自然言語処理 113－7
音声言語情報処理 11－16
（1996. 5. 28）

効用最大化法に基づく多義性解消用事例の選択的収集

藤井敦, 乾健太郎, 徳永健伸, 田中穂積
東京工業大学大学院情報理工学研究科
{fujii,inui,take,tanaka}@cs.titech.ac.jp

本論文は、多義性解消システムに用いる事例ベースを効率的に収集する手法を提案する。従来、事例への語義付与にかかる人間の負担、及び事例ベースのサイズが問題とされてきた。本手法の特長は、人間がその事例の語義を教えてシステムを訓練することがそれ以降の事例収集に及ぼす影響、すなわち効用を定量化し、効用の最も高い事例を選択的に収集する点にある。さらに本論文は、1000 文以上の例文を含むコーパスを用いた実験について報告する。ランダムに事例を抽出する手法との比較実験の結果、本手法を用いて収集した事例ベースは、より少ない人間の負担、より小さいサイズで、より高い多義性解消の精度を実現できることが確認された。

Selective Example Sampling for Word Sense Disambiguation

FUJII Atsushi, INUI Kentaro,
TOKUNAGA Takenobu, TANAKA Hozumi
Department of Computer Science
Tokyo Institute of Technology
{fujii,inui,take,tanaka}@cs.titech.ac.jp

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 utility of training: 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 great utility. The paper reports the effectivity of our method through an experiment on over one thousand sentences. According to the comparative experiment 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 kinds of natural language processing applications, such as word selection in machine translation [18], pruning of syntactic structures in parsing [12, 13] and text retrieval [8, 21]. Research on word sense disamibiguation has variously been utilized in recent corpus-based approaches, reflecting the growth in the number of machine readable texts [17]. Unlike rule-based approaches, corpus-based approaches free us from the task of generalizing observed phenomena to produce rules for word sense disambiguation, e.g. subcategorization rules. Our verb sense disambiguation system also follows the corpus-based approach, and more precisely the example-based approach [4]. Since this approach requires a given number of examples where verbs are already disambiguated in their senses, we have to manually disambiguate verbs appearing in a corpus prior to the execution of the system. Our preliminary experiment on ten Japanese verbs showed that the system needed an average of about one hundred examples for each verb to achieve an 82% accuracy for verb sense disambiguation. With the aim to construct a practical system, we have enumerated the problems which should be taken into consideration:

1. Since there are about a thousand basic verbs in Japanese, a considerable overhead for manual sense disambiguation is required.
2. With the limitations of human overhead, we can not manually handle whole corpora which can provide virtually infinite input.
3. Since example-based natural language systems, including our verb sense disambiguation system, search for the most similar examples to the input in the example database, the cost of computation can be a crucial problem for a database of a practical working size [9].

These arguments suggest we attempt to *select* a small number of optimally informative examples from a given corpora. We will call these examples "samples" hereafter. Our method, based on the utility maximization principle, decides the order in which we manually disambiguate these samples.

To counter problems 1 and 2 above, we can apply several proposed methods from other categories of natural language research, which can be called *selective sampling*. The overall design of these methods can be illustrated as in figure 1, where "system" refers to NLP systems specialized for a certain purpose. The example sampling process basically iterates the execution and training phases. In the execution phase, the system outputs the result of the analysis, such as part-of-speech, text categories and word sense. In the training phase, the system selects samples for training out of the outputs.

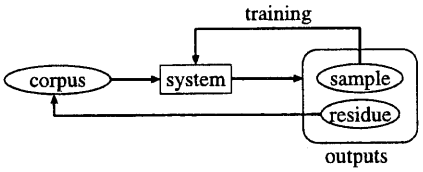

Figure 1: A conceptual diagram of the example sampling system

In this phase, a human supplies the correct resolutions to the samples, with which the system can then be trained for the subsequent execution on the residue of the data. There are a number of proposed methods following this general schema. Lewis et al. proposed an example sampling method for statistic-based text classification [11]. In this method, the system always selects samples which are uncertain with respect to the correctness of the answer. Dagan et al. proposed a commitee-based sampling method, which is currently applied to HMM training for part-of-speech tagging [2]. This method selects samples based on the utility of training by examples, i.e. to which the examples are informative for the future training. However, these methods are implemented for statistic-based models, and there is a need to explore the way to formalize these concept on example-based approaches.

