation is a crucial task in many NLP applications, such as machine translation [1], parsing [14, 16] and text retrieval [10, 23]. Given the growing utilization of machine readable texts, word sense disambiguation techniques have been variously used in corpus-based approaches [1, 3, 5, 12, 18, 20, 21, 24]. Unlike rule-based approaches, corpus-based approaches release us from the task of generalizing observed phenomena in order to disambiguate word senses. Our system is based on such an approach, or more precisely it is based on an example-based approach [5]. Since this approach requires a certain number of examples of disambiguated verbs, we have to carry out this task manually, that is, we disambiguate verbs appearing in a corpus prior to their use by the system. A preliminary experiment on ten Japanese verbs showed that the system needed on average about one hundred examples for each verb in order to achieve 82% of accuracy in disambiguating verb senses. In order to build an operational system, the following problems have to be taken into account:

1. Since there are about one thousand basic verbs in Japanese, a considerable overhead is associated with manual word sense disambiguation.

2. Given human resource limitations, it is not reasonable to manually analyze large corpora as they can provide virtually infinite input.

3. Given the fact that example-based natural language systems, including our system, search the example-database (database, hereafter) for the most similar examples with regard to the input, the computational cost becomes prohibitive if one works with a very large database size [11].

All these problems suggest a different approach, namely to *select* a small number of optimally informative examples from a given corpora. Hereafter we will call these examples "samples."

Our method, based on the utility maximization principle, decides on which examples should be included in the database. This decision procedure is usually called *selective sampling.* Selective sampling directly addresses the first two problems mentioned above. The overall control flow of systems based on selective sampling can be depicted as in figure 1, where "system" refers to dedicated NLP applications. The sampling process basically cycles between the execution and the training phases. During the execution phase, the system generates an interpretation for each example, in terms of parts-of-speech, text categories or word senses. During the training phase, the system selects samples for training from the previously produced outputs. During this phase, a human expert provides the correct interpretation of the samples so that the system can then be trained for the execution of the remaining data. Several researchers have proposed such an approach.

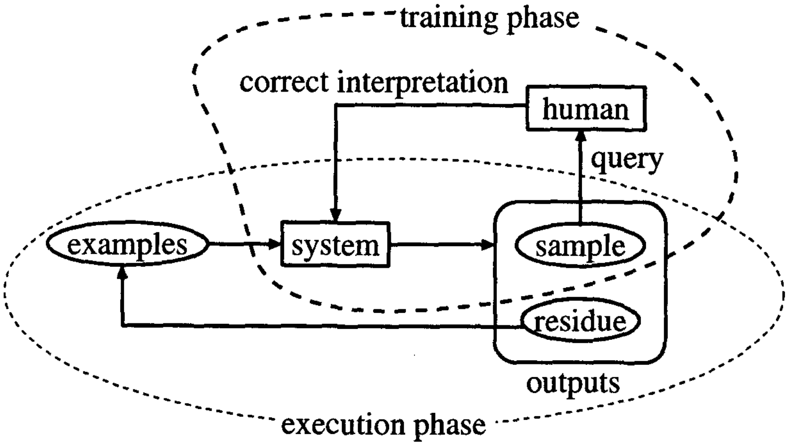

Figure 1: Flow of control of the example sampling system

Lewis et al. proposed an example sampling method for statistics-based text classification [13]. In this method, the system always selects samples which are not certain with respect to the correctness of the answer. Dagan et al. proposed a committee-based sampling method, which is currently applied to HMM training for part-of-speech tagging [2]. This method selects samples based on the training utility factor of the examples, i.e. the informativity of the data with respect to future training. However, as all these methods are implemented for statistics-based models, there is a need to explore how to formalize and map these concepts into the example-based approach.

With respect to problem 3, a possible solution would be the generalization of redundant examples [8, 19]. However, such an approach implies a significant overhead for the manual training of each example prior to the generalization. This shortcoming is precisely what our approach allows to avoid: reducing both the overhead as well as the size of the database.

Section 2 briefly describes our method for a verb sense disambiguation system. The next Section 3 elaborates on the example sampling method, while section 4 reports on the results of our experiment. Before concluding in section 6, discussion is added in section 5.

2 Example-based verb sense disambiguation system

toru:				
suri (pickpocket) *kanojo* (she) *ani* (brother)	*ga*	*kane* (money) *saifu* (wallet) *otoko* (man) *uma* (horse) *aidea* (idea)	*o*	*toru* (to take/steal)
kare (he) *kanojo* (she) *gakusei* (student)	*ga*	*menkyoshô* (license) *shikaku* (qualification) *biza* (visa)	*o*	*toru* (to attain)
kare (he) *chichi* (father) *kyaku* (client)	*ga*	*shinbun* (newspaper) *zasshi* (journal)	*o*	*toru* (to subscribe)
kare (he) *dantai* (group) *ryokôkyaku* (passenger) *joshu* (assistant)	*ga*	*kippu* (ticket) *heya* (room) *hikôki* (airplane)	*o*	*toru* (to reserve)
⋮		⋮		⋮

