Combining Example and Statistical Based Methods of Word Sense Disambiguation towards Reading Assistant System

By Pulkit Kathuria

A thesis submitted to
School of Information Science,
Japan Advanced Institute of Science and Technology,
in partial fulfillment of the requirements
for the degree of
Master of Information Science
Graduate Program in Information Science

Written under the direction of
Associate Professor Kiyoaki Shirai

March, 2013

Combining Example and Statistical Based Methods of Word Sense Disambiguation towards Reading Assistant System

By Pulkit Kathuria (1010205)

A thesis submitted to
School of Information Science,
Japan Advanced Institute of Science and Technology,
in partial fulfillment of the requirements
for the degree of
Master of Information Science
Graduate Program in Information Science

Written under the direction of
Associate Professor Kiyoaki Shirai

and approved by
Professor Akira Shimazu
Professor Hiroyuki Iida

February, 2013 (Submitted)

Contents

List of Figures

List of Tables

Abstract

This thesis presents a precision oriented example based approach for word sense disambiguation (WSD) for a reading assistant system for Japanese learners. Our WSD classifier chooses a sense associated with the most similar sentence in a dictionary only if the similarity is high enough, otherwise chooses no sense. We propose sentence similarity measures by exploiting collocations and syntactic dependency relations for a target word, that measures similarity between two sentences. The example based classifier is combined with a Robinson classifier to compensate recall. First precision oriented example based classifier is applied, if it cannot choose a sense then Robinson classifier is applied. We further improve WSD performance by automatically acquiring bilingual sentences from a parallel corpus. Automatic acquisition of example sentences also enables us to prepare more examples to be shown to the user in the Reading Assistant System. For WSD we present two manually annotated data sets from 49 target words including nouns, verbs and adjectives. According to the results of our experiments, the accuracy of automatically extracted sentences was 85%, while the proposed WSD method achieves 65% accuracy which is 7% higher than the baseline.

Chapter 1

Introduction

1.1 Motivation

Japanese learners often look up words in dictionaries/internet when they read Japanese documents. One word has several possible translations, although a word has only one meaning when it appears in the document. It is rather hard for non-native readers of Japanese to read definition sentences of all meanings. It would be useful to build a system which can not only show the target word's definition sentence in English and its example usage but also select the correct meaning. In this research we focus on building a Reading Assistant System that aims to assist Japanese language learners to properly understand the meanings of Japanese words in context.

Currently, ASUNARO[1] is the only reading assistant system for Japanese learners with Word Sense Disambiguation (WSD hereafter) module. However, definition sentences of EDR dictionary[2] produced by ASUNARO are sometimes unnatural and no example sentence is shown for each sense. There exists a more appropriate dictionary for language learners. For example, EDICT[3], the Japanese-English bilingual dictionary that includes definition sentences in English as well as example sentences in Japanese and English. We believe that example sentences are indispensable for Japanese learners to understand meanings of words.

[1] http://hinoki.ryu.titech.ac.jp/asunaro/index-e.php

[2] http://www2.nict.go.jp/out-promotion/techtransfer/EDR/index.html

[3] http://wow.csse.monash.edu.au/~jwb/edict.html

1.2 Goal

The goal of this research is to develop a novel Reading Assistant system for Japanese learners that shows both definition and example sentences for a target word. It also disambiguates the sense of the target word in the context, then only show the information about the correct meaning.

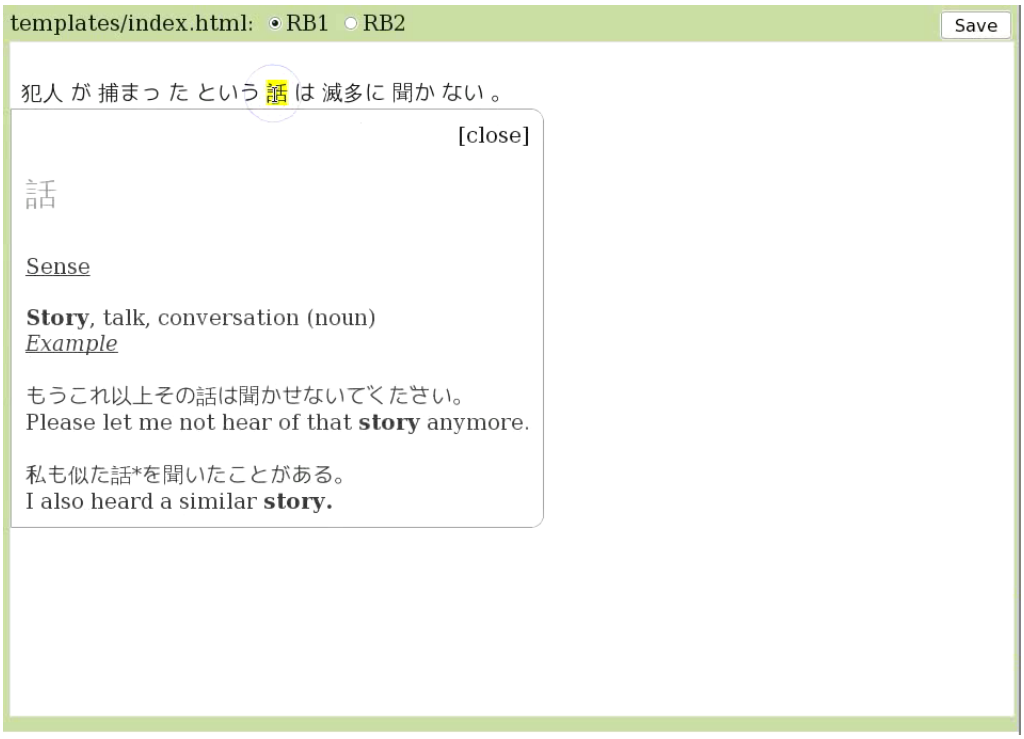


Figure 1.1: Snapshot of Reading Assistant System

Figure 1.1 shows the snapshot of the proposed user interface of the Reading Assistant System, where the input sentence can either be typed or pasted in the editor. The Japanese word 話 (*hanashi*) has two meanings, one is "story" and the other is "discussions". The reading assistant system chooses the correct meaning and only the information associated with the correct meaning ("story, talk, conversation" shown in Figure 1.1) of target word 話 (*hanashi*; story) is presented. It also shows example sentences in both Japanese and English. The operations for the end user in this proposed interface are straightforward. Upon clicking the desired word in context, the system presents the disambiguated sense definition and the chosen similar example sentences, in a pop up window. And the user can resume back to reading the text by dismissing

the pop up, by clicking the close button. Because this interface[4] is coded in HTML5, Javascript and jQuery, it can be opened in any web browser across various operating systems including smartphones (javascript enabled).

