自然言語処理 119－8
（1997. 5. 27）

単語多義性解消法の比較検討

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

本論文はコーパスに基づく単語多義性解消法の比較実験について報告する。近年提案されている手法の多くは、与えられた入力と単語語義を教示した例文の間の類似度に基づいて多義性解消を行う。日本語動詞を対象とした多義性解消の実験において、異なる類似度計算法を比較した結果、(a) 動詞と補語の統語構造、(b) 人手で作成したシソーラスを導入することによって最大で30%以上の多義性解消精度の向上が得られた。

A Comparative Evaluation of Recent Corpus-Based Word Sense Disambiguation Techniques

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

This paper describes an extensive comparative evaluation of recent corpus-based word sense disambiguation techniques, focusing around ten Japanese verbs. The basis of this task is the computation of the similarity between a given input and examples which have been annotated with its verb sense, and we compare different methods modeling this process. Through our experiments and discussion, we found the effective components of word sense disambiguation to be: (a) the use of the syntactic relation between a target verb and its complements (case pattern), and (b) the use of existing thesauri to approximate human knowledge.

1 Introduction

This paper describes an extensive comparative evaluation of recent word sense disambiguation (WSD) techniques, which represent a crucial component of numerous NLP applications. We currently focus on the sense disambiguation of Japanese verbs, for example, the following input sentence containing the sense ambiguous verb *tsukau*:

kodomo ga	*kozukai wo*	*tsukau.*
(children-NOM)	(allowance-ACC)	(?)

In Japanese, each verb complement consists of a noun phrase (case filler) and a case-marking suffix (case marker), for example *ga* (nominative), *ni* (dative) or *wo* (accusative). The "EDR" Japanese machine readable dictionary [3] defines multiple senses for the verb *tsukau*, sample of which are "to employ", "to operate" and "to spend". Among these candidates, one may notice that the correct interpretation of the *tsukau* in the above input is "to spend". Note that the task of disambiguation as discussed in this paper can be termed "categorization" because the plausible verb sense is selected from *predefined* candidates.

Reflecting the growing utilization of machine readable texts, a number of corpus-based WSD techniques have recently been proposed. The basis of these methods is the computation of the scored similarity between a given input and an example sentence set already annotated with its verb sense (mostly annotated by human experts). Suppose we have an example related to the sense "to spend":

kare ga	*chokin wo*	*tsukau.*
(he-NOM)	(savings-ACC)	("to spend")

One may notice that the verb in the input above can easily be interpreted as the sense "to spend" based on the previous example, given that the case fillers "*kozukai*" and "*chokin*" in the accusative are semantically similar.

While proposed methods have described their effectivity for a dedicated application, to the best of our knowledge no study has seriously compared different approaches. Figure 1 shows the flow of corpus-based word (verb) sense disambiguation, for the given three approaches. Prior to the core disambiguation process, morphological analysis, including lexical segmentation and part-of-speech tagging, is needed because Japanese sentences lack lexical segmentation. Given this input, there are two alternative methodologies. The first approach is sensitive to the syntactic and semantic content of complements the target verb requiring disambiguation, based on the intuitively feasible assumption that the sense of a verb is likely to be dependent on its syntactically governing complements. This approach is commonly used, and found in a number of verb sense disambiguation techniques [6, 12, 13, 22], most of which can be termed "example-based" sense disambiguation methods. This notion is also evident in a number of verb clustering methods [2, 10, 21].

The second approach simply relies on collocational information, disregarding syntactic relations. This approach is used in many WSD techniques for noun sense disambiguation [2, 17, 24].

We can further subdivide the syntactic approach into two different subapproaches. The first uses an example database (database, hereafter) which contains examples of the verb-complement structure, as in their original form. This can be called the *rigid* example-based approach. On the other hand, since this rigid approach is susceptible to the data sparseness problem, most example-based techniques divide the verb-complement structure into individual complements (in other words, the database contains a set of tuples of the form <noun, case marker, verb>), and allows any possible combination of complements marked with different cases. We shall call this the *integrated* example-based approach.

Subsequent the disambiguation process, we also estimate the degree of the certainty as to its interpretation, a topic which has not been discussed seriously in most WSD research.

Section 2 elaborates on the candidate methods given in figure 1 (bracketed by "{}"), and section 3 compares them by way of experiments. Discussion is added in section 4, followed by the conclusion.