Figure 2: A fragment of a database, and the entry associated with the Japanese verb *toru*

Our method for disambiguating verb senses uses a database containing examples of collocations for each verb sense and its associated case frame(s). Figure 2 shows a fragment of the entry associated with the Japanese verb *toru*. As with most words, the verb *toru* has multiple senses, a sample of which are "to take/steal," "to attain," "to subscribe" and "to reserve." The database specifies the case frame(s) associated with each verb sense. In Japanese, a complement of a verb consists of a noun phrase (case filler) and its case marker suffix, for example *ga* (nominative) or *o* (accusative). The database lists several case filler examples for each case. The task of the system is "to interpret" the verbs occurring in the input text, i.e. to choose one sense from among a set of candidates. All verb senses we use are defined in "IPAL" [7], a machine readable dictionary. IPAL also contains example case fillers as shown in figure 2. Given an input, in our case a simple sentence, the system identifies the verb sense on the basis of the scored similarity between the input and the examples given for each verb sense. Let us take as an example the sentence below:

hisho ga	*shindaisha o*	*toru.*
(secretary-NOM)	(sleeping car-ACC)	(?)

In this example, one may consider *hisho* ("secretary") and *shindaisha* ("sleeping car") to be semantically similar to *joshu* ("assistant") and *hikôki* ("airplane") respectively, and since both collocate with the "to reserve" sense of *toru* one could infer that *toru* may be interpreted as "to reserve." The similarity between two different case fillers is estimated according to the length of the path between them in a thesaurus. Our current experiments are based around the Japanese word thesaurus *Bunruigoihyo* [17]. Figure 3 shows a fragment of *Bunruigoihyo* including some of the nouns in both figure 2 and the example sentence above, with each word corresponding to a leaf in the structure of the thesaurus. As with most thesauri, the length of the path between two terms in *Bunruigoihyo* is expected to reflect their relative similarity. In table 1, we show our measure of similarity, based on the length of the path between two terms, as proposed by Kurohashi et al [12].

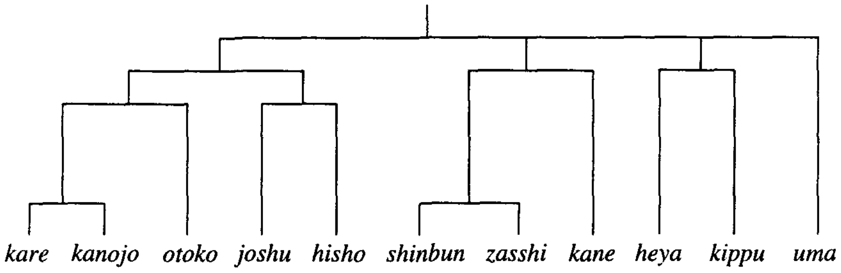

Figure 3: A fragment of *Bunruigoihyo*

Furthermore, since the restrictions imposed by the case fillers in choosing the verb sense are not equally selective, we consider a weighted case contribution to the disambiguation (CCD) of the verb senses. This CCD factor is taken into account when computing the score of a verb's sense. Consider again the case of *toru* in figure 2. Since the semantic range of nouns collocating with the verb in the nominative does not seem to have a strong delinearization in a semantic sense (in figure 2, the nominative of each verb sense displays the same general concept, i.e. animate), it would be difficult, or even risky, to properly interpret the verb sense based on the similarity in the nominative. In contrast, since the ranges are diverse in the accusative, it would be feasible to rely more strongly on the similarity here. This argument can be illustrated as in figure 4, in which the symbols "1" and "2" denote example case fillers of different case frames, and an input sentence includes two case fillers denoted by "x" and "y."

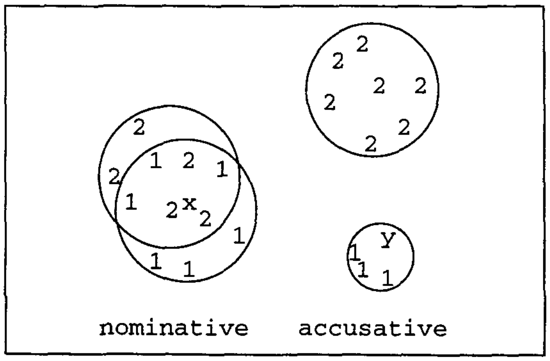

Figure 4: The semantic ranges of the nominative and accusative with verb *toru*

The figure shows the distribution of example case fillers for the respective case frames, denoted in a semantic space. The semantic similarity between two given case fillers is represented by the physical distance between two symbols. In the nominative, since "x" happens to be much closer to a "2" than any "1," "x" may be estimated to belong to the range of "2"s, although "x" actually belongs to both sets of "1"s and "2"s. In the accusative, however, "y" would be properly estimated to belong to the set of "1"s due to the mutual independence of the two accusative case filler sets, even though examples did not fully cover each of the ranges of "1"s and "2"s. Note that this difference would be critical if example data were sparse. We will explain the method used to compute CCD later in this section.