In the next subsections we introduce our approach to build the Reading Assistant System: the example based word sense disambiguation, machine learning and combination of classifiers and automatic acquisition of example sentences.

1.2.1 Example based WSD

In this research, example-based WSD is considered, since it would be suitable for our reading assistant system showing examples to users. It should handle all words, including low frequency words, in a document. Therefore our WSD method does not rely on a sense tagged corpus that requires much human labor to construct, although many of current WSD methods use supervised machine learning [15, 17]. For example based WSD, we have used EDICT's (Japanese-English dictionary) as sense inventory and example database. We present the details of EDICT in Section 3.1 and our example based classifiers in Section 3.2.

1.2.2 Machine Learning and Combination

We propose an example based classifier that uses EDICT as example database and is designed to choose a sense only in reliable cases, that there is a similar sentence in example database. Such a system would achieve high precision but low recall. To compensate recall, our example based method is combined with a more robust WSD method based on machine learning. Details of the machine learning namely Robinson classifier are presented in Section 3.4 & combination of classifiers is shown in Section 3.5.

[4]The source code of this interface is available on Github https://github.com/kevincobain2000/reading-assistant-system

1.2.3 Example Sentences Expansion

Although, EDICT has 150,000 example sentences for 160,000 lexical entries, the number of example sentences might be enough for high frequency words but as we move towards rare senses or low frequency words there are a few or no example sentences available. Moreover, among high frequency words there are many senses which have no examples. It is difficult to disambiguate a sense in such cases and moreover no example usage can be shown to the user in the Reading Assistant System.

To tackle this problem, we automatically extract bilingual example pairs from parallel corpora. It enables us to improve the performance of WSD classifiers as well as prepare more examples to be shown to users. We show in detail about the parallel corpora used in Section 4.1 and our approach on extracting reliable parallel example sentences for senses in Section 4.2.

1.3 Organization of this Thesis

Further this thesis is organized as follows.

Chapter 2 discusses the related work on Word Sense Disambiguation (WSD) and example sentences expansion. Chapter 3 presents the details about the proposed WSD method which includes the example based, machine learning and combination of respective classifiers. Chapter 4 follows the proposed method detailing about our approach to automatically acquire example sentences from parallel corpora. Chapter 5 presents data, results and discussion on empirical evaluation on example sentences expansion and proposed WSD method. Chapter 6 concludes this thesis.

Chapter 2

Related Work

Our research can be divided into two separate tasks. One is the word sense disambiguation and the other is labeled data expansion. Further in this chapter, we talk about the previous work on both successively.

2.1 Word Sense Disambiguation

WSD in our reading assistant system is a task of translation selection in machine translation. It has been shown in previous researches that lexical translation selection by using WSD helps to improve performance and quality in Machine Translation (MT). Carpuat et al. and Chan et al. integrated WSD in MT and showed significant improvements in terms of general MT quality, on a Chinese-English MT framework [4, 5].

Many researches in translation selection have been devoted. Dagan et al. proposed a method to use word co-occurrence in target language corpus [6]. Later approaches adopting co-occurrence statistics use simple mapping information between a source and its target words. Lee et al. showed the defect of using 'word-to-word' translation and proposed a translation selection method based on the 'word-to-sense' and 'sense-to-word' [10].

While example based Japanese WSD has also been studied. For example, Fujii et al. proposed a method for verb sense disambiguation, which measures sentence similarity based on semantic similarity of case filler nouns and weights of cases considering the influence of the case for WSD

[8]. Then they proposed a method of selective sampling to reduce the cost of sense tagging to construct an example database. Target words are restricted to verbs in their research, while WSD of nouns and adjectives is also considered in this research. Shirai et al. proposed a method to disambiguate a sense of a word in a given sentence by finding the most similar example sentence in monolingual dictionary [21]. However, their method used only syntactic relations for measuring the similarity between sentences, while our method also considered collocation including the target word.

In our knowledge, this research is the first attempt which tackles Japanese WSD task by using sense inventory taken from EDICT. In previously studied example based classifier by Shirai et al., example sentences from Japanese dictionary *Iwanami Kokugo Jiten* were used. As *Iwanami* was originally a paper dictionary, sentences consists of an average of 4 words due to space constraints, hence are fragmented and do not provide sufficient information for classification. Long length of example sentences from EDICT leverages us to classify senses more precisely. Shirai et al. also proposed a method combining example based WSD with Support Vector Machine (SVM) trained from a sense tagged corpus. While our method does not rely on a sense tagged corpus which requires much cost to construct.

2.2 Labeled Data Expansion

Approaches for automatic expansion of labeled example sentences have been seen in recent years. Fujita et al. expanded the labeled data by collecting sentences that include an exact match for example sentences in *Iwanami* dictionary [9]. Note that the extracted sentences would be much longer than ones in the dictionary, providing more information for WSD. Sentences extracted by Fujita's method are homogeneous since only sentences similar to examples in the dictionary are obtained. While, our method can retrieve heterogeneous or wide variety of example sentences, which would be more suitable for WSD.

A different approach suggested by Mihalcea finds example sentences by using a set of seed expressions to create appropriate queries to Web search engines [12]. For example, for the fibre optic channel sense of word 'channel', appropriate queries would be 'optical fiber channel', 'channel telephone', 'transmission channel'. This method works well when such multi-word

constructions can be constructed.

Melo et al. used a parallel corpus and relied on an aligned sense inventory with sense tagged corpus to extract sense disambiguated example sentences [7]. In which sense of the target word is disambiguated by looking at the information in both language pairs individually. This approach is able to cover senses with no prior examples. However, disambiguation relies on an aligned sense inventory and sense tagged corpora for two languages. Furthermore they proposed an algorithm, which chooses a set of valuable example sentences to showcase to user, by employing a weighing scheme using n-grams. However, they did not use extracted example sentences for WSD.

Chapter 3

Proposed WSD Method

In this chapter, we present the details of the proposed method for WSD. First, EDICT, a sense inventory used in this research, is introduced in Section 3.1. Following that, two WSD classifiers are presented: Example based classifiers (Section 3.2 and 3.3) and Robinson classifier (Section 3.4). Our final classifier is an ensemble of the above classifiers, presented in Section 3.5.

