IRAL '97

Design and Evaluation of Semantic-Based Sentence Categorization

Atsushi Fujii, Kentaro Inui, Takenobu Tokunaga and Hozumi Tanaka

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 describes a method for Japanese sentence categorization based on word usage with different meanings, and demonstrates its effectivity by way of experiments. Unlike conventional character-based or word-based categorization, our categorization scheme is characterized by its reliance on *semantic* similarity between an input and supervised examples in a database. For the similarity computation, we tentatively compared two methods: a thesaurus-based method and statistical method. Our framework also provides a means of minimizing the overhead required for constructing and searching a database for a system of practical size. Our experiments showed that our method reduced the overhead, without degenerating the performance of categorization.

1 Introduction

This paper describes a method for sentence categorization based on word usage with different meanings (senses), and demonstrates its effectivity by way of experiments. Our current research focuses on Japanese verbs, such as the following input sentence containing the polysemous verb *toru*:

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

In Japanese, a complement of a verb consists of a noun phrase (case filler) and its case marker suffix, for example *ga* (nominative), *ni* (dative) or *wo* (accusative). The verb *toru* has multiple senses, some of which are "to take/steal", "to attain" and "to reserve". As with most categorization methods, we rely on the use of a database, that is, a set of supervised examples (manually disambiguated and listed with verb sense). The input is then categorized based on the scored similarity between examples in the database. Here, suppose the following examples related to the senses "to attain" and "to reserve" for *toru*:

gakusei ga	*biza wo*	*toru,*
(student-NOM)	(visa-ACC)	("to attain")
joshu ga	*hikouki wo*	*toru.*
(assistant-NOM)	(airplane-ACC)	("to reserve")

One may notice that the verb in the original input can easily be categorized as the sense "to reserve" based on these examples, given that the case fillers *hisho* ("secretary") and *shindaisha* ("sleeping car") are semantically similar to *joshu* ("assistant") and *hikouki* ("airplane") respectively, and both collocate with the "to reserve" sense of *toru*. It should be noted that this is different from conventional character-based or word-based categorization in respect of the fact that our framework is susceptible to ***semantic relatedness*** between an input and supervised examples. It should also be noted that this framework is equivalent to the process termed "word sense disambiguation" (WSD), which is one of the crucial tasks of natural language processing (NLP).

Our preliminary experiments showed that the accuracy of categorization is roughly 60% when using only limited examples taken from a machine readable dictionary, providing one or more examples for each word sense entry. To achieve higher accuracy, we simply obtain a text ("corpus") and enhance the database manually by annotating sentences in the corpus with correct verb senses. On the other hand, the following problems should also be taken into account:

- given human resource limitations, it is not reasonable to supervise every example in large corpora ("overhead for supervision"),
- given the fact that example-based systems, including our system, search the database for the most similar examples with regard to the input, the computational cost becomes prohibitive if one works with a very large database size ("overhead for search").

To counter these problems, our system extracts a small number of optimally informative examples ("samples") from given corpora, for human supervision. Consequently we can expect to reduce both overheads without degrading the system's performance. In our previous work, we demonstrated the effectivity of this method by comparison with a random sampling method [6]. In this paper, we describe a more extensive evaluation, in which we compare our sampling method with an existing uncertainty sampling method [13].

Section 2 elaborates on our categorization engine, and describes an empirical evaluation of it, in which our system achieved an accuracy of more than 80%. Section 3 then introduces the notion of "certainty" to achieve higher accuracy. Finally, section 4 elaborates on how to minimize overhead, without degrading the system's accuracy.