2 The different methodologies

2.1 Syntactic analysis

Syntactic analysis, in which the verb-complement structure is extracted from its input, is especially poignant when the input comprises a complex sentence. To achieve this process, we have two candidate methods. On the one hand, full parsing with rich grammar rules is ideal. On the other hand, partial parsing with simple heuristics can be preferable because (a) the manual construction of a grammar is expensive, (b) automatic grammar acquisition does not seem to be advanced enough to be practical, and (c) we only need complements of the target verb, rather than a full syntactic analysis.

To conduct full parsing, we experimentally used the Japanese "QJP" parser [11]. We also used this parser as the morphological analyzer for all three WSD approaches focused on in this paper. In re-

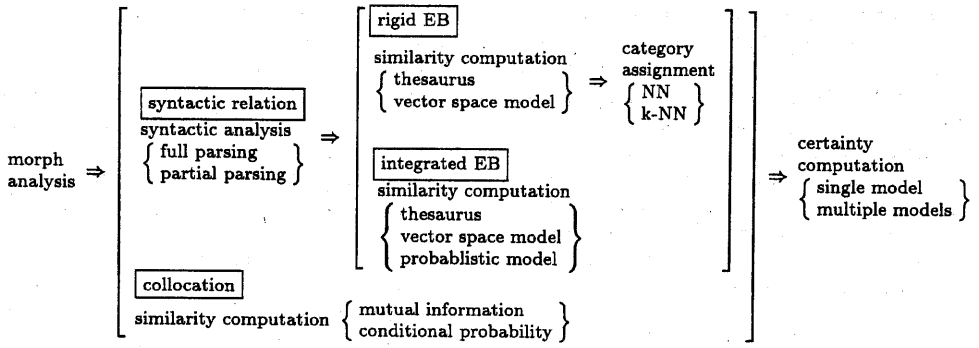


Figure 1: The flow of corpus-based verb sense disambiguation, and candidate methods for each process

gard to partial parsing, we introduced the two simple heuristics given below:

- each complement (noun + case marker) is associated to the predicate of highest proximity,
- complements containing the genitive case marker *no* are not considered because they can constitute either possessive or nominative case markers, and are thus confusing

It should be noted that based on the extracted syntactic structure, we discard verb sense candidates with case frames not corresponding to the obligatory case content of the input[1]. Those discarded candidates are not considered in the following process.

2.2 Similarity computation for the rigid example-based approach

In the *rigid* example-based approach, the similarity between an input and an example is computed based on the similarity between their respective case fillers. Figure 2 depicts a general schema for this notion, in which x denotes an input, and e denotes an example associated with verb sense s in the database. x_c and e_c denote the case fillers marked with case c, in x and e, respectively. In the following two paragraphs, we will explain two different ways to compute the similarity between case fillers.

input x:	x_{c_1}	x_{c_2}	...	x_{c_n}	v (?)
	↕	↕		↕	
example e:	e_{c_1}	e_{c_2}	...	e_{c_n}	v (s)

Figure 2: Similarity computation between the input and examples under the rigid example-based approach

[1]In the example-based approach, the case frame of a verb sense is given as the case pattern of each example associated with a given verb sense in the database.

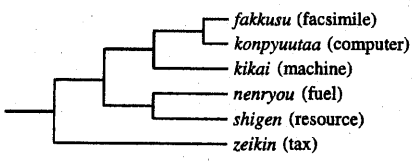


Figure 3: A fragment of the *Bunruigoihyo* thesaurus

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 relative 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

Hand-crafted thesauri Most example-based techniques [6, 12, 13, 22] use existing hand-crafted thesauri (for example, Roget's thesaurus [1], WordNet [14] or *Bunruigoihyo* [16]) for the similarity computation. For our experiments, we used the Japanese *Bunruigoihyo* thesaurus, which is commonly used in much NLP research. Figure 3 shows a fragment of the *Bunruigoihyo* thesaurus including some of the noun entries, with each noun corresponding to a leaf in the structure of the thesaurus. The similarity between two nouns can be measured based on the length of the path between them. We used the metric proposed by Kurohashi et al., as shown in table 1.

The *total* similarity between an input and examples is computed by summing the similarity between the input case filler and the example case filler for each case, as in equation (1).

$$sim(x, e) = \sum_{c \in x} sim(x_c, e_c) \qquad (1)$$

Here, $sim(x, e)$ denotes the similarity between the input x and example e, and $sim(x_c, e_c)$ denotes the similarity between the case fillers x_c and e_c, which can be measured based on table 1.