3.1 EDICT

話

S1: story, talk , conversation , speech, chat

E1: もうこれ以上その**話** は 聞かせな いでください。

(Please let me not hear of that story any more.)

S2: discussions, argument, negotiation

E2: 3時間議論したが、我々は**話**がまとまらなかった。

(After 3 hours of discussion we got nowhere.)

Figure 3.1: EDICT, Sense and Example Sentences of 話 (*hanashi*; story or discussions)

In this thesis, word senses or meanings are defined according to the Japanese-English dictionary EDICT. The EDICT is a freely-available dictionary, developed by Monash University, and is widely used as a source of lexical material in dictionary systems and text-processing projects. It includes 160,000 Japanese word entries with 150,000 Japanese-English example sentences from Tanaka corpus[1], where each sentence is manually annotated by the developers of EDICT. Figure 3.1 shows the entry of the word **話** (*hanashi*; story or discussions) with its 2 senses and respective example sentences.

3.2 Example Based WSD

In this section we present the details of the proposed example based WSD and its associated similarity measures, collocation $coll(I, E)$ and syntactic similarity *syn(I,E)* between input I and example sentence E.

3.2.1 Overview

In EDICT, word definitions often contain example sentences in both Japanese and English. We develop the WSD classifier that calculates similarity between the input Japanese sentence and example sentences from a dictionary. Then choose example sentence which is the most similar to the input sentence.

Figure 3.1 shows the sense definitions **S** and example sentences **E** for the Japanese noun 話 (*hanashi*; story or discussions). For example, let us consider the case where the word sense of 話 (*hanashi*) is to be disambiguated in input Japanese sentence **I**.

I 犯人が 捕まったという**話**は滅多に聞かない。

(Rarely hear a story that the culprit got caught.)

As shown in the Figure 3.2, the classifier measures the similarity between input sentence **I** and the example sentences **E1** and **E2**. Among them, **E1** has the highest similarity with **I**, with

[1] http://wow.edrdg.org/wiki/index.php/Tanaka_Corpus

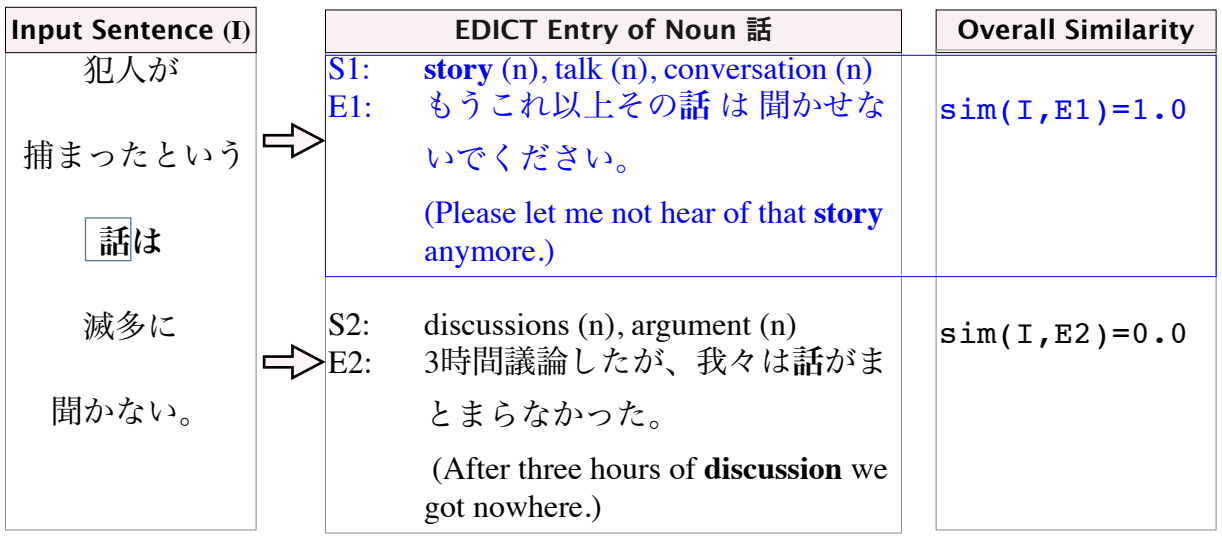


Figure 3.2: Overview of Example Based WSD

an overall similarity score $sim(I, E1) = 1.0$. Therefore, the classifier selects **S1** (story) as the correct sense definition for the word 話 (*hanashi*).

In order to choose example sentence (E) which is most similar to an input sentence (I), we build an example based classifier which measures overall similarity $sim(I, E)$ as a sum of collocation similarity $coll(I, E)$ and similarity calculated by comparing syntactic relations $syn(I, E)$. It chooses the sense associated with the example sentence whose overall similarity score $sim(I, E) = coll(I, E) + syn(I, E)$ is highest and doesn't choose any sense if the overall score is less than or equal to a threshold T, because the classifier cannot find an example sentence similar enough. Rare cases, when two or more senses have same score, sense with highest number of examples is chosen. Two similarity measures $coll(I, E)$ and $syn(I, E)$ are explained next.

3.2.2 Collocation Similarity

$coll(I, E)$ refers to collocation similarity score based on match sequences of n-grams of sizes 4, 5 and 6 between sentences I and E. 4-grams are a sequence of 4 words including a target word from a sentence. As shown below, 4 sequences from 4-grams are obtained where **TW** is the target word and w_{-1} and w_1 are previous and next word to the target word, respectively and

so on.

$$
\begin{aligned}
&w_{-3}\text{-}w_{-2}\text{-}w_{-1}\text{-}\mathbf{TW} \\
&\quad w_{-2}\text{-}w_{-1}\text{-}\mathbf{TW}\text{-}w_1 \\
&\quad\quad w_{-1}\text{-}\mathbf{TW}\text{-}w_1\text{-}w_2 \\
&\quad\quad\quad \mathbf{TW}\text{-}w_1\text{-}w_2\text{-}w_3
\end{aligned}
\tag{3.1}
$$

Sequences for 5-grams and 6-grams are defined in the same way. $coll(I, E)$ score by using n-grams is calculated as per Equation (3.2). Weights for n-grams are determined in ad-hoc manner.

$$
coll(I, E) = \begin{cases}
1 & \textbf{if} \quad \text{one of 6-grams is same} \\
0.75 & \textbf{elif one of 5-grams is same} \\
0.5 & \textbf{elif one of 4-grams is same} \\
0 & \textbf{otherwise}
\end{cases}
\tag{3.2}
$$

3.2.3 Syntactic Similarity

$syn(I, E)$ refers to syntactic similarity between two sentences I and E, for which we exploited the Japanese dependency structure usually represented by the linguistic unit called *bunsetsu*, which is a chunk consisting one or more content words and zero or more functional words. We use the same input sentence **I1** as an example to show such dependency structure in Figure 3.3. Each *bunsetsu* has one head which is represented by bold face, followed by a *case marker* such as が (*ga*), は (*ha*)[2] or other functional words. Each head *bunsetsu* is always placed to the right of its modifier and the dependencies do not cross each other. We obtain such Japanese dependency structure by using analyzer Cabocha[3].