2 Categorization engine

Our system, which was proposed by Kurohashi et al. [12] and enhanced by Fujii et al. [7], uses a database containing examples of collocations for each verb sense and their associated case frame(s). Figure 1 shows a fragment of the database associated with the Japanese verb *toru*. The database specifies the case frame(s) associated with each verb sense. The database lists several case filler examples for each case. Given an input, the system identifies the verb sense on the basis of the similarity between the input and examples for each verb sense contained in the database. In this process, we use nearest neighbor resolution, that is, the verb in the input is interpreted by superimposing the sense of the verb appearing in the example of highest similarity. To formalize this notion, the system computes the plausibility score for each verb sense candidate, and chooses the sense that maximizes this score. The score is computed by considering the weighted average of the similarity of the input case fillers with respect to each of the corresponding example case fillers listed in the database for the sense under evaluation. Formally, this is expressed by equation (1), where $Score(s)$ is the score for verb sense s. n_C denotes the case filler for case c, and $\mathcal{E}_{s,c}$ denotes a set of case filler examples for each case c of sense s (for example, $\mathcal{E} = \{kare, kanojo, gakusei\}$ for the *ga* case in the "to attain" sense in figure 1). $sim(n_C, e)$ stands for the similarity between n_C and an example case filler e.

$$Score(s) = \frac{\sum_C CCD(c) \cdot \max_{e \in \mathcal{E}_{s,c}} sim(n_C, e)}{\sum_C CCD(c)} \quad (1)$$

$CCD(c)$ expresses the weight factor of the contribution of case c to (current) verb sense disambiguation. Intuitively, preference should be given to cases displaying case fillers which are classified in semantic categories of greater disjunction. In this way, 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 fewer elements. This can be realized by equation (2).

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

Here, 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. Our preliminary experiment showed that the stronger influence we allow CCD to have, the better performance we gain.

With regard to the computation of the similarity between two different case fillers ($sim(n_C, e)$ in equation (1)), there are two alternative approaches. The first approach uses semantic resources, that is, hand-crafted thesauri (such as the Roget's thesaurus [1] or WordNet [16] in the case of English, and *Bunruigoihyo* [17] or EDR [3] in the case of Japanese), based on the intuitively feasible assumption that words located near each other within the structure of a thesaurus have similar meaning. Therefore, the similarity between two given words is represented by the length of the path between them in the thesaurus structure [12, 14, 21]. We used the similarity function proposed by Kurohashi et al., in which the relation between the length of the path in the *Bunruigoihyo* thesaurus and the similarity, is defined as shown in table 1. Figure 2 shows a fragment of the *Bunruigoihyo* thesaurus including some of the nouns in both figure 1 and the input sentence above, with each word corresponding to a leaf in the structure of the thesaurus. When nouns are associated with multiple concepts, that is, they are multiply located in the *Bunruigoihyo* thesaurus, we determine the similarity between each combination of senses associated with the given nouns, and take the maximal similarity (equation (3)).

$$sim(n_1, n_2) = \max_{c_1, c_2} sim(c_1, c_2) \quad (3)$$

Here, c_1 and c_2 denote senses associated with nouns n_1 and n_2, respectively.

The second approach is based on statistical modeling [2, 9, 20]. Here, each noun n is represented by a vector comprising statistical co-occurrence fac-

{ *suri* (pickpocket), *kanojo* (she), *ani* (brother) } *ga*	{ *kane* (money), *saifu* (wallet), *otoko* (man), *uma* (horse), *aidea* (idea) } *wo*		*toru* (to take/steal)
{ *kare* (he), *kanojo* (she), *gakusei* (student) } *ga*	{ *menkyoshou* (license), *shikaku* (qualification), *biza* (visa) } *wo*		*toru* (to attain)
{ *kare* (he), *dantai* (group), *ryokoukyaku* (passenger), *joshu* (assistant) } *ga*	{ *kippu* (ticket), *heya* (room), *hikouki* (airplane) } *wo*		*toru* (to reserve)

Figure 1: A fragment of the database, and the entry associated with the Japanese verb *toru*

Table 1: The relation between the length of the path between two nouns n_1 and n_2 in the *Bunruigoihyo* thesaurus ($len(n_1, n_2)$) and their similarity ($sim(n_1, n_2)$)