To illustrate the overall algorithm, we will consider an abstract specification of both input and the datatbase (see figure 5). Let the input be $\{n_{c1}\text{-}m_{c1}, n_{c2}\text{-}m_{c2}, n_{c3}\text{-}m_{c3}, v\}$, where n_{c_i} denotes the case filler for the case c_i, and m_{c_i} denotes the case marker for c_i. The interpretation candidates for v are derived from the database as s_1, s_2 and s_3. The database contains also a set $\mathcal{E}_{s_i,c_j}$ of case filler examples for each case c_j of each sense s_i ("—" indicates that the corresponding case is not allowed).

Table 1: The relation between the length of the path between two nouns X and Y ($len(X,Y)$) in *Bunruigoihyo* and their relative similarity ($sim(X,Y)$)

$len(X,Y)$	0	2	4	6	8	10	12
$sim(X,Y)$	11	10	9	8	7	5	0

input	n_{c1}-m_{c1}	n_{c2}-m_{c2}	n_{c3}-m_{c3}		v (?)
database	$\mathcal{E}_{s1,c1}$	$\mathcal{E}_{s1,c2}$	$\mathcal{E}_{s1,c3}$	—	v (s_1)
	$\mathcal{E}_{s2,c1}$	$\mathcal{E}_{s2,c2}$	$\mathcal{E}_{s2,c3}$	$\mathcal{E}_{s2,c4}$	v (s_2)
	—	$\mathcal{E}_{s3,c2}$	$\mathcal{E}_{s3,c3}$	—	v (s_3)

Figure 5: An input and the database

During the verb sense disambiguation process, the system discards first those candidates whose case frame does not fit the input. In the case of figure 5, s_3 is discarded because the case frame of v (s_3) does not subcategorize for the case c_1.

In the next step the system computes the score of the remaining candidates and chooses as the most plausible interpretation the one with the highest score. The score of an interpretation is computed by considering the *weighted* average of the similarity degrees of the input complements with respect to each of the example case fillers (in the corresponding case) listed in the database for the sense under evaluation. Formally, this is expressed by equation (1), where $S(s)$ is the score of the sense s of the input verb, and $SIM(n_c, \mathcal{E}_{s,c})$ is the maximum similarity degree between the input complement n_c and the corresponding complements in the database example $\mathcal{E}_{s,c}$ (equation (2)).

$$S(s) = \frac{\sum_c SIM(n_c, \mathcal{E}_{s,c}) \cdot CCD(c)}{\sum_c CCD(c)} \tag{1}$$

$$SIM(n_c, \mathcal{E}_{s,c}) = \max_{e \in \mathcal{E}_{s,c}} sim(n_c, e) \tag{2}$$

In equation (2), *sim* stands for the similarity degree between n_c and an example case filler e as given by table 1.

$CCD(c)$ expresses the weight factor of the case c contribution to the (current) verb sense disambiguation. Intuitively preference should be given to cases displaying case fillers which are classified in semantic categories of greater independence. Let v be a verb with n senses $(s_1, s_2, \ldots, s_n)$ and let $\mathcal{E}_{s_i,c}$ be the set of example case fillers for the case c, associated with the sense s_i. Then, c's contribution to v's sense disambiguation, $CCD(c)$, is likely to be higher if the example case filler sets $\{\mathcal{E}_{s_i,c} \mid i = 1, \ldots, n\}$ share less elements. The notion of sharing is defined based on the similarity as in equation (3).

$$\{X\} \cup \{Y\} = \{X\} \quad \text{if } sim(X,Y) >= 9 \tag{3}$$

With these definitions, $CCD(c)$ is given by equation (4).

$$CCD(c) = \left(\frac{1}{{}_nC_2}\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}\frac{|\mathcal{E}_{s_i,c}| + |\mathcal{E}_{s_j,c}| - 2|\mathcal{E}_{s_i,c} \cap \mathcal{E}_{s_j,c}|}{|\mathcal{E}_{s_i,c}| + |\mathcal{E}_{s_j,c}|}\right)^{\alpha} \qquad (4)$$

Where α is the constant for parameterizing the extent to which CCD influences verb sense disambiguation. The larger α, the stronger CCD's influence on the system's output.

3 Example sampling algorithm

3.1 Overview

Let us look again at figure 1 in section 1. In this diagram, "outputs" refers to a corpus in which each sentence is assigned the proper interpretation of the verb during the execution phase. In the "training" phase, the system stores samples of manually disambiguated verb senses (simply checked or appropriately corrected by a human) in the database to be later used in a new execution phase. This is the issue we turn to in this section.

Lewis et al. proposed the notion of uncertain example sampling for the training of statistics-based text classifiers [13]. Their method selects those examples that the system classifies (in this case, matching a text category) with minimum certainty. This method is based on the assumption that there is no need for teaching the system the correct answer when it answered with high certainty. However, we should take into account the training effect a given example has on other examples. In other words, by selecting an appropriate example as a sample, we can get more correct examples in the next cycle of iteration. In consequence, the number of examples to be taught will decrease. We consider maximization of this effect by means of a training utility function (TUF) aiming at ensuring that the example with the highest training utility figure, is the most useful example at a given point in time.