No parser is 100% accurate and Cabocha is no exception. Although, instances where a wrong output is produced are not handled and is out of scope of this research. However, keeping incorrect instances produced by Cabocha in mind, we focus on precision oriented and strict syntactic similarity measure, details of which follows next.

[2] が (*ga*) and は (*ha*) are nominative (NOM) and topic (TOP) case markers, respectively.

[3] http://code.google.com/p/cabocha/

We calculate $syn(I, E)$ by comparing syntactic relations r extracted from *bunsetsu* dependencies as:

$$
\begin{aligned}
r &= w_1 - rel - w_2 \\
rel &= \begin{cases} case\,marker & \textbf{if} \text{ case marker follows } w_1 \\ \text{adnominal} & \textbf{elif} \text{ POS}(w_2) = \text{Noun} \\ \text{adverbial} & \textbf{otherwise} \end{cases}
\end{aligned} \tag{3.3}
$$

Where w_1 and w_2 are a head of modifier and modifiee *bunsetsus* respectively and rel is the relation type. In the classifier, not all but only relations where either w_1 or w_2 is a target word are extracted. r_1 and r_2 below are the extracted relations for sentence **I1** from its dependency structure shown in Figure 3.3.

r_1 : 捕ま −adnominal− **話**
r_2 : 話 −は− 聞か

Head word 捕ま (*tsukama*; catch) of *bunsetsu* #2 directly modifies bunsetsu #3, where head is the target word 話 (*hanashi*; story). Further ahead, 話 (*hanashi*; story) directly modifies *bunsetsu* #5, therefore head 聞か (*kika*; hear) is extracted as w_2 in r_2.

Next, $syn(I, E)$ is defined as follows.

$$
syn(I, E) = \sum_{(r_i, r_e) \in R_I \times R_E} s_r(r_i, r_e) \tag{3.4}
$$

$$
s_r(r_i, r_e) = \begin{cases} s_w(r_i(w_1), r_e(w_1)) & \textbf{if } r_i(w_2) = r_e(w_2) = t \\ & \text{and } r_i(rel) = r_e(rel) \\ s_w(r_i(w_2), r_e(w_2)) & \textbf{if } r_i(w_1) = r_e(w_1) = t \\ & \text{and } r_i(rel) = r_e(rel) \\ 0 & \textbf{otherwise} \end{cases} \tag{3.5}
$$

$$
s_w(w_i, w_j) = \begin{cases} 1 & \textbf{if } w_i = w_j \\ \frac{x}{8} & \textbf{otherwise} \end{cases} \tag{3.6}
$$

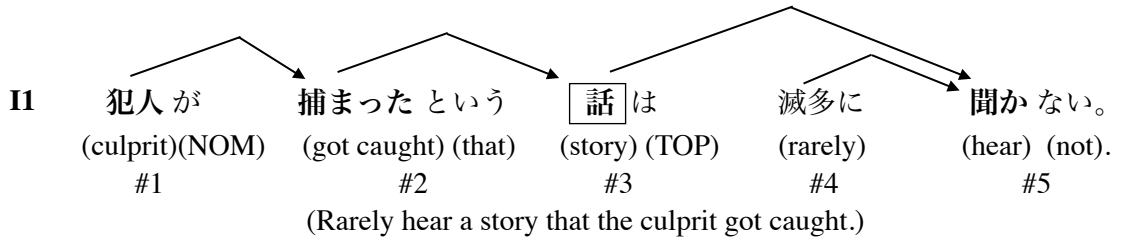


Figure 3.3: Example of *Bunsetsu* Dependencies

In Equation (3.4), $syn(I, E)$ is the sum of similarity scores $s_r(r_i, r_e)$ obtained by comparing all relations r_i and r_e extracted from input and example sentence respectively. Equation (3.5) compares two relations of same relation type rel and whose respective target word t is of same dependency structure in both relations i.e. either modifier or modifiee. Finally similarity of such relations is calculated by semantic similarity between words $s_w(w_i, w_j)$ as Equation (3.6). Here w_i and w_j are modifier words from two relations that modifies the target word, vice versa are modifiee words when target word is the modifier. Note that $w_i$and w_j stand for the base form (not surface form) of words. In Equation (3.6), x is the length of common prefix of semantic codes of two words in *Bunrui Goi Hyo* [14]. $s_w(w_i, w_j)$ is normalized to limit its score to < 1[4]. Although similarity between two words can also be measured using Japanese Wordnet[5], but it is only limited to nouns and might result in low recall. Therefore we have used *Bunrui Goi Hyo* in this research.

3.3 Classifiers RTW & RCW

In calculation of $syn(I, E)$ as shown in the previous subsection, only syntactic relations with respect to a target word are considered to measure the similarity between two sentences. It might be problematic because they seem insufficient to calculate sentence similarities precisely. To use more information for measuring similarity between sentences, we pay attention to common words in two sentences.

For syntactic similarity, relations with respect to not only target word but also common words

[4]Note that a semantic code in *Bunrui Goi Hyo* is represented as 7 digits.

[5]http://nlpwww.nict.go.jp/wn-ja/index.en.html

are used to obtain syntactic similarity. That is, in Equation (3.5), t refers to a target word or a common word. For example, there are two common words 聞か (*kika*; hear) and 話 (*hanashi*; story) between **E1** and **I1**. A similarity between "滅多に (*mettani*; rarely) - adverbial - 聞か (*kika*; hear)" in **I1** and "もう (*mo*; anymore) - adverbial - 聞か (*kika*; hear)" in **E1** is also added to the score $syn(I1, E1)$. Considering common words to calculate $syn(I, E)$ will naturally affect in an increased recall, but may or may not affect the precision.