$len(n_1, n_2)$	0	2	4	6	8	10	12	14
$sim(n_1, n_2)$	12	11	10	9	8	7	5	0

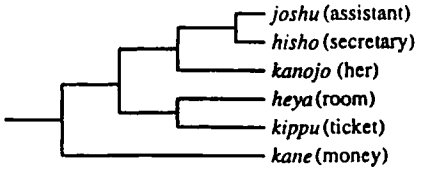

Figure 2: A fragment of the *Bunruigoihyo* thesaurus

tors, and as such, this model can be called a "vector space model" (VSM). This can be expressed by equation (4), where $\vec{n}$ is the vector for the noun in question, and t_i is the co-occurrence statistics for n and each co-occurring verb.

$$\vec{n} = < t_1, t_2, \ldots, t_i, \ldots > \quad (4)$$

Co-occurrence data was extracted from the RWC text base RWC-DB-TEXT-95-1 [18]. This text base consists of 4 years worth of Mainichi Shimbun [19] newspaper articles, which have been automatically annotated with morphological tags. The total morpheme content is about 100 million. Instead of conducting full parsing on the text, several heuristics were used in order to obtain dependencies between complements (noun + case marker) and verbs in the form of tuples $<n, c, v>$. In regard to t_i, we used the notion of TF·IDF [5], in which t_i is calculated as in equation (5), where $f(<n, c, v>)$ is the frequency of the tuple $<n, c, v>$, $nf(<c, v>)$ is the number of *noun types* which collocate with verb v in the case c, and N is the number of noun types within the overall co-occurrence data.

$$t_i = f(<n, c, v>) \cdot \log \frac{N}{f(<c, v>)} \quad (5)$$

We then compute the similarity between nouns n_C and e by the cosine of the angle between the two vectors $\vec{n_C}$ and $\vec{e}$. This is realized by equation (6).

$$sim(n_C, e) = \frac{\vec{n_C} \cdot \vec{e}}{|\vec{n_C}||\vec{e}|} \quad (6)$$

Here, let us compare the following three methods of similarity computation:

- vector space model (VSM),
- use of the *Bunruigoihyo* thesaurus (BGH),
- use of the *Bunruigoihyo* thesaurus, combined with the notion of case contribution (BGH+CCD)[1].

The training/test data used in the experiment contained about one thousand simple Japanese sentences collected from news articles. Prior to the core disambiguation process, morphological analysis, including lexical segmentation and part-of-speech tagging, is needed because Japanese sentences lack lexical segmentation. These process can be automated through the use of existing NLP tools such as JUMAN [15] (a morphological analyzer) and QJP [11] (a morphological and syntactic analyzer). However, to avoid the bias from incorrect analyses, we manually executed these processes and identified verbs and their complements. We also manually annotated each verb with its correct sense, as described in the machine readable Japanese dictionary "IPAL" [10]. Each of the sentences in the training/test data contained one or more complement(s) followed by one of the eleven verbs described 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, and "# of senses" denotes the number of verb senses contained in IPAL.

[1]The influence of CCD, i.e. α in equation (2), was extremely large, so much so that the system virtually relied solely on the similarity of the case with the greatest CCD.

Table 2: The verbs contained in the corpus used

verb	English gloss	# of sentences	# of senses
ataeru	give	136	4
kakeru	hang	160	29
kuwaeru	add	167	5
motomeru	require	204	4
noru	ride	126	10
osameru	govern	108	8
tsukuru	make	126	15
toru	take	84	29
umu	bear offspring	90	2
wakaru	understand	60	5
yameru	stop	54	2
total	—	1315	—

For each of the eleven verbs, we conducted six-fold cross validation; that is, we divided the training/test data into six equal parts, and conducted six trials in each of which a different one of the six parts was used as test data and the rest was used as training data. We shall call the former the "test set" and the latter the "training set", in each case. Figure 3 shows the results, in which the x-axis denotes the number of the data used from the training set. In figure 3, one can see that the use of the *Bunruigoihyo* thesaurus combined with the notion of CCD outperformed other methods. When the whole training set was provided, the accuracies were roughly 83% (BGH+CCD), 79% (BGH) and 78% (VSM). It is also important to estimate the lower bound of the task: the accuracy gained by using a naive method, in which the system systematically chooses the most frequently appearing interpretation in the training data [8]. We found that the lower bound was about 51%, meaning the accuracy for any of the three similarity computation methods is greatly superior to this value. In the following section, we enhance our system based on the *Bunruigoihyo* thesaurus and its reliance on the notion of CCD.