In regard to problem 3, a feasible solution is the generalization of redundant examples [7, 16]. However, the degree to which examples can be generalized remains a difficult issue. Besides, this approach requires the overhead for the manual training of each example in a given corpus, prior to the generalization. Our approach is expected to reduce the overhead as well as the size of the database.

2 Example-based verb sense disambiguation system

Our proposed method for verb sense disambiguation uses an example database containing examples of collocations between each verb sense and its governing case frame(s), as in figure 2. 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 gives one or more case frame(s) associated with the verbs for each of their senses. In Japanese, a complement of a verb, which is a constituent of the case frame of the verb, consists of a noun phrase (case filler) followed by a case marker such as *ga* (nominative) or *o* (accusative). The database has a set of case filler examples for each case. The task of the system is "to interpret" a verb in an input sentence, i.e. to choose one sense

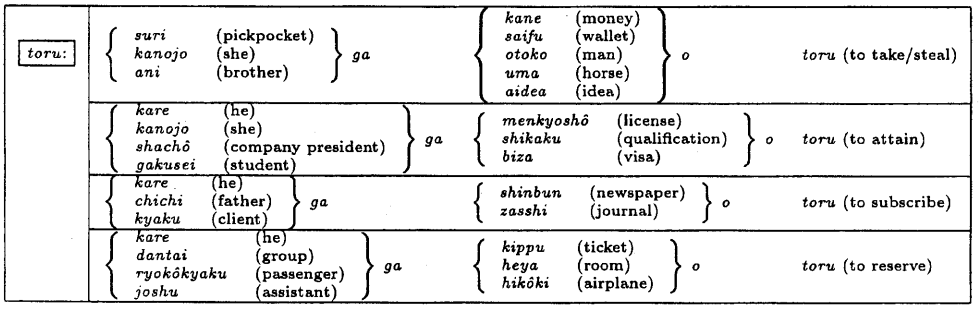

Figure 2: A fragment of an example database, and the entry associated with Japanese verb *toru*

from a set of candidate senses of the verb. The set of verb senses we use are those defined in the existing machine readable dictionary "IPAL" [6]. IPAL also contains example case fillers, exemplified by those shown in figure 2. Given an input sentence, which we assume to be a simple sentence, the system interprets the verb in the input based on the score of each verb sense, i.e. semantic similarity between the input and examples of each verb sense. Let us take the example sentence below:

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

In this example, it may be judged that *hisho* ("secretary") and *shindaisha* ("sleeping car") in (1) are semantically similar to *joshu* ("assistant") and *hikôki* ("airplane"), respectively, which are examples that collocate with *toru* ("to reserve"). As such, the sense of *toru* in (1) can be interpreted as "to reserve." The similarity between case frames is estimated according to the length of the path between two case frames in a thesaurus. We experimentally use the Japanese word thesaurus *Bunruigoihyo* [14]. As with most thesauruses, the length of the path between two words in *Bunruigoihyo* is expected to reflect the similarity between them. Kurohashi et al. proposed the similarity between two words by using the length of path between them as shown in table 1 [10].

Furthermore, we take the degree of contribution of each case to verb sense disambiguation (CCD) into consideration for the computation of the score of verb sense. In the case of *toru*, since the semantic range of nouns collocating with the verb in the nominative do 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 in the accusative. Note that this difference would be critical if example data were sparse (for more detail, see our paper [4]).