RTW Hereafter, we call the example based WSD classifier which considers syntactic relations only with respect to target word as RTW.

RCW The classifier considering relations with respect to common words as RCW.

Note that both RTW and RCW also use collocation similarity $coll(I, E)$. We will empirically compare precision and recall of RTW and RCW in Chapter 5.

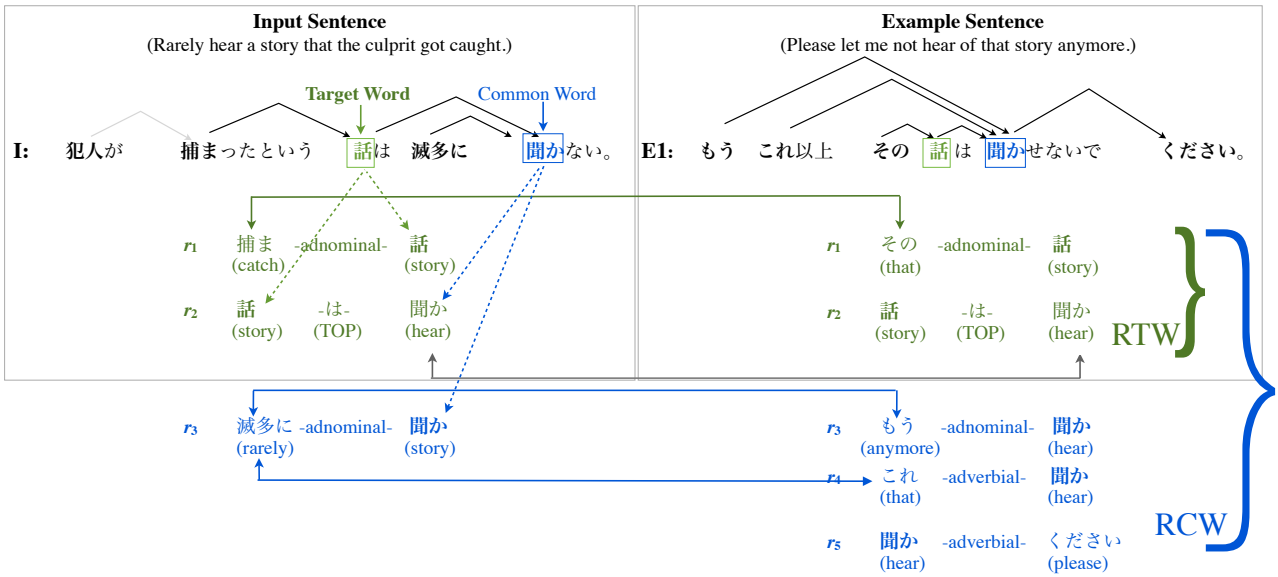


Figure 3.4: Difference between RTW and RCW

Before moving ahead, we illustrate again the difference between these two example based classifiers with the help of an example shown in the Figure 3.4. The target word in the input sentence **I** is 話 (*hanashi*; story). In classifier **RTW**, syntactic relations (r_1 & r_2) are extracted only if they include the target word 話 (*hanashi*; story) as modifier or modifiee. Therefore, similarity among these syntactic relations from input and example sentence, is calculated in the classifier RTW.

While in **RCW**, not only the syntactic relations for the target word but also the common words between two sentences are extracted. For example, there is one common word 聞か (*kika*; hear) between **I** and **E1** with its syntactic relations in the Figure 3.4. Hence, all the syntactic relations that can be extracted (r_3, r_4 and r_5) between two sentences using common words are used in RCW. Note that RTW is a subset of RCW as the target word is always one common word between these two sentences.

3.4 Robinson Classifier (ROB)

As we implied earlier that the proposed example based classifiers focus on achieving high precision rather than recall. Therefore, we also train a robust machine learning classifier, this way both classifiers compensate each other upon combination. For the machine learning, we incorporate a statistical approach of Bayesian classifier proposed by Robinson [19], popularly used in spam detection. In previously reported results by Blosser et al. [1], Robinson classifier showed good performance on binary classification and for the first time we implement this classifier on a WSD task with multiple categories. We used the publicly made available tool[6] , implementation of which is shown in next subsection.

3.4.1 Method

For each sense s, the score S is calculated as Equation (3.7), then the sense which has the highest score is chosen. In Equation (3.8) and (3.9), P and Q estimate the likelihood and unlikelihood of a sense, respectively.

[6]https://github.com/kevincobain2000/Bayes

$$\mathrm{S} = \frac{1 + (\mathrm{P} + \mathrm{Q})/(\mathrm{P} - \mathrm{Q})}{2} \tag{3.7}$$

$$\mathrm{P} = 1 - ((1 - p(f_1)) \times (1 - p(f_2)) \times ... \times (1 - p(f_n)))^{\frac{1}{n}} \tag{3.8}$$

$$\mathrm{Q} = 1 - (p(f_1) \times p(f_2) \times ... \times p(f_n))^{\frac{1}{n}} \tag{3.9}$$

$p(f_i)$ stands for the probability of each feature defined as:

$$p(f_i) = \frac{b(f_i)}{b(f_i) + g(f_i)} \qquad i \in \{1, 2, \ldots, n\} \tag{3.10}$$

where $b(f_i)$ and $g(f_i)$ are the posterior probability $P(f_i|s)$ and $P(f_i|\bar{s})$ estimated by Maximum Likelihood estimation, respectively. $b(f_i)$ and $g(f_i)$ are calculated as

$$b(f_i) = \frac{\text{num of times } f_i\text{occurs in this sense}}{\text{total num of features in other senses}} \tag{3.11}$$

$$g(f_i) = \frac{\text{num of times } f_i\text{occurs in other senses}}{\text{total num of features in this sense}} \tag{3.12}$$

3.4.2 Features

To train the Robinson classifier we use the example sentences in EDICT as the training data. We use the conventional features constituting:

- Collocation of bi-grams and tri-grams including the target word.
- Bag-of-words of content words from each sentence.

A feature set is extracted from the example sentences in EDICT. When no feature from an input sentence occurs, the most frequent sense in example database is chosen. Here, most frequent sense is the sense which has highest number of example sentences.