3 Certainty computation

Since, as shown in figure 3, our system still finds it difficult to achieve a 100% accuracy, it is important to select presumably correct outputs from the overall outputs (potentially sacrificing system coverage), for practical applications. To achieve this, it is useful to estimate the degree of certainty as to the interpretation, so that we can gain higher accuracy selecting only outputs with greater certainty degree.

Lewis and Gale estimate the certainty of an interpretation as the ratio between the probability of the most plausible text category, and the probability of any other text category, excluding the most probable one [13]. Similarly, Fujii et al. proposed the notion of interpretation certainty of examples based on the following preference conditions [6]:

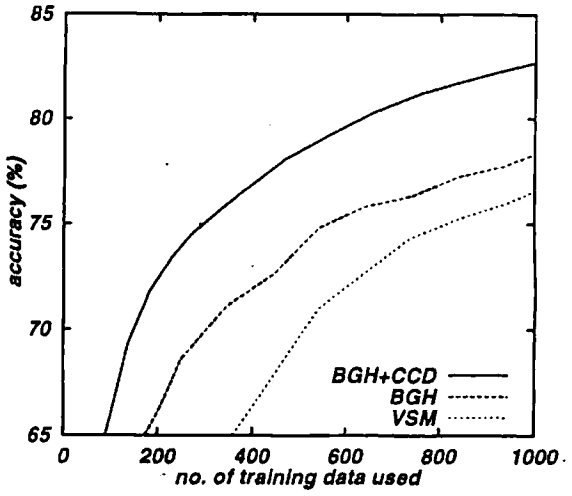

Figure 3: the accuracy of each method, for each size of training data

1. the highest interpretation score is greater,
2. the difference between the highest and second highest interpretation scores is greater.

The rationale for these conditions is given below. Consider figures 4 and 5, where each symbol denotes an example in a given corpus, with symbols x as unsupervised examples and symbols e as supervised examples. The curved lines delimit the semantic vicinities (extents) of the two verb senses 1 and 2, respectively. The semantic similarity between two examples is graphically portrayed by the physical distance between the two symbols representing them. In figure 4, x's located inside a semantic vicinity are expected to be interpreted as being similar to the example e which defines this vicinity, with high certainty. This fact is in line with condition 1 above. However, in figure 5, 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 occurs when the case fillers associated with two or more verb senses are not selective enough to allow for a clear cut delineation between them. This situation is explicitly rejected by condition 2.

Based on the above two conditions, we compute interpretation certainties using equation (7), where $C(x)$ is the interpretation certainty of an example x. $Score_1(x)$ and $Score_2(x)$ are the highest and second highest scores for x, respectively. λ, which ranges from 0 to 1, is a parametric constant used to control the degree to which each condition affects

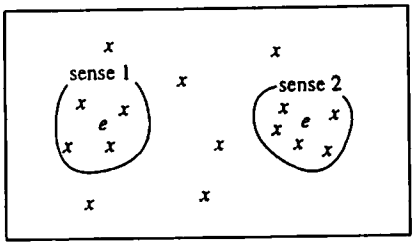

Figure 4: The case where the interpretation certainty of the enclosed x's is great

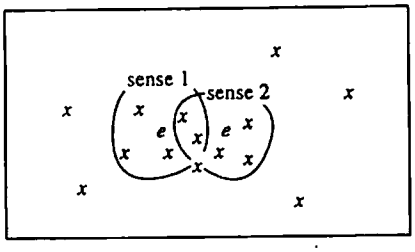

Figure 5: The case where the interpretation certainty of the x's contained in the intersection of senses 1 and 2 is small

the computation of $C(x)$.