Vector space model There are a number of statistical models for word similarity computation, but the model with widest usage is the "vector space model" [2, 10, 21]. Unlike methods based on handcrafted thesauri, this model tends to avoid human overhead and bias. In this model, each noun n is represented by a vector comprising statistical co-occurrence factors. This can be expressed by equation (2), where $\vec{n}$ is the vector for the noun in question, and t_i is the co-occurrence statistics of n and each co-occurring verb.

$$\vec{n} = < t_1,\ t_2,\ \ldots,\ t_i,\ \ldots > \qquad (2)$$

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 [4], in which t_i is calculated as in equation (3), where $f(<n, c, v>)$ is the frequency of the tuple $<n, c, v>$, $f(<c, v>)$ is the frequency of tuple $<c, v>$, and N is the total number of the tuples within the overall co-occurrence data.

$$t_i = f(<n, c, v>) \cdot \log \frac{N}{f(<c, v>)} \qquad (3)$$

We then compute the similarity between nouns x_c and e_c by the cosine of the angle between the two vectors $\vec{x_c}$ and $\vec{e_c}$. This is realized by equation (4).

$$sim(x_c, e_c) = \frac{\vec{x_c} \cdot \vec{e_c}}{|\vec{x_c}||\vec{e_c}|} \qquad (4)$$

In regard to the total similarity between the input and an example, we simply use equation (1).

2.3 Category assignment

In category assignment, we compute the score for each sense candidate of the input verb, based on the similarity computed in the previous process, and then select the sense with maximal score.

Nearest neighbor In the nearest neighbor (NN) method, only the example which is most similar to the input is considered in the score computation. The score of verb sense s, $Score(s)$, is expressed by equation (5), where $sim(x, e)$ denotes the similarity between input x and example e, and $\mathcal{E}_s$ denotes the example set associated with sense s in the database.

$$Score(s) = \max_{e \in \mathcal{E}_s} sim(x, e) \qquad (5)$$

k-nearest neighbor In the k-nearest neighbor (k-NN) method [7], k most similar examples ($\mathcal{K}$) "vote" on the sense of the input verb[2]. That is, the score of verb sense s is expressed by the number of examples associated with sense s in $\mathcal{K}$.

2.4 Similarity computation for the integrated example-based approach

Figure 4 shows a fragment of the entry associated with the verb *tsukau* for the *integrated* example-based approach. The database lists several case filler examples for each case. The difference with the *rigid* approach is that since this approach allows any possible combinations of case filler examples (marked with different cases), the number of example sentences is implicitly increased by a combinatorial factor. Thus this approach tends to avoid the data sparseness problem. On the other hand, this method ignores collocational restriction between complements of a verb. We will further discuss this trade-off through experiments in section 3.

Based on the proposed methods [12, 22], we compute the score for sense s by way of equation (6), where $sim(x_c, e_c)$ for either a thesaurus-based method or vector space model can be computed in exactly the same way as for the rigid approach (see section 2.2).

$$Score(s) = \sum_{c \in x} \max_{e_c \in \mathcal{E}_s} sim(x_c, e_c) \qquad (6)$$

2.5 Probabilistic model

From the viewpoint of probability theory, one may propose a naive approximation of the probability that an input x takes a verb sense s, by computing the product of the probability that each verb sense s takes x_c as its complement (equation (7)).

$$P(x{=}s) \simeq \prod_{c \in x} P(<x_c, c, s> | x_c) \qquad (7)$$

However, since each x_c does not always appear in the database due to data sparseness, we need to employ a smoothing technique. Grishman et al. proposed a probabilistic smoothing method into the task of acquiring verb selectional patterns [9]. The essence of this method is to compute the "confusion probability", $P_C(n|n')$, which indicates the possibility that a noun n can be replaced with a different

[2] One may argue that NN is a special case of k-NN: the score of a verb sense s is 1 when the example most similar to an input is associated with s, otherwise the score is 0. However, a scalable score is more preferable when we compute the degree of the certainty of the interpretation (see section 2.7).