3.5 Combined Model

The final WSD classifier is an ensemble of example based and Robinson classifiers (ROB). First, precision oriented example based classifier is applied. When it does not choose a sense ($sim(I, E)$ is zero or less than a threshold), a sense from ROB is chosen. The main reason to combine example based classifier with Robinson classifier is to compensate recall because ROB is more robust. Note that the combined model can always choose a sense for given sentences. This is a typical method of combination of classifiers. Our two implementations of example based classifier RTW & RCW results in two combined models i.e.,

1. RTW+ROB
2. RCW+ROB

Any machine learning algorithms can be combined with the proposed example based classifiers. In Section 5.6, Robinson and other machine learning based classifiers are empirically compared in terms of performance of both single and combined models.

3.6 Expanding knowledge for classifiers

Although, we have both Japanese and English sentences, but until now we have been using only Japanese sentences to disambiguate the senses. Example based classifiers being precision oriented we apprehend that low recall as a serious problem, which is due to less knowledge available for classifiers. We do extract more knowledge from the parallel and monolingual corpora that we will present later in this thesis in Chapter 4. But one feasible way to use more knowledge from existing resources. Currently the WSD classifiers proposed in previous sections only use features from the Japanese example sentence. Apart from just Japanese we also incorporate features from example sentences in English as well.

In order to incorporate such features, the input Japanese test sentence is translated using EDICT and EDR Japanese-English bilingual dictionary respectively into English tokens. Then we calculate a score by measuring overlap of words. The details and results following this approach comes next.

3.6.1 Using English Tokens as Features

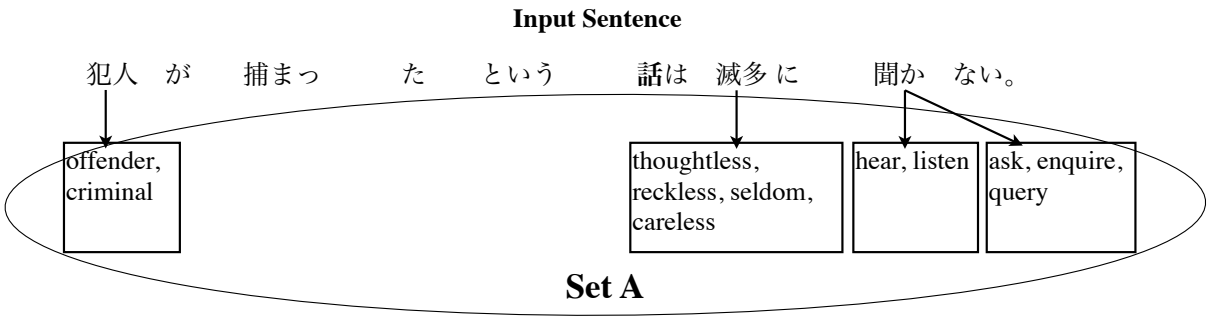


Figure 3.5: Extracting English Features from Japanese Sentence

First we prepare a **Set A** which is obtained by translating each Japanese content word from the input sentence into its English definitions. Here english definitions are obtained from EDICT dictionary. When EDICT doesn't have an entry of the word, EDR is looked up. Illustration of preparing Set A is shown in the Figure 3.5. For example, entry of the Japanese word 捕まっ (*tsukama*; catch) is neither in EDICT nor in EDR hence no english tokens for such words are appended to Set A. Functional words such as が (*ga*) and た (*ta*) are also ignored. Note that definitions from all senses are appended to Set A, for example in case of 聞か (*kika*; hear) two sense definitions (hear, listen) are appended.

To post-process we only remove the stop-words from Set A. Because EDR is much enricher dictionary than EDICT, many translated tokens belong to EDR. Upon which, one may say that its better to look up an entry in EDR first, rather than the other. However, EDICT definition sentences are much more natural and contains more information, while definition sentences in EDR are usually one word translation.

Next, we make successive **Set B** from tokens (except for stop-words) from the example sentences in English for each sense given in EDICT and example sentences we acquire acquire automatically from parallel corpora. Then for each sense we calculate a $score$ and choose the sense which has the highest score as shown in the equation below.

$$score = \frac{|A \cap B|}{|A|} \tag{3.13}$$

Where $|A \cap B|$ is the count of words that overlap in two sets. Hereon, we refer to this approach as Modified Dice. We present the results using this approach later in the evaluation chapter, Subsection 5.7.2.

Chapter 4

Expansion of Example Sentences

In this chapter, we describe the extraction of reliable example sentences from parallel corpora. A method acquiring training example by bootstrapping is also presented.

4.1 Corpora

Since our reading assistant system will show examples in both Japanese and English, the goal here is to extract pairs of Japanese and English sentences. We used following two sentence aligned parallel corpora:

1. JENAAD Constitutes of 150,000 Japanese-English sentence pairs [22]. Sentences from this corpus come from newspaper articles.

2. Wikipedia-Corpus 432,005 parallel sentences [16]. Unlike JENAAD the domain of this corpora is heterogeneous. The articles in this corpora are divided into 15 categories such as school, railway, family, literature etc. More details on the categories can be found on the website[1].

For preprocessing we used BerkleyAligner[2] and Morpha [13] to produce word alignments and lemmatized forms of words.

[1] http://alaginrc.nict.go.jp/WikiCorpus/index_E.html#category
[2] http://code.google.com/p/berkeleyaligner/

4.2 WSD Heuristics to Acquire Parallel Sentences

For each sense of the target word (TW), pairs of example sentences are extracted if they fulfill the following three requirements:

1. There must exist an English word t_e aligned with TW. t_e or a compound word including t_e should match against one of the words or compound words in the sense definition.

   E.g., for target word "守る" (*mamoru*; keep) with sense S_2: {to keep (i.e. a promise), to abide (by the rules)}.

   If t_e is "keep" or "abide" as shown in Figure 4.1, a sentence pair is extracted for the sense S_2. In cases of verbs, we omit checking the preceding non content words such as "to".