$$C(x) = \lambda \cdot Score_1(x) + (1-\lambda) \cdot (Score_1(x) - Score_2(x)) \quad (7)$$

We estimated the validity of the notion of interpretation certainty through an experiment, in which we used the same corpus as that used for the experiment described in section 2. In this experiment, we evaluated the relation between the applicability and the accuracy of the system, given that 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. By raising the value of the threshold, the accuracy also increases (at least theoretically), while the applicability decreases. Figure 6 shows the result of the experiment with several values of λ, from which the optimal λ value seems to be in the range around 0.5. It can be seen that, as we assumed, both of the above conditions are essential for the estimation of the interpretation certainty, and for example, we could achieve an accuracy of 93% with an applicability of 60%.

4 Minimizing the overhead

4.1 Overview

Our example sampling method, based on the utility maximization principle, decides on the preference for including a given example in the database. This decision procedure is usually called *selective sampling*. The overall control flow of selective sampling systems can be depicted as in figure 7, where

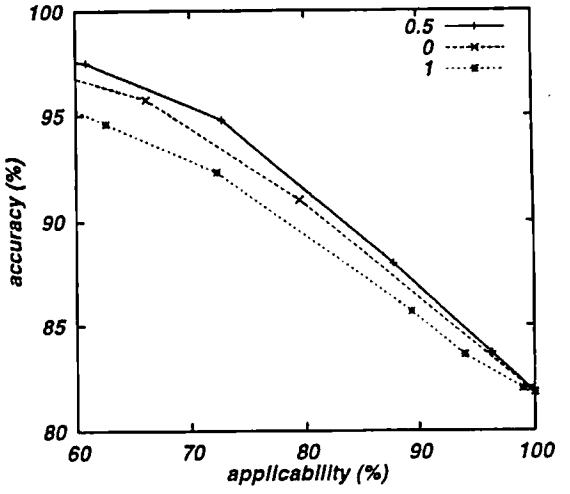

Figure 6: The relation between applicability and accuracy with different λ's

"system" refers to our verb sense disambiguation system, and "examples" refers to an unsupervised example set. The sampling process basically cycles between the word sense disambiguation (WSD) and training phases. During the WSD phase, the system generates an interpretation for each polysemous verb contained in the input example ("WSD outputs"). This phase is equivalent to semantic-based categorization as described in section 2. During the training phase, the system selects samples for training from the previously produced outputs. During this phase, a human expert supervises samples, that is, provides the correct interpretation for the verbs appearing in the samples. Thereafter, samples are contained in the database, meaning that the system can be trained on the remaining examples ("residue") for the next iteration. Iterating these two phases, the system progressively enhances the database. Note that the selective sampling procedure gives us an optimally informative database of a given size irrespective of the stage at which processing is terminated.

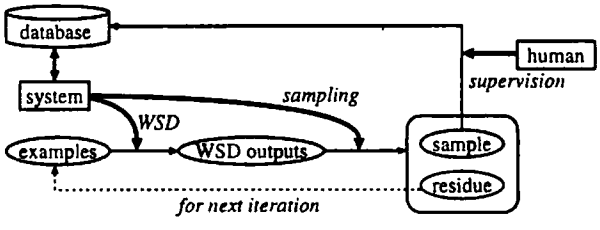

Figure 7: flow of control of the example sampling system

Lewis and Gale proposed the notion of uncertainty sampling for the training of statistics-based text classifiers [13]. Their method selects those ex-

amples that the system classifies with minimum certainty, based on the assumption that there is no need for teaching the system the correct answer when it has answered with sufficiently high certainty. However, we should take into account the training effect a given example has on other remaining (unsupervised) examples. In other words, we would like to select samples such as to be able to correctly disambiguate as many examples as possible in the next iteration. If this is successfully done, the number of examples requiring supervision will decrease. We consider maximization of this effect by means of a training utility function aimed at ensuring that the example with the greatest training utility factor, is the most useful example at a given point in time. Intuitively speaking, the training utility of an example is greater when we can expect greater increase in the interpretation certainty of the remaining examples after training using that example.