Let **S** be a set of sentences, i.e. a given corpus, and **T** be a subset of **S** in which each sentence has already been manually disambiguated for training. In other words, sentences in **T** have been selected as samples, and are hence stored in the database. Let **X** be the set of the residue, realizing equation (5).

$$\mathbf{S} = \mathbf{X} \cup \mathbf{T} \qquad (5)$$

We introduce a utility function $TUF(x)$, which computes the training utility figure for an example x. The sampling algorithm gives preference to examples of maximum utility, by way of equation (6).

$$\arg\max_{x \in \mathbf{X}} TUF(x) \qquad (6)$$

We will explain in the following sections how one could estimate TUF, based on the estimation of the certainty figure of an interpretation. Ideally the sampling size, i.e. the number of samples selected at each iteration would be such as to avoid retraining of similar examples. It should be noted that this can be a critical problem for statistics-based approaches [1, 3, 18, 20, 24], as the reconstruction of statistic classifiers is expensive. However, example-based systems [5, 12, 21] do not require the reconstruction of the system, but examples have to be stored in the database. It also should be noted that in each iteration, the system needs only compute the similarity between each example x belonging to **X** and the newly stored example, instead of every example belonging to **T**, because of the following reasons:

- storing an example of verb sense interpretation s_i, will not affect the score of other verb senses,

- if the system memorizes the current score of s_i for each x, the system simply needs to compare it with the newly computed score between x and the newly stored example in **T** and choose the greater of the two to be the new plausibility of s_i.

This reduces the time complexity of each iteration from $O(N^2)$ to $O(N)$, given that N is the total number of examples in **S**.

3.2 Interpretation certainty

Lewis et al. estimate certainty of an interpretation by the ratio between the probability of the most plausible text category, and the probability of any other text category, excluding the most probable one. Similarly, in our example-based verb sense disambiguation system, we introduce the notion of interpretation certainty of examples based on the following applicability restrictions:

1. the highest interpretation score is sufficiently large,

2. the highest interpretation score is significantly larger than the second highest score.

The rationale for these restrictions is given below. Consider figure 6, where each symbol denotes an example in **S**, with symbols "x" belonging to **X** and symbols "e" belonging to **T**. The curved lines delimit the semantic vicinities (extents) of the two "e"s, i.e. sense 1 and sense 2, respectively[1]. The semantic similarity between two sentences is graphically portrayed by the physical distance between the two symbols representing them. In figure 6-a, "x"s located inside a semantic vicinity are expected to be interpreted with high certainty as being similar to the appropriate example "e," a fact which is in line with restriction 1 mentioned above. However, in figure 6-b, the degree of certainty for the interpretation of any "x" which is located inside the intersection of the two semantic vicinities cannot be great. This happens when the case fillers of two or more verb senses are not selective enough to allow a clear cut delineation among them. This situation is explicitly rejected by restriction 2.

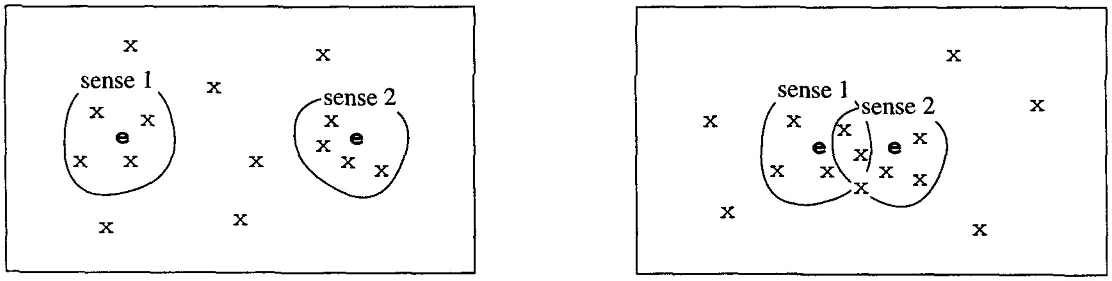

Figure 6-a: The case where the interpretation certainty of the enclosed "x" is great

Figure 6-b: The case where the interpretation certainty of the the enclosed "x" is small

Figure 6: The concept of interpretation certainty

Considering the two restrictions, we compute interpretation certainties by using equation (7), where $C(x)$ is the interpretation certainty of an example x. $S_1(x)$ and $S_2(x)$ are the highest

[1] Note that this method can easily be extended for a verb which has more than two senses. In section 4, we conducted an experiment using multiply ambiguous verbs.

and second highest scores for x, respectively. λ, which ranges from 0 to 1, is a parametric constant to control the degree to which each condition affects the computation of $C(x)$.

$$C(x) = \lambda \cdot S_1(x) + (1 - \lambda) \cdot (S_1(x) - S_2(x)) \tag{7}$$

We estimated the validity of the notion of the interpretation certainty through a preliminary experiment, in which we used the same corpus used for another experiment as described in section 4. In this experiment, we conducted a six fold-cross validation, that is, we divided the training/test data into six equal parts, and conducted six trials in which a different part was used as test data each time, and the rest as training data. We shall call these two sets the "test set" and the "training set." Thereafter, we evaluated the relation between the applicability and the precision of the system.