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

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

To illustrate the overall algorithm, we replace the illustrative example sentence mentioned above with a slightly more general case as in figure 3. The input is $\{n_{c_1}\text{-}m_{c_1}, n_{c_2}\text{-}m_{c_2}, v\}$, where n_{c_i} denotes the case filler in the case c_i, and m_{c_i} denotes the case maker of c_i. The candidates of interpretation for v are derived from the database as s_1, s_2 and s_3. The database also gives a set $\mathcal{E}_{s_i,c_j}$ of case filler examples for each case c_j of each sense s_i. "—" denotes that the corresponding case is not allowed.

input	n_{c_1}-m_{c_1}	n_{c_2}-m_{c_2}		v (?)
database	$\mathcal{E}_{s_1,c_1}$	$\mathcal{E}_{s_1,c_2}$	—	v (s_1)
	$\mathcal{E}_{s_2,c_1}$	$\mathcal{E}_{s_2,c_2}$	$\mathcal{E}_{s_2,c_3}$	v (s_2)
	—	$\mathcal{E}_{s_3,c_2}$	—	v (s_3)

Figure 3: An input and the database

In the course of the verb sense disambiguation process, the system first discards the candidates whose case frame constraint is grammatically violated by the input In the case of figure 3, s_3 is discarded because the case frame of v (s_3) does not subcategorize the case c_1. In contrast, s_2 will not be rejected at this step. This is based on the fact that in Japanese, cases can be easily omitted if they are inferable from the given context.

Thereafter, the system computes the score of the remaining candidates of interpretation and chooses the most plausible interpretation assigned the highest score as its output. We compute the score of an interpretation by the *weighted* average of the degree of similarity between the input complement and the example complements [1] for each case as in equation (1), where $S(s)$ is the score of interpretation of the input verb as sense s, and $SIM(n_C, \mathcal{E}_{s,C})$ is the degree of the similarity between the input complement

[1] $\mathcal{E}_{s_2,c_3}$ is not taken into consideration in the computation since c_3 does not appear in the input.

n_C and example complements $\mathcal{E}_{s,c}$.

$$S(s) = \frac{\sum_C SIM(n_C, \mathcal{E}_{s,c}) \cdot CCD(c)}{\sum_C CCD(c)} \quad (1)$$

$SIM(n_C, \mathcal{E}_{s,c})$ is the maximum degree of similarity between n_C and each of $\mathcal{E}_{s,c}$ as in equation (2), where *sim* is the similarity between n_C and e given by table 1.

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

Here, $CCD(c)$ is the degree of contribution of the case to verb sense disambiguation. $CCD(c)$ is greater when the degree of c's contribution is higher. The degree of contribution of case to verb sense disambiguation (CCD) is computed in the following way. The degree of contribution of a case should be high if the semantic range of the example case fillers in that case is diverse in the case frame. Let a certain verb have n senses $(s_1, s_2, \ldots, s_n)$ and the set of example case fillers of a case c associated with s_i be $\mathcal{E}_{s_i,c}$. Then, the degree of c's contribution to disambiguation, $CCD(c)$, is expected to be higher if the example case filler sets $\{\mathcal{E}_{s_i,c} \mid i = 1, \ldots, n\}$ share less elements. This can be realized by equation (3).

$$CCD(c) = \left(\frac{1}{{}_nC_2} \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \frac{|\mathcal{E}_{s_i,c}| + |\mathcal{E}_{s_j,c}| - 2|\mathcal{E}_{s_i,c} \cap \mathcal{E}_{s_j,c}|}{|\mathcal{E}_{s_i,c}| + |\mathcal{E}_{s_j,c}|} \right)^{\alpha} \quad (3)$$

α is the constant for parameterizing to what extent CCD influences verb sense disambiguation. When α is larger, CCD more strongly influences the system's output. Considering the data sparseness problem, we do not distinguish two nouns X and Y in equation (3) if X and Y are similar enough, as in equation (4).

$$\{X\} + \{Y\} = \{X\} \text{ if } sim(X, Y) >= 9 \quad (4)$$

3 Example sampling algorithm

3.1 Overview

Let us consider figure 1 in section 1 again. In this figure, "outputs" refers to a corpus in which each sentence is assigned the interpretation of the verb. In "training," the system stores samples of outputs manually disambiguated in verb senses (verified or corrected in their interpretations) in the database. The task of the system focused on in this paper is to select some samples for training, out of the outputs.