Let S be a set of sentences, i.e. a given corpus, and D be the subset of supervised examples stored in the database. Further, let X be the set of unsupervised examples, realizing equation (8).

$$\mathbf{S} = \mathbf{D} \cup \mathbf{X} \tag{8}$$

The example sampling procedure can be illustrated as:

1. $WSD(\mathbf{D}, \mathbf{X})$
2. $e \leftarrow \arg\max_{x \in \mathbf{X}} TU(x)$
3. $\mathbf{D} \leftarrow \mathbf{D} \cup \{e\}$, $\mathbf{X} \leftarrow \mathbf{X} \cap \overline{\{e\}}$
4. goto 1

where $WSD(\mathbf{D}, \mathbf{X})$ is the verb sense disambiguation process on input X using D as the database. In this disambiguation process, the system outputs the following for each input: (a) a set of verb sense candidates with interpretation scores, and (b) an interpretation certainty. These factors are used for the computation of $TU(x)$, newly introduced in our method. $TU(x)$ computes the training utility factor for an example x. The sampling algorithm gives preference to examples of maximum utility.

We will explain in the following sections how one can estimate $TU(x)$, based on the estimation of the interpretation certainty.

4.2 Training utility

The training utility of an example a is greater than that of another example b when the total interpretation certainty of unsupervised examples increases more after training with example a than with example b. Let us consider figures 8 and 9, in which the x-axis mono-dimensionally denotes the semantic similarity between two unsupervised examples, and the y-axis denotes the interpretation certainty of each example. Let us compare the training utility of examples a and b in figure 8. Note that in this figure, whichever example we use for training, the interpretation certainty for each unsupervised example (x) neighboring the chosen example increases based on its similarity to the supervised example. Since the increase in the interpretation certainty of a given x becomes smaller as the similarity to a or b diminishes, the training utility of the two examples can be represented by the shaded areas. It is obvious that the training utility of a is greater, as it has more neighbors than b. On the other hand, in figure 9, b has more neighbors than a. However, since b is semantically similar to e, which is already contained in the database, the total *increase* in interpretation certainty of its neighbors, i.e. the training utility of b, is smaller than that of a.

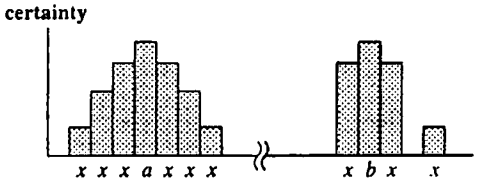

Figure 8: The case where the training utility of a is greater than that of b because a has more unsupervised neighbors

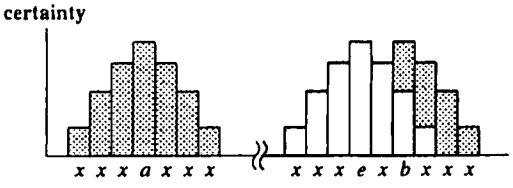

Figure 9: The case where the training utility of a is greater than that of b because b closely neighbors e, contained in the database

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. $TU(x{=}s)$, which is the training utility function for x taken with sense s, can be computed by way of equation (9)[2].

$$TU(x{=}s) = \sum_{y \in \mathbf{X}} \Delta C(x{=}s, y) \tag{9}$$

[2]It should be noted that in equation (9), we can replace X with a smaller subset of X which consists of neighbors of x: example y is a neighbor of x if they share the same supervised example e as their nearest neighbor. Since the system determines e during the WSD phase, identifying the neighbors of x does not require any extra computational overhead.