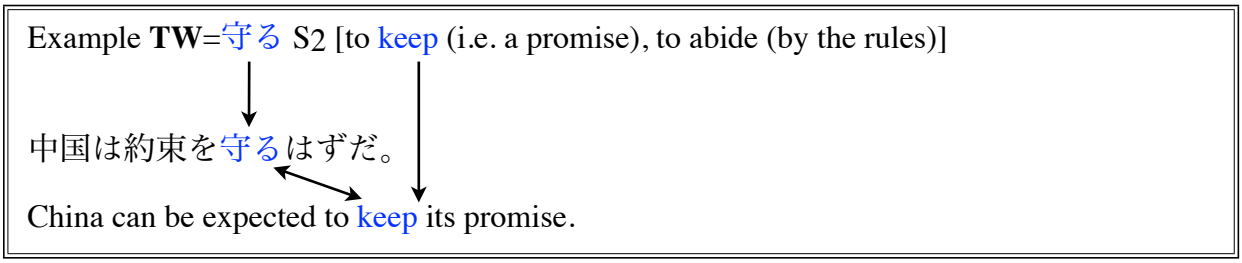


Figure 4.1: Heuristic 1, Example Sentences Expansion

2. When a definition also consists of a short description in parenthesis, one of the content words in parenthesis must be contained in an English sentence.

   E.g., let us consider the target word "守る" (*mamoru*; keep) and its sense S_2: {to keep (i.e. a promise), to abide (by the rules)}.

   Sentence pair is extracted if t_e is "keep" and "promise" appears in an English sentence. As shown in Figure 4.2.

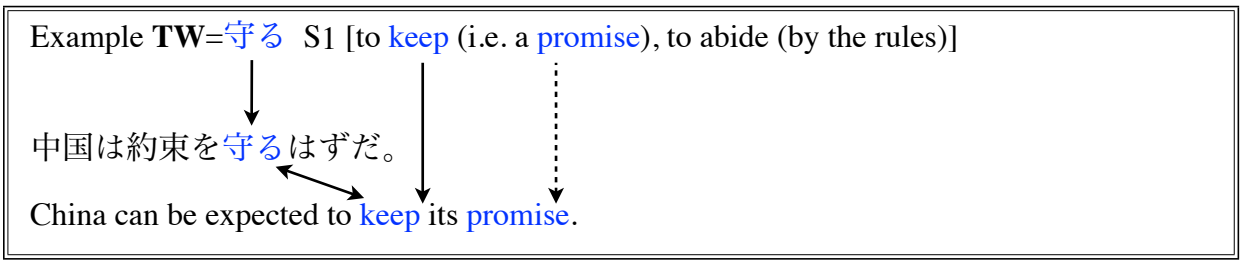


Figure 4.2: Heuristic 2, Example Sentences Expansion

3. t_e should match against a word in sense definition for only one sense.

   E.g., for target word "作る" (*tsukuru*; prepare) with sense S_1: {to prepare, to brew} and S_2: {to prepare, to make out, to write}

   In case of Figure 4.3, sentences are not extracted if t_e is "prepare", since the sense of the target word is ambiguous.

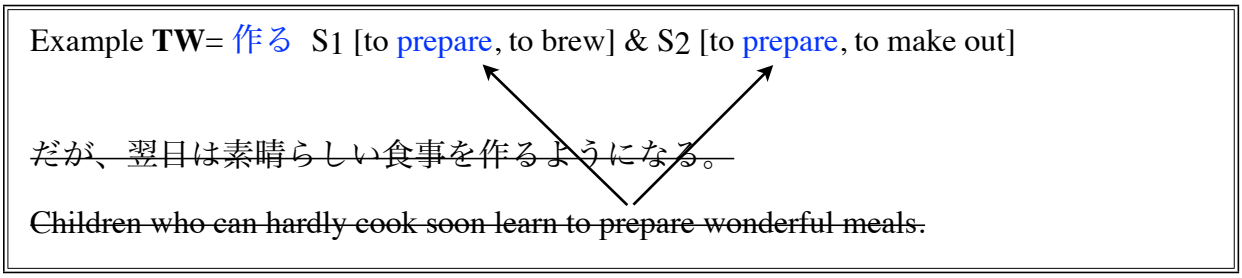


Figure 4.3: Heuristic 3, Example Sentences Expansion

These three constraints are likely to reject many good candidates, but it is crucial to extract sentences with high accuracy. Although pairs of Japanese and English sentences are added to example database, we only use Japanese examples for both example based and Robinson classifiers. The accuracy, coverage and effects on WSD upon expansion are discussed in Chapter 5.

4.3 Bootstrapping Examples

The method to automatically acquire example sentences from parallel corpora has been presented. However, the size of parallel corpora tends to be limited. To increase the number of examples without parallel corpora, monolingual corpora can be used to acquire example sentences in bootstrapping manner. That is, the initial classifier (RTW) using only EDICT as an example database is applied to the sentences in monolingual corpus. The sentences where the sense of the target word is determined with high reliability are extracted and they are added to the example database.

We used, JEITA Public Morphologically Tagged Corpus (in ChaSen[3] format), a public, automatically tagged (morphologically analyzed) corpus of Project Sugita Genpaku[4] and Aozora Bunko[5]. In order to add more example sentences to the senses in EDICT, first we find all the instances (as candidates) of the target word from corpora. Next we used the existing examples and the classifier RTW to decided the sense of the candidate sentence. When RTW is able to disambiguate the sentence we add it as example to the inventory using RTW with Threshold T=0. We discuss the WSD results on the proposed classifier using bootstrapped examples, in next chapter under Subsection 5.7.3.

[3]http://chasen.aist-nara.ac.jp

[4]http://wow.genpaku.org/

[5]http://wow.aozora.gr.jp/

Chapter 5

Evaluation

In this chapter, we present the data prepared for evaluation, results on our automatically acquired examples and results on the proposed WSD classifier.

5.1 Data

In this section we start with presenting the data used for WSD experiments and evaluation.

Table 5.1: Development and Evaluation Data

Data	**# of TWs**	# of test instances	Avg. Sense per TW
Development	**17**	330	3.41
Evaluation	**49**	937	4.65

To evaluate the performance of WSD, we prepared two sense tagged corpora, a development and an evaluation data, as shown in Table 5.1.

Development data as (D_d) is the sense tagged data used to design our example based WSD method and optimize a threshold (T). It consists of 330 input sentences of 17 target words (8 nouns, 8 verbs and 1 adjective).

Evaluation data as (D_e) is another sense tagged data and is built to measure performance of our proposed method. It consists of 937 input sentences of 49 target words (23 nouns, 24 verbs and 2 adjectives).