In this experiment, the applicability is the ratio between the number of cases where the certainty of the system's interpretation of the outputs is above a certain threshold, and the number of inputs. The precision is the ratio between the number of correct outputs, and the number of inputs. Increasing the value of the threshold, the precision also increases (at least theoretically), while the applicability decreases. Figure 7 shows the result of the experiment with several values of λ, in which the optimal λ value seems to be in the range 0.25 to 0.5. It can be seen that, as we assumed, both restrictions are essential for the estimation of the interpretation certainty.

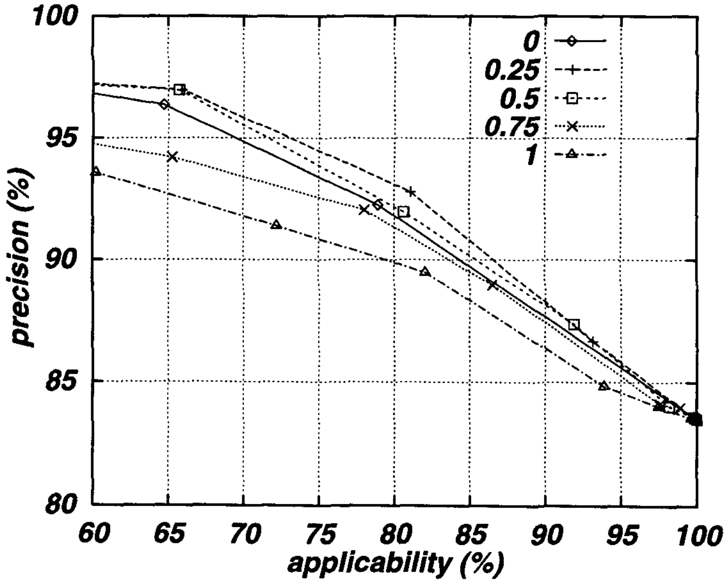

Figure 7: The relation between applicability and precision with several λ's

3.3 Training utility

The training utility of an example "a" is greater than that of another example "b" when the total interpretation certainty of examples in **X** increases more after training using the example "a" than after using the example "b." Let us consider figure 8, with the basic notation as in figure 6, and let us compare the training utility of the examples "a," "b" and "c." Note that in this figure, whatever example we use for training, the interpretation certainty for the neighbours ("x"s) of the chosen example increases. However, it is obvious that we can increase the total interpretation certainty of "x"s when we use "a" for training as it has more neighbours than either "b" or "c." In consequence, one can expect that the size of the database, which is directly proportional to the number of training examples, can be decreased. Let $\Delta C(x=s,y)$ be the difference in the interpretation certainty of $y \in \mathbf{X}$ after training with $x \in \mathbf{X}$ taken with the sense s. $TUF(x=s)$, which is the training utility function for x taken with sense s, can be computed by equation (8).

$$TUF(x=s) = \sum_{y \in \mathbf{X}} \Delta C(x=s,y) \tag{8}$$

We compute $TUF(x)$ by calculating the average of each $TUF(x=s)$, weighted by the probability that x takes sense s. This can be realized by equation (9), where $P(x=s)$ is the probability that x is used in training with the sense s.

$$TUF(x) = \sum_{s} P(x=s) \cdot TUF(x=s) \tag{9}$$

Given the fact that (a) $P(x=s)$ is difficult to estimate in the current formulation, and (b) the cost of computation for each $TUF(x=s)$ is not trivial, we temporarily approximate $TUF(x)$ as in equation (10), where $\mathbf{K}$ is a set of the k-best verb sense(s) of x with respect to the interpretation score in the current state.

$$TUF(x) \simeq \sum_{s \in \mathbf{K}} \frac{1}{k} \cdot TUF(x=s) \tag{10}$$

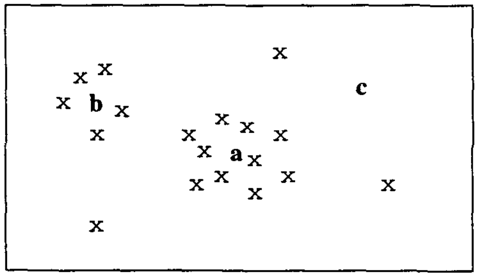

Figure 8: The concept of training utility

4 Evaluation

We compared the performance of our example sampling method with random sampling, in which a certain proportion of a given corpus is randomly selected for training. We compared the two sampling methods by evaluating the relation between various numbers of examples in training, and the performance of the system on another corpus. We conducted a six fold-cross validation as described in section 3.2, but in this experiment, each method selected some proportion of the training set as samples. We used the same corpus as described in table 2 as training/test data. Both sampling methods used examples from IPAL to initialize the system (as seeds) with the number of example case fillers for each case being on average of about 3.7.