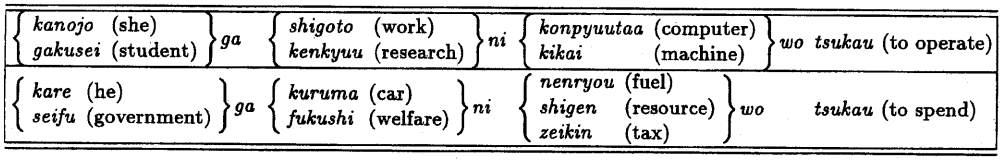


Figure 4: A fragment of the database for the integrated example-based approach, and the entry associated with the Japanese verb *tsukau*

noun n' in a given context (equation (8)).

$$P_C(n|n') = \sum_{<c,v>} P(n|<c,v>) \cdot P(<c,v>|n') = \frac{f(<n,c,v>)}{f(<c,v>)} \cdot \frac{f(<n',c,v>)}{f(n')} \quad (8)$$

All of the statistical factors in equation (8) are derivable from the co-occurrence data taken from the RWC text base (see section 2.2). Introducing P_C into equation (7), we replace the probability $P(<x_c,c,s>|x_c)$ with a *smoothed* factor based on the product of (a) the probability that verb sense s takes an example case filler e_c as its complement in the database, and (b) the confusion probability of x_c and e_c. This notion can be expressed by equation (9), where the example case filler which maximizes the smoothed factor is considered in the score computation.

$$Score(s) = \prod_{c \in x} \max_{e_c \in \mathcal{E}_s} P_C(x_c|e_c) \cdot P(<e_c,c,s>|e_c) \quad (9)$$

Since $P(<e_c,c,s>|e_c)$ requires that the co-occurrence data be annotated with appropriate verb senses, we used the tuple entries in the database rather than those take from the RWC text base.

2.6 Similarity computation based on collocation

By way of using word collocation instead of syntactic relations, the database comes to contain not only complements of verbs but also collocations. In real terms, we only used collocational information for nouns because functional words such as case markers are generally more noisy than informative. Based on the database, the statistical factor for each collocating word and verb sense, that is, the degree of association between them, is calculated prior to sense disambiguation. In regard to the score for each verb sense, a commonly proposed implementation is used, in which the score is computed by summing the statistical factor of each collocating word which appears in a given input [23], as in equation (10).

$$Score(s) = \sum_{w \in input} A(s,w) \quad (10)$$

Where $A(s,w)$ denotes the degree of association between sense s and each collocating word w. In the following two paragraphs, we explain two types of statistical factors which were used in our experiments.

Mutual information The notion of *mutual information* is used in much NLP research for estimating the degree of the association between two given terms. In this implementation, $A(s,w)$ is expressed by equation (11), where $f(s,w)$ denotes the frequency of w collocating with sense s, and $f(s)$ and $f(w)$ denote the frequency of w and s, respectively. All these factors are calculated based solely on the database[3].

$$A(s,w) = \frac{f(s,w)}{f(s) \cdot f(w)} \quad (11)$$

Conditional probability Another implementation calculates $A(s,w)$ as the probability that s occurs, when w occurs [20]. This is expressed by equation (12), closely resembling that for mutual information.

$$A(s,w) = \frac{f(s,w)}{f(w)} \quad (12)$$

2.7 Certainty computation

Since recent WSD techniques still find 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 purposes. To achieve this, it is useful to estimate the degree of the certainty as to the interpretation, so that we can gain higher accuracy selecting only outputs with greater certainty degree.

Single model Fujii et al. proposed a method for the computation of interpretation certainty [5]. This is based on the following preference conditions:

1. the highest score ($Score_1$) is greater,
2. the difference between the highest and second highest scores ($Score_1 - Score_2$) is greater.

This notion is expressed by equation (13), where $C(x)$ is the interpretation certainty of an input x.

$$C(x) = Score_1 + (Score_1 - Score_2) \quad (13)$$

[3]We estimated $f(s)$ as the number of sentences associated with sense s in the database.

Multiple models In this paper, we newly introduce another method for certainty computation, founded on the rationale of asking for a "second opinion". Intuitively speaking, in similarity computation, if complementary methods like "thesaurus-based method" and "vector space model" make the same interpretation, the certainty of their interpretation is expected to be great. Let s_a and s_b be verb senses for an input x selected by different methods. The interpretation certainty for x is expressed by equation (14), in which unlike the single model method, the certainty only takes a boolean value.

$$C(x) = \begin{cases} \text{"certain"} & \text{if } s_a = s_b \\ \text{"uncertain"} & \text{otherwise} \end{cases} \quad (14)$$

3 Evaluation

3.1 Sentence set

We collected sentences (as test/training data) from the EDR Japanese corpus [3] (originally produced from news articles). The EDR corpus provides sense information for each word, based on the EDR dictionary, and we used this as a means of checking the interpretation. Our derived corpus contains ten verbs frequently appearing in the EDR corpus, which are summarized in table 2. In table 2, the column of "English gloss" describes typical English translations for the Japanese verbs. The column of "# of sentences" denotes the number of sentences for that verb in the corpus, while "# of senses" denotes the number of verb senses, based on the EDR dictionary. For each of the ten verbs, we conducted 4-fold cross validation: that is, we divided the corpus into four equal parts, and conducted four trials, in each of which a different one of the four parts was used as test data and the remaining parts were used as training data (the database).