Since there is no guarantee that x will be supervised with any given sense s, it can be risky to rely solely on $TU(x=s)$ for the computation of $TU(x)$. We estimate $TU(x)$ by the expected value of x, calculating the average of each $TU(x=s)$, weighted by the probability that x takes sense s. This can be realized by equation (10), where $P(x=s)$ is the probability that x takes sense s.

$$TU(x) = \sum_s P(x=s) \cdot TU(x=s) \qquad (10)$$

Given the fact that (a) $P(x=s)$ is difficult to estimate in the current formulation, and (b) the cost of computation for each $TU(x=s)$ is not trivial, we temporarily approximate $TU(x)$ as in equation (11), where K is a set of the k-best verb sense(s) of x with respect to the interpretation score in the current state.

$$TU(x) \simeq \frac{1}{k} \sum_{s \in \mathrm{K}} TU(x=s) \qquad (11)$$

4.3 Experimentation

In this experiment, we compared the following four example sampling methods:

1. a control (random), in which a certain proportion of a given corpus is randomly selected for training,
2. uncertainty sampling (US), in which examples with minimum interpretation certainty are selected [13],
3. committee-based sampling (CBS) [4],
4. our method based on the notion of training utility (TU).

In committee-based sampling, the system selects samples based on the degree of disagreement between models randomly taken from a given set of supervised examples (these models are called "committee members"). This is achieved by the iteratively repeating the steps given below, in which the number of committees is given as two without loss of generality[3]:

1. draw two models randomly,
2. classify unsupervised example x according to each model, producing classifications C_1 and C_2,
3. if $C_1 \neq C_2$ (the committee members disagree), select x for the training of the system.

Note that the method of "random" is equivalent to the method labeled as "BGH+CCD" in the experiment in section 2. We compared these sampling methods by evaluating the relation between the number of training examples and the performance of the system. We conducted six-fold cross validation and carried out sampling on the training set. During the initial phase, each sampling method randomly selected one example ("seed") for each verb sense from the training set, and a human expert provided the correct interpretation to initialize the system. We used the same corpus as that used for the experiment described in section 2.

We evaluated each system's performance according to its accuracy, that is the ratio of the number of correct outputs, compared to the total number of inputs. For the purpose of this experiment, we set $\lambda = 0.5$ for equation (7), and $k = 1$ for equation (11). Based on a preliminary experiment, increasing the value of k either did not improve the performance over that for $k = 1$, or lowered the overall performance. Figure 10 shows the relation between the size of the training data and the accuracy of the system. In figure 10, zero on the x-axis represents the system using only the seeds. Looking at figure 10 one can see that (a) our sampling method outperformed other methods, and (b) compared with random sampling, our sampling method reduced the size of the training data required to achieve any given accuracy. For example, to achieve an accuracy of 80%, the size of the training data required for our method was roughly half of that for random sampling. Through this comparative experiment, we can conclude that our example sampling method is able to decrease the size of the training data, i.e. the overhead for both supervision and searching, without degrading the system's performance.

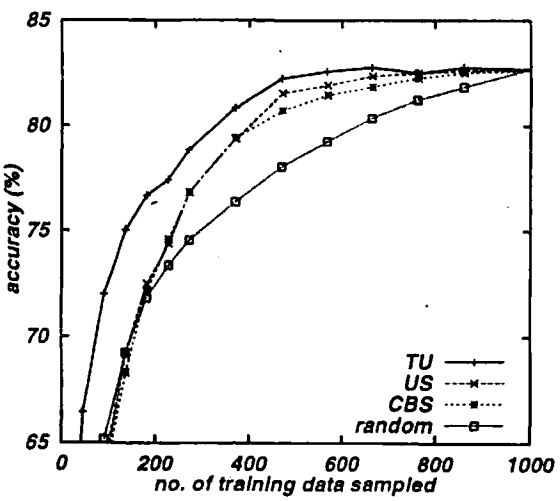

Figure 10: The relation between the number of training data sampled and accuracy of the system

[3]Engelson and Dagan applied this framework to HMM training for part-of-speech tagging [4].

5 Conclusion

Let us summarize the main points that have been made in this paper.