The training/test data used in the experiment contained about one thousand simple Japanese sentences collected from news articles. Each of the sentences in the training/test data used in our experiment contained one or several complement(s) followed by one of the ten verbs enumerated in table 2. In table 2, the column of "English gloss" describes typical English translations of the Japanese verbs. The column of "# of sentences" denotes the number of sentences in the corpus, "# of senses" denotes the number of verb senses based on IPAL, and "lower bound" denotes the precision gained by using a naive method, where the system systematically chooses the most frequently appearing interpretation in the training data [6].

Table 2: The corpus used for the experiments

verb	English gloss	# of sentences	# of senses	lower bound
ataeru	give	136	4	66.9
kakeru	hang	160	29	25.6
kuwaeru	add	167	5	53.9
noru	ride	126	10	45.2
osameru	govern	108	8	25.0
tsukuru	make	126	15	19.8
toru	take	84	29	26.2
umu	bear offspring	90	2	81.1
wakaru	understand	60	5	48.3
yameru	stop	54	2	59.3
total	—	1111	—	43.7

We at first estimated the system's performance by its precision, that is the ratio of the number of correct outputs, compared to the number of inputs. In this experiment, we set $\lambda = 0.5$ in equation (7), and $k = 1$ in equation (10). The influence of CCD, i.e. α in equation (4), was extremely large so that the system virtually relied solely on the SIM of the case with the greatest CCD.

Figure 9 shows the relation between the size of the training data and the precision of the system. In figure 9, when the x-axis is zero, the system has used only the seeds given by IPAL. It should be noted that with the final step, where all examples in the training set have been provided to the database, the precision of both methods is equal. Looking at figure 9 one can see that the precision of random sampling was surpassed by our training utility sampling method. It solves the first two problems mentioned in section 1. One can also see that the size of the database can be reduced without degrading the system's precision, and as such it can solve the third problem mentioned in section 1.

We further evaluated the system's performance in the following way. Integrated with other NLP systems, the task of our verb sense disambiguation system is not only to output the most plausible verb sense, but also the interpretation certainty of its output, so that other systems can vary the degree of reliance on our system's output. The following are properties which are required for our system:

- the system should output as many correct answers as possible,
- the system should output correct answers with great interpretation certainty,
- the system should output incorrect answers with diminished interpretation certainty.

Motivated by these properties, we formulated a new performance estimation measure, PM, as shown in equation (11). A greater accuracy of performance of the system will lead to a greater

PM value.

$$PM = \frac{1}{N}\sum_{x}\delta \cdot \frac{C(x)}{C_{max}} \tag{11}$$

In equation (11), C_{max} is the maximum value of the interpretation certainty, which can be derived by substituting the maximum and the mimimum interpretation score for $S_1(x)$ and $S_2(x)$, respectively, in equation (7). Following table 1, we assign 11 and 0 to be the maximum and the minimum of the interpretation score, and therefore $C_{max} = 11$, disregarding the value of λ in equation (7). N is the total number of the inputs and δ is a coefficient defined as in equation (12).

$$\delta = \begin{cases} 1 & \text{if the interpratation of } x \text{ is correct} \\ -p & \text{otherwise} \end{cases} \tag{12}$$

In equation (12), p is the parametric constant to control the degree of the penalty for a system error. For our experiment, we set $p = 1$, meaning that PM was in the range -1 to 1.

Figure 10 shows the relation between the size of the training data and the value of PM. In this experiment, it can be seen that the performance of random sampling was again surpassed by our training utility sampling method, and the size of the database can be reduced without degrading the system's performance.

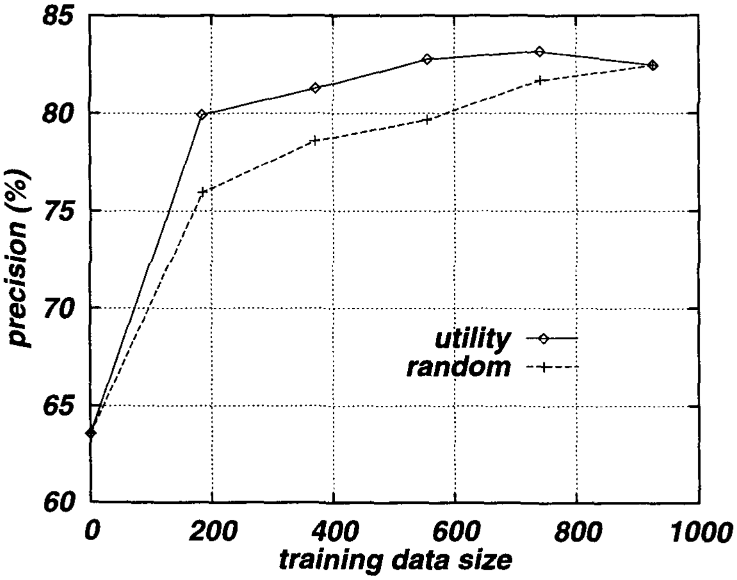

Figure 9: The relation between the training data size and precision of the system

5 Discussion

In this section, we will discuss several remaining problems. First, since in equation (8), the system calculates the similarity between x and each example in $\mathbf{X}$, computation of $TUF(x=s)$ becomes time consuming. To avoid this problem, a method used in efficient database search techniques [9, 22], in which the system can search some neighbour examples of x with optimal time complexity, can be potentially used.