Table 2: The verbs contained in the corpus used

verb	English gloss	# of sentences	# of senses
dasu	evict	967	5
kaku	write	503	2
kuwaeru	add	516	4
miru	see	1476	17
motomeru	request	1123	5
motsu	hold	1636	12
moukeru	establish	399	3
okuru	send	434	9
tsukau	spend	2117	7
ukeru	receive	1709	10
total	—	10880	—

3.2 Comparative experiments

The number of possible combinations of candidate methods is so great that we only present the most representative combinations here, based on preliminary evaluation of all possible combinations. First, we compared the following seven methods (hereafter, we shall use the notation "BGH" and "VSM" for the *Bunruigoihyo* thesaurus, and the vector space model, respectively).

(1) rigid EB + BGH + NN
(2) rigid EB + VSM + NN
(3) integrated EB + BGH
(4) integrated EB + probabilistic model
(5) collocation + mutual information
(6) collocation + conditional probability
(7) lower bound: a naive method, in which the system systematically chooses the verb sense appearing most frequently in the database [8].

In this evaluation, methods (1) to (4) used the full parsing method with the QJP parser. In Japanese, complements of a verb are not always provided because they are often omitted if they are easily predictable (based on human judgment) from the context. In such a situation, methods (1) to (4) simply use method (7). Table 3 shows the accuracy of each method, given that the accuracy is the ratio of the number of correct interpretations, to the number of outputs. It should be noted that according to our preliminary observation, the EDR corpus contains a number of sense tagging errors, and we assume this is why the accuracy of each approach was generally not acceptable. However, all methods except (6) outperformed the lower bound results produced by method (7). We can derive from the overall results that (a) the use of syntactic relations outperforms the use of collocation, (b) in both the rigid and integrated example-based approaches, the use of BGH outperforms other similarity computation methods, and (c) rigidity and integration in the example-based approach are quite competitive when BGH is used. In the following, we mainly focus on the rigid example-based approach (methods (1) and (2)).

Table 4 shows the accuracy on different methods of category assignment, in which the nearest neighbor method is superior to k-NN, and a greater value of k led to diminished accuracy in general.

Table 5 shows the accuracy using the different parsing methods of full parsing and partial parsing. The two parsing techniques did not yield a significant difference in terms of the accuracy. Therefore, we can improve on the performance of verb sense disambiguation without the considerable overhead for syntactic analysis.

Finally, let us evaluate the effectivity of the certainty computation methods, by the trade-off be-

Table 3: The accuracy of each method

(1)	(2)	(3)	(4)	(5)	(6)	(7)
62.6	59.1	61.5	54.9	50.1	29.2	44.9

Table 4: The accuracy of NN and k-NN with different values of k

	NN	$k = 5$	$k = 10$	$k = 20$
BGH	62.6	55.0	54.5	53.5
VSM	59.1	50.0	48.5	49.1

Table 5: The accuracy with full parsing and partial parsing

	BGH	VSM
full parsing	62.6	59.1
partial parsing	62.0	59.0

tween the accuracy and the applicability, given that the applicability is the ratio between the number of cases where the interpretation certainty of the outputs is above a certain threshold, and the number of inputs[4]. To estimate this trade-off, for the single model method, we progressively increased the value of the threshold on the interpretation certainty, and investigated the relation between the applicability and the accuracy. In this case, we used BGH for the similarity computation. On the other hand, in the multiple model method, we used BGH and VSM as the two different models, and calculate the applicability as the ratio of the number of outputs labeled with "certain", to the total number of outputs. Figure 5 shows the results, in which the x-axis denotes the applicability, and the y-axis denotes the accuracy. From this figure, we can conclude that (a) we obtained higher accuracy than that without the degree of certainty, for example, the accuracy becomes over 80% when the applicability is 50%, and (b) the multiple models did not further improved on this trade-off.