Evaluation of semantic-based sentence categorization Each sentence is categorized based on the semantic similarity between examples in the database. To compute semantic similarity, we tested two approaches: use of a hand-crafted thesaurus (*Bunruigoihyo*) and a statistical model. We also introduced the notion of case contribution to disambiguation (CCD) in the similarity computation. The result of our experiments showed that use of the thesaurus combined with the notion of CCD improved on the accuracy of categorization for the statistical model (from 78% to 83%).

Introduction of interpretation certainty To achieve higher accuracy, we selected presumably correct outputs from the overall categorized outputs by use of the notion of interpretation certainty. While sacrificing system coverage from 100% to 60%, the accuracy was improved from 83% to 93%.

Minimization of the overhead As with most categorization methods, our framework assumes a certain amount of supervised examples in the database, and therefore the overhead for supervision of examples and the overhead for searching the database are crucial problems. However, our sampling method is effective in selecting a smaller-sized example set for system usage, without degrading the overall performance. Compared with random sampling, our sampling method reduce the size of the database by roughly half, in achieving an accuracy of 83%.

Acknowledgments

The authors would like to thank Mr. Timothy Baldwin (TITECH, Japan) for his comments on the earlier version of this paper.

References

[1] Robert L. Chapman. *Roget's International Thesaurus (Fourth Edition)*. Harper and Row, 1984.

[2] Eugene Charniak. *Statistical Language Learning*. MIT Press, 1993.

[3] EDR. *EDR Electronic Dictionary Technical Guide*, 1995. (In Japanese).

[4] Sean P. Engelson and Ido Dagan. Minimizing manual annotation cost in supervised training from corpora. In *Proceedings of ACL*, pp. 319–326, 1996.

[5] William B. Frankes and Ricardo Baeza-Yates. *Information Retrieval: Data Structure & Algorithms*. PTR Prentice-Hall, 1992.

[6] Atsushi Fujii, Kentaro Inui, Takenobu Tokunaga, and Hozumi Tanaka. Selective sampling of effective example sentence sets for word sense disambiguation. In *Proceedings of the Fourth Workshop on Very Large Corpora*, pp. 56–69, 1996.

[7] Atsushi Fujii, Kentaro Inui, Takenobu Tokunaga, and Hozumi Tanaka. To what extent does case contribute to verb sense disambiguation? In *Proceedings of COLING*, pp. 59–64, 1996.

[8] William Gale, Kenneth Ward Church, and David Yarowsky. Estimating upper and lower bounds on the performance of word-sense disambiguation programs. In *Proceedings of ACL*, pp. 249–256, 1992.

[9] Donald Hindle. Noun classification from predicate-argument structures. In *Proceedings of ACL*, pp. 268–275, 1990.

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

[11] Masayuki Kameda. A portable & quick Japanese parser : QJP. In *Proceedings of COLING*, pp. 616–621, 1996.

[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 *Proceedings of ACM SIGIR*, pp. 3–12, 1994.

[14] Xiaobin Li, Stan Szpakowicz, and Stan Matwin. A WordNet-based algorithm for word sense disambiguation. In *Proceedings of IJCAI*, pp. 1368–1374, 1995.

[15] Yuji Matsumoto, et al. *JUMAN Users Manual*. Kyoto University and Nara Institute of Science and Technology, 1993.

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

[17] National Language Research Institute. *Bunruigoihyo*, revised and enlarged edition, 1996. (In Japanese).

[18] Real World Computing Partnership. RWC text database, 1995.

[19] Mainichi Shimbun. Mainichi shimbun CD-ROM '91-'94, 1991-1994.

[20] Takenobu Tokunaga, Makoto Iwayama, and Hozumi Tanaka. Automatic thesaurus construction based on grammatical relations. In *Proceedings of IJCAI*, pp. 1308–1313, 1995.

[21] Naohiko Uramoto. Example-based word-sense disambiguation. *IEICE TRANSACTIONS on Information and Systems*, Vol. E77-D, No. 2, pp. 240–246, 1994.