Lewis et al. proposed the notion of uncertain example sampling for the training of statistic-based text classifiers [11]. The method selects such examples that the system answers (in this case, the answer is a text category) with least certainty. This method is executed based on the assumption that we do not have to teach the correct answer when the system answers with great certainty. However, we should take the training effect of a given example on other examples into consideration. In other words, by selecting an appropriate example as a sample, we can get more certain examples in the next iteration. Consequently, the number of examples which we have to teach will be decreased. We introduce this effect as the notion of the utility of training. The system always selects the example which has greatest utility at that point.

Let $\boldsymbol{S}$ be a set of sentences, i.e. a given corpus, and $\boldsymbol{T}$ be a subset of $\boldsymbol{S}$ in which each sentence has already manually been disambiguated for training. In other words, $\boldsymbol{T}$ has beed selected as sample, and thus they are stored in the database. Let $\boldsymbol{X}$ be the set of the residue, realizing equation (5).

$$\boldsymbol{S} = \boldsymbol{X} + \boldsymbol{T} \quad (5)$$

We introduce the utility function $U(x)$, which represents the degree of the utility of training by x. The system selects the example which satisfies equation (6) as a sample.

$$\arg\max_{x \in \boldsymbol{X}} U(x) \quad (6)$$

We will explain the way to estimate the certainty of interpretation and the utility of training in the following sections.

The sampling size, i.e. the number of samples selected, in each iteration would ideally be such as to avoid our having to retrain similar examples. It should be noted that this can be a critical problem for statistic-based approaches [1, 3, 15, 19, 22], in which the cost of the reconstruction of statistic classifiers is expensive. However, example-based systems [4, 10, 20] does not require the reconstruction of the system, except for storing examples in the database.

3.2 Certainty of interpretation

Lewis et al. estimates the certainty by the ratio of the probability of the most plausible text category, to that of the rest of candidates of text categories. Similarly, in our example-based verb sense disambiguation system, we introduce the notion of certainty of interpretation of examples based on the following conditions:

1. the highest score of interpretation is sufficiently large,
2. the highest score of interpretation is sufficiently larger than second highest score.

This argument can be illustrated as shown in figure 4, where each symbol denotes an example in $\boldsymbol{S}$, and symbols "x" belong to $\boldsymbol{X}$. The symbols "e" belong to $\boldsymbol{T}$ and curved lines represent the semantic

ranges of the "e"s, i.e. sense 1 and sense 2, respectively [2]. The semantic similarity between two usages is represented by the physical distance between the two symbols. In figure 4-a, "x"s located inside the semantic range are expected to be interpreted as neighbor verb senses with great certainty, which supports condition 1 mentioned above. However, in figure 4-b, since sense 1 and sense 2 do not have a strong delinearization in semantic range, the certainty of interpretation of "x" being located inside either semantic range can not be great. This argument supports condition 2.

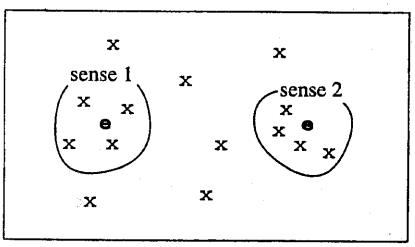

Figure 4-a: The case where the certainty of interpretation of the enclosed x is great

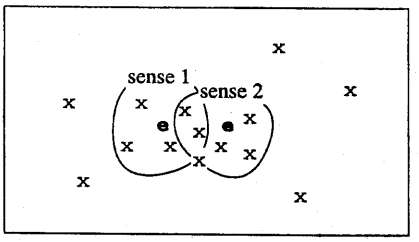

Figure 4-b: The case where the certainty of the interpretation of the enclosed x is small