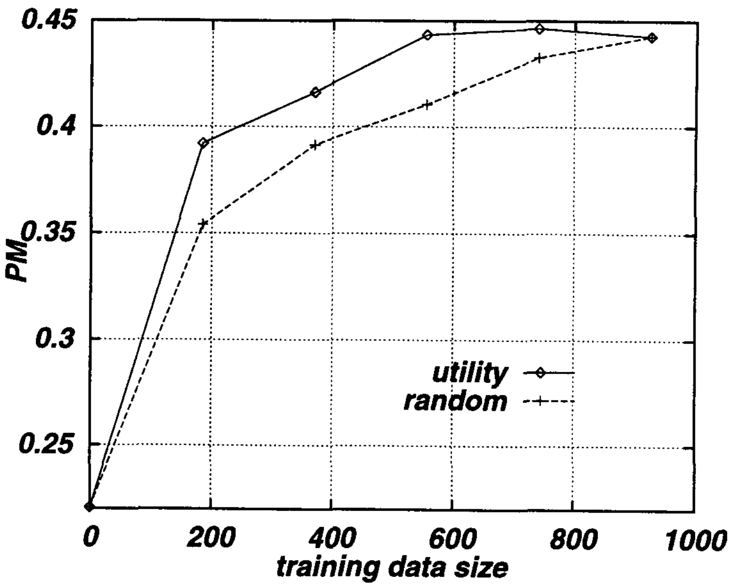

Figure 10: The relation between the training data size and performance of the system

Second, there is a problem as to when to stop the training: that is, as mentioned in section 1, it is not reasonable to manually analyze large corpora as they can provide virtually infinite input. One plausibile solution would be to select a point when the increment of the total interpretation certainty of remaining examples in **X** is not expected to exceed a certain threshold.

Finally, we should also take the semantic ambiguity of case fillers (noun) into account. Let us consider figure 11, where the basic notation is the same as in figure 6, and one possible problem caused by case filler ambiguity is illustrated. Let "x1" and "x2" denote different senses of a case filler "x." Following the basis of equation (7), the interpretation certainty of "x" is small in both figure 11-a and 11-b. However, in the situation as in figure 11-b, since (a) the task of distinction between the *verb* senses 1 and 2 is easier, and (b) instances where the sense ambiguity of case fillers corresponds to distinct verb senses will be rare, training using either "x1" or "x2" will be less effective than as in figure 11-a. It should also be noted that since *Bunruigoihyo* is a relatively small-sized thesaurus and does not enumerate many word senses, this problem is not critical in our case. However, given other existing thesauri like the EDR electronic dictionary [4] or WordNet [15], these two situations should be strictly differentiated.

6 Conclusion

In this paper we proposed an example sampling method for example-based verb sense disambiguation. We also reported on the system's performance by way of experiments. The experiments showed that our method, which is based on the notion of training utility, has reduced the overhead for the training of the system, as well as the size of the database.

As pointed out in section 1, the generalization of examples [8, 19] is another method for reducing the size of the database. Whether coupling these two methods would increase overall effectivity is an empirical matter requiring further exploration.

Future work will include more sophisticated methods for verb sense disambiguation and methods of acquiring seeds, the acquisition of which is currently based on an existing dictionary.

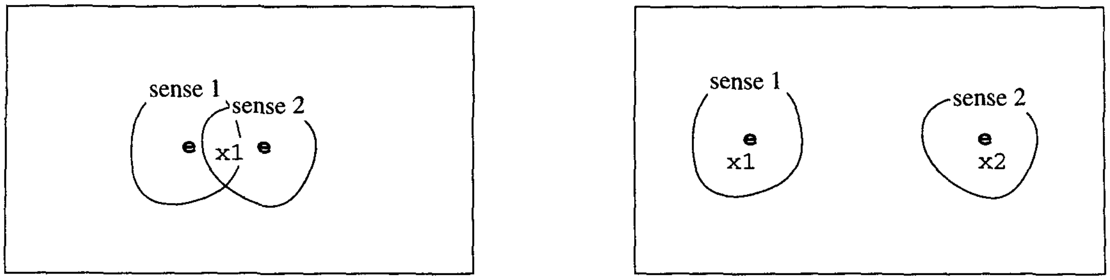

Figure 11-a: Interpretation certainty of "x" is small because "x" lies in the intersection of distinct verb senses

Figure 11-b: Interpretation certainty of "x" is small because "x" is semantically ambiguous

Figure 11: Two separate scenaries where the interpretation certainty of "x" is small

We will also build an experimental database for natural language processing using our example sampling method.