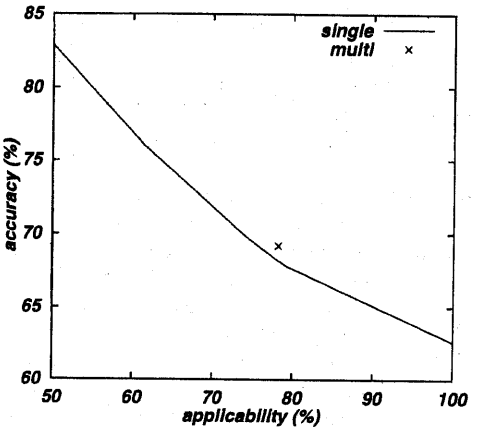


Figure 5: The trade-off between the applicability and accuracy

4 Discussion

As far as our experiments in section 3 go to show, the rigid example-based approach did not greatly improve on the accuracy of the integrated approach. However, in the EDR corpus, a lot of complements are missing because of ellipsis or zero anaphora. Assuming successful ellipsis/anaphora analysis providing the complete complement pattern, the rigid approach is expected to further outperform other approaches.

From the relative inferiority of mutual information in table 3, we assume that this factor gives lower values for frequently appearing verb senses, and therefore those verb senses are rarely selected as the interpretation.

One may argue that given sufficient statistics, the vector space model should outperform hand-crafted thesauri. We investigate this prediction in table 6, which shows the the relation between the frequency of nouns appearing in the co-occurrence data and the accuracy of verb sense disambiguation, in which the "frequency" entry denotes the threshold of the frequency of nouns. The "coverage" entry denotes the ratio between the number of inputs including at least one noun with frequency over a given threshold, and the total number of inputs. The last two entries show the accuracy with different similarity measures, for each coverage. Surprisingly, not only the accuracy of VSM but also the accuracy of BGH increased as the threshold of the frequency increased, and VSM did not outperform BGH for any of the thresholds. We could assume that (a) nouns which frequently appear in the co-occurrence data also appear in the database, and therefore they provide the maximum similarity (that is, "exact matching") independent of which similarity measure is used, and (b) frequently appearing nouns are used so commonly that even human lexicographers can reasonably define the similarity between them in a thesaurus. Putting this result aside, we do not believe our co-occurrence data was insufficient because it was taken from *four* years worth of newspaper articles.

5 Conclusion

In this paper, we reviewed recent approaches for verb sense disambiguation and compared them by way of experiments, to an extent previously unat-

[4]Note that this figure differs from the recall, which is commonly used in NLP research.

Table 6: The frequency of nouns and resultant accuracy of verb sense disambiguation

frequency	≥100	≥1000	≥10000
coverage (%)	73.9	58.2	16.8
BGH	68.7	72.3	74.7
VSM	65.0	69.9	73.4

tempted, as far as we know. Through evaluation, we concluded that the following three features are useful: (a) syntactic relations between a target verb and its complements, (b) the rigidity of maintaining complement patterns rather than decomposing them into individual complements, and (c) handcrafted thesauri for the similarity computation. Future work will include the evaluation of the effectivity of other information, including discourse analysis [15, 24], which the EDR corpus does not provide.

Acknowledgments

The authors would like to thank Mr. Masayuki Kameda (RICOH Co., Ltd., Japan) for his support with the QJP parser, Mr. Timothy Baldwin (TITECH, Japan) for his comments on the earlier version of this paper, and Mr. Naoyuki Sakurai (TITECH, Japan) for aiding with the experiments.

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

[5] 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. `http://xxx.lanl.gov/ps/cmp-lg/9702010`.

[6] 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.

[7] Keinosuke Fukunaga. *Introduction to Statistical Pattern Recognition*. Academic Press Inc., 1972.

[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] Ralph Grishman and John Sterling. Generalizing automatically generated selectional patterns. In *Proceedings of COLING*, pp. 742–747, 1994.

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

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

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

[15] Tetsuya Nasukawa. Discourse constraint in computer manuals. In *Proceedings of TMI*, pp. 183–194, 1993.

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

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

[18] Real World Computing Partnership. RWC text database. `http://www.rwcp.or.jp/wswg.html`, 1995.

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

[20] Takenobu Tokunaga and Makoto Iwayama. Text categorization based on weighted inverse document frequency. *Information Processing Society of Japan, SIGNL*, Vol. 94, No. 100, pp. 33–40, 1994.

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

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

[23] David Yarowsky. Word-sense disambiguation using statistical models of Roget's categories trained on large corpora. In *Proceedings of COLING*, pp. 454–460, 1992.

[24] David Yarowsky. Unsupervised word sense disambiguation rivaling supervised methods. In *Proceedings of ACL*, pp. 189–196, 1995.