Figure 4: The concept of certainty of interpreration

Considering the two conditions, we compute certainty of interpretation by equation (7), where $C(x)$ is the certainty of interpretation of an example x. $S_1(x)$ and $S_2(x)$ are the highest and second highest score of x, respectively. w, which ranges from 0 to 1, is a parametric constant to control the degree to which each condition affects the computation of $C(x)$.

$$C(x) = w \cdot S_1(x) + (1 - w) \cdot (S_1(x) - S_2(x)) \quad (7)$$

We estimated the validity of the notion of the certainty of interpretation through a preliminary experiment, in which we used the same corpus used for another experiment in section 4. The basic trial performed was that the system interpreted each example and computed its certainty by equation (7). The rest of the examples in the given corpus were then used as the database. Thereafter, we evaluated the relation between the applicability and the precision of the system. In this experiment, the applicability is the ratio of the number of the cases where the system outputs interpretations with a certainty over a certain threshold, to the number of inputs. The precision is the ratio of the number of correct outputs, to the number of inputs. Increasing the value of threshold, the precision becomes theoretically greater while the applicability becomes smaller. Figure 5 shows the result of the experiment with several values of w, in which with a greater w, the precision is increasing as the applicability decreases. This indicates that equation (7) with great w is expect to reflect the actual certainty of interpretation.

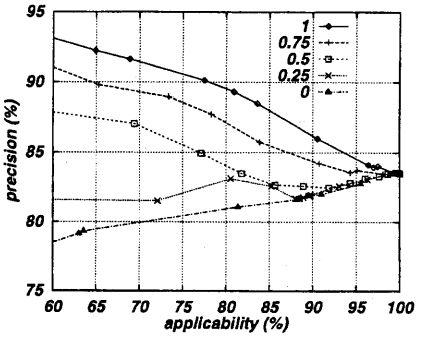

Figure 5: The relation between applicability and precision with several w's

3.3 Utility of training

The utility of training with an example is greater when the total certainty of interpretation of examples in $\boldsymbol{X}$ increases more after training using the example. Let us consider figure 6, where the basic notation is the same as in figure 4, and compare the utility of training by the examples "a," "b" and "c." Note that in this figure, whatever example we use for training, the certainty of interpretation of neighbors ("x"s) of the example increase. However, it is obvious that we can increase the total certainty of interpretation of "x"s when we use "a," which has more neighbors than "b" and "c," for training. Conseqently, it is expected that the size of the database, which is equal to the number of examples for training, can be decreased. Let $\Delta C(x = s, y)$ be the difference of the certainty of interpretation of $y \in \boldsymbol{X}$ after training with $x \in \boldsymbol{X}$ as verb sense s. $U(x = s)$, which is the utility of training with x as verb sense s, can be computed by equation (8).

$$U(x = s) = \sum_{y \in \boldsymbol{X}} \Delta C(x = s, y) \quad (8)$$

In regard to $U(x)$, we compute it by calculating the average of each $U(x = s)$, weighted by the proba-

[2]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.

bility that x is trained as the verb sense s. This can be realized by equation (9), where $P(x = s)$ is the probability that x is interpreted as s.

$$U(x) = \sum_s P(x = s) \cdot U(x = s) \qquad (9)$$

Here, since (a) $P(x = s)$ is difficult to estimate in the current formulation, and (b) the cost of computation for each $U(x = s)$ is not trivial, we temporarily assume that the greater score of a verb sense s leads to a greater $P(x = s)$. Therefore, we approximate $U(x)$ as in equation (9), where $s*$ is the verb sense which has the highest score for x.