Acknowledgments

The authors would like to thank Dr. Manabu Okumura (JAIST, Japan), Mr. Timothy Baldwin (TITech, Japan), Michael Zock and Dan Tufis (LIMSI, France) for their comments on an earlier version of this paper.

References

[1] Peter F. Brown, Stephen A. Della Pietra, and Vincent J. Della Pietra. Word-Sense Disambiguation Using Statistical Methods. In *Proc. of ACL*, pp. 264–270, 1991.

[2] Ido Dagan and Sean P. Engelson. Selective Sampling in Natural Language Learning. In *IJCAI-95 Workshop on New Approaches to Learning Netural Language Processing*, pp. 41–48, 1995.

[3] Ido Dagan and Alon Itai. Word Sense Disambiguation Using a Second Language Monolingual Corpus. *Computational Linguistics*, Vol. 20, No. 4, pp. 563–596, 1994.

[4] EDR. *EDR Electronic Dictionary Specifications Guide*, 1993. (In Japanese).

[5] Atsushi Fujii, Kentaro Inui, Takenobu Tokunaga, and Hozumi Tanaka. To What Extent Does Case Contribute to Verb Sense Disambiguation? In *Proc. of COLING*, 1996. (To appear).

[6] William Gale, Kenneth Ward Church, and David Yarowsky. Estimating Upper and Lower Bounds on the Performance of Word-Sense Disambiguation Programs. In *Proc. of ACL*, pp. 249–256, 1992.

[7] IPA. *IPA Lexicon of the Japanese Language for computers IPAL (Basic Verbs)*, 1987. (In Japanese).

[8] Hiroyuki Kaji, Yuuko Kida, and Yasutsugu Morimoto. Learning Translation Templates from Bilingual Text. In *Proc. of COLING*, pp. 672–678, 1992.

[9] Janet Kolodner. *CASE-BASED REASONING.* Morgan Kaufmann, 1993.

[10] Robert Krovets and W. Bruce Croft. Lexical Ambiguity and Information Retrieval. *ACM Transactions on Information Systems*, Vol. 10, No. 2, pp. 115–141, 1992.

[11] Ikuo Kudo and Naomi Inoue. Co-Occurrence Knowledge Acquisition from Corpora and Its Application. *Journal of Japanese Society for Artificial Intelligence*, Vol. 10, No. 2, pp. 205–212, 1995. (In Japanese).

[12] Sadao Kurohashi and Makoto Nagao. A Method of Case Structure Analysis for Japanese Sentences Based on Examples in Case Frame Dictionary. *IEICE TRANSACTIONS on Information and Systems*, Vol. E77-D, No. 2, pp. 227–239, 1994.

[13] David D. Lewis and William A. Gale. A Sequential Algorithm for Training Text Classifiers. In *Proc. of SIGIR*, pp. 3–12, 1994.

[14] Steven L. Lytinen. Dynamically Combining Syntax and Semantics in Natural Language Processing. In *Proc. of AAAI*, pp. 574–578, 1986.

[15] George A. Miller, et al. Five Papers on WordNet. Technical report, Cognitive Science Laboratory, Princeton University, 1993.

[16] Katashi Nagao. A Preferential Constraint Satisfaction Technique for Natural Language Analysis. *IEICE TRANSACTIONS on Information and Systems*, Vol. E77-D, No. 2, pp. 161–170, 1994.

[17] National-Language Research Institute, editor. *Bunruigoihyo.* Syuei publisher, 1964. (In Japanese).

[18] Yoshiki Niwa and Yoshihiko Nitta. Co-occurrence vectors from corpora vs. distance vectors from dictionaries. In *Proc. of COLING*, pp. 304–309, 1994.

[19] Hiroshi Nomiyama. Machine Translation by Case Generalization. *Information Processing Society of Japan*, Vol. 34, No. 5, pp. 905–912, 1993. (In Japanese).

[20] Hinrich Schütze. Word sense disambiguation with sublexical representations. In *Workshop Notes, Statistically-Based NLP Techniques, AAAI*, pp. 109–113, 1992.

[21] Naohiko Uramoto. Example-Based Word-Sense Disambiguation. *IEICE TRANSACTIONS on Information and Systems*, Vol. E77-D, No. 2, pp. 240–246, 1994.

[22] Takehito Utsuro. Efficient Retrieval of Similar Examples based-on Similarity Templates. *Information Processing Society of Japan SIG Notes*, Vol. 94, No. 103, pp. 33–40, 1994. (In Japanese).

[23] Ellen M. Voorhees. Using WordNet to Disambiguate Word Senses for Text Retrieval. In *Proc. of SIGIR*, pp. 171–180, 1993.

[24] David Yarowsky. Unsupervised Word Sense Disambiguation Rivaling Supervised Methods. In *Proc. of ACL*, pp. 189–196, 1995.