$$U(x) \simeq U(x = s*) \qquad (10)$$

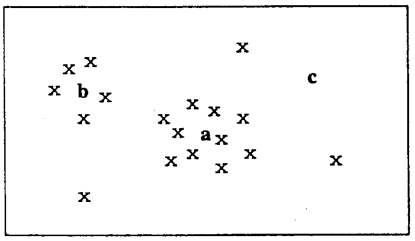

Figure 6: The concept of utility of training

4 Evaluation

We compared the performance of our example sampling method with random sampling, which always randomly selects a certain proportion of a given corpus 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. In this experiment, 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 some proportion of the rest was used as training data. We shall call the former the "test set" and the latter the "training set," in each case. Both sampling methods used examples given by IPAL as the initial database, i.e. seeds, the number of which is, on average, about 3.7 for each case. When more than one interpretation of an input verb is assigned the highest plausibility score, the system will choose as its output the one that appears most frequently in the training set. Therefore, the applicability of the system is 100%, given that the applicability is the ratio of the number of the cases where the system gives only one interpretation, to the number of inputs. Thus, in this experiment, we estimated the performance of the system by the precision, which is in our case equal to the ratio of the number of correct outputs, to the number of inputs.

Table 2: The corpus used for the experiments

verb	data size	# of candidates	lower bound (%)
ataeru	136	4	66.9
kakeru	160	29	25.6
kuwaeru	167	5	53.9
noru	126	10	45.2
osameru	108	8	25.0
tsukuru	126	15	19.8
toru	84	29	26.2
umu	90	2	81.1
wakaru	60	5	48.3
yameru	54	2	59.3
total	1111	—	43.7

The training/test data used in the experiment contained over one thousand simple Japansese sentences collected from news articles. Each of the sentences in the training/test data used in our experiment consisted of one or more complement(s) followed by one of the ten verbs enumerated in table 2. In table 2, the column of "lower bound" denotes the precision gained in a naive method such that the system always chooses the interpretation most frequently appearing in the training data [5].

Figure 7 shows the relation between the size of the training data and the precision of the system. In figure 7, when the x-axis is zero, the system has used only the seeds given by IPAL. What can be derived from figure 7 is that with a given number of training data, the precision of random sampling was improved on by using the notion of utility of training, which can solve problem 1 and 2 mentioned in section 1. It can also be concluded that the size of the database can be reduced without degeneration of the system's performance, which can solve problem 3 in section 1.

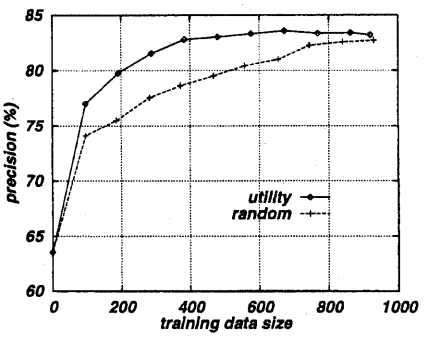

Figure 7: The relation between overhead and precision of the system

5 Conclusion

In this paper, we proposed an example sampling method for example-based verb sense disambiguation, and reported its performance through an experiment. In the experiment, it was shown that our method, which is based on the notion of the utility of training, has reduced the overhead for the training of the system, as well as the size of the database.

As we mentioned in section 1, the generalization of examples [7, 16] is another approach which realizes the reduction of the size of the database. The effectivity of this approach within the framework of our method is on issue that requires further exploration.

Future work will include the further sophistication of the the verb sense disambiguation system and the method of acquiring seeds, which is currently based on an existing dictionary. We will also construct a large-sized example base for natural language processing by using our example sampling method.

Acknowledgments

The authors would like to thank Dr. Manabu Okumura (JAIST, Japan) and Mr. Timothy Baldwin (TITech, Japan) for their comments on the 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 *the 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] Atsushi Fujii, Kentaro Inui, Takenobu Tokunaga, and Hozumi Tanaka. To What Extent Does Case Contribute to Verb Sense Disambiguation? *IPSJ SIG Notes*, Vol. 111, No. 9, pp. 55–62, 1996.

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

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

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

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

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

[10] 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.

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

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

[13] 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.

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

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

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

[17] Manabu Okumura. How to Resolve Semantic Ambiguity (in Japanese). *Journal of Japanese Society for Artificial Intelligence*, Vol. 10, No. 3, pp. 332–339, 1995.

[18] Satoshi Sato. MBT1: Example-Based Word Selection (in Japanese). *Journal of Japanese Society for Artificial Intelligence*, Vol. 6, No. 4, pp. 592–600, 1991.

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

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

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

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