Table 5.2: List of Target Words from Development and Evaluation Data

S.No.	Development & Evaluation Data	S.No.	Evaluation Data	S.No.	Evaluation Data
1.	中	18.	使う	34.	娘
2.	人	19.	話す	35.	迎える
3.	入る	20.	聞く	36.	置く
4.	出す	21.	高い	37.	立つ
5.	出る	22.	話	38.	起きる
6.	前	23.	作る	39.	相手
7.	地方	24.	地方	40.	切る
8.	得る	25.	言葉	41.	帰る
9.	情報	26.	目	42.	頭
10.	持つ	27.	気	43.	合わせる
11.	日	28.	記録	44.	占める
12.	時	29.	調べ	45.	守る
13.	時間	30.	手	46.	力
14.	自分	31.	間	47.	体
15.	言う	32.	取る	48.	乗る
16.	認める	33.	家	49.	上
17.	開く	34.	娘		

17 target words on D_d are also target words on D_e. Table 5.2 shows the list of selected target words. In both data, input sentences were excerpted from Mainichi Shimbun 1994 articles. Test sentences in D_d and D_e are mutually exclusive even for common target words. The correct senses are manually tagged.

5.2 Results on Expansion

In this section, statistics on the example sentences in the sense inventory after the automatic acquisition of example sentences and accuracy of the expansion are presented.

5.2.1 Statistics on Expansion of Example Sentences

Table 5.3: Comparison of Statistics Before and After Expansion (E+)

	# of TWs	Avg. Sense per TW	Total # of Eg Sents		Avg. Eg Sent per Sense		# of Senses with no Eg Sents	
				E+		E+		E+
T_d	17	3.41	4,252	16,988	73.31	292.9	10	7
T_e	49	4.65	10,998	32,748	48.23	143.6	70	57

Table 5.4: Number of Expanded Example Sentences per Corpus

	E+
JENAAD	16,468
Wiki-Corpus	16,280
Total	32,748

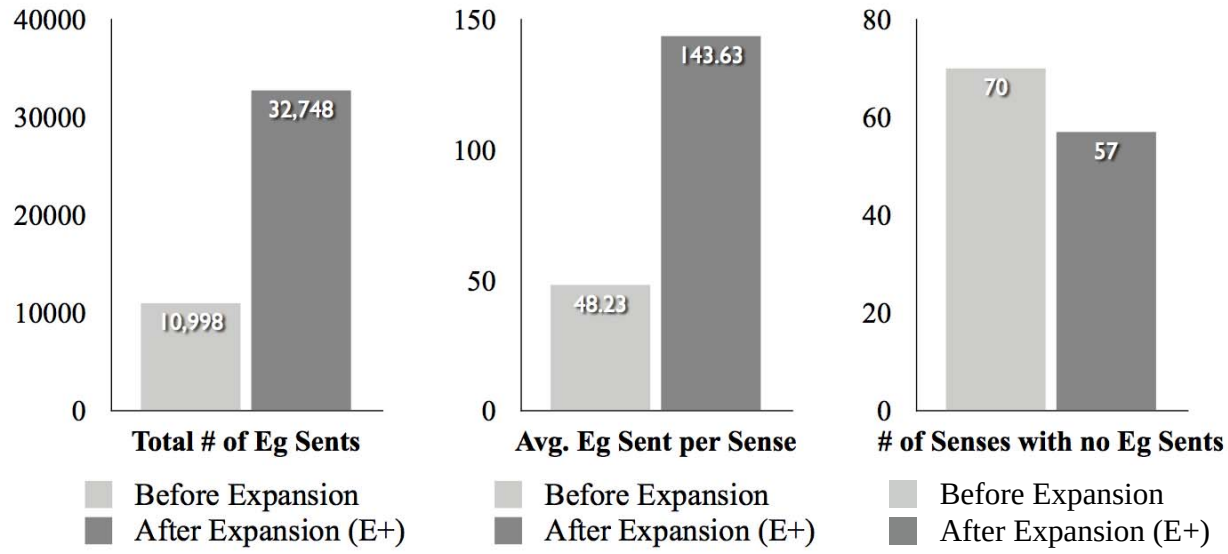


Figure 5.1: Statistics after Examples Expansion

Table 5.3 and Figure 5.1 show the statistics before and after expansion of example sentences, where T_d and T_e are sets of target words (TWs) on D_d and D_e, respectively. Statistics below the label E+ represents the figures after expansion by the method shown in Section 4.2 original numbers from EDICT. Number of example sentences are increased by about 3 times by expansion. Automatically expanded examples covered 53% of senses for T_e. Furthermore, number of senses with no example (6th column) are decreased. Note that senses with no example are crucial for our WSD method since such a sense is never chosen. No example sentences can be shown in the Reading Assistant System either.

Table 5.4 reports the number example sentences expanded per corpus. The Wikipedia-Corpus which consists of 432,005 parallel sentences in total, and around 3 times more than the size of JENAAD (150,000 sentences). It is interesting to note that, the number of expanded examples from Wikipedia-Corpus are less than JENAAD. The reason behind this could be the quality of parallel sentences in the respective corpora. Both corpora are automatically sentence aligned and morphologically parsed. The constraints designed for WSD heuristics to extract examples, which we presented in Section 4.2, effectively prunes false candidates. Results indicate that majority of false candidates may belong to the Wikipedia-Corpus. Nevertheless, our goal is to extract the sentences only in reliable cases.

5.2.2 Accuracy of Expansion

Table 5.5: Results on Examples Expansion

# of Sents	Correct	Incorrect	Accuracy
652	553	99	85%

Among 21,750 expanded sentences for 49 target words, we randomly chose 10 sentences at most for each sense, then manually evaluate if extracted sentences are correct or not. Evaluated sentences are the ones from JENAAD. Accuracy of automatically expanded example sentences was 85% as shown in Table 5.5. The relatively high accuracy indicated that constraints on checking information in both languages effectively prunes false candidates generated due to misalignments or errors on morphological analysis.

5.2.3 Error Analysis on Examples Expansion

Causes of errors on example expansion are investigated. Among incorrect, 5 instances were due to wrong morphological analysis on Japanese while wrong sense is chosen in 94 instances. We found that one English word could correspond to two or more senses of target words in most cases where wrong sense is chosen. Some examples are shown below:

- S_1: {inside, in} is one of senses of noun 中 (*naka*).

  S_1 means that a word is used with the name of a container or place to say where something is, such as 机の**中** (*Tsukue no naka*; in the desk). However, senses other than S_1 are often translated as "in", such as 自由の**中** (*Jiyū no naka*; in freedom).

- Another example is the noun 人 (*hito*).

  Two senses of this target word are S_1: {man, person} and S_2: {mankind, people}. If the target word of S_1 is translated as plural form of "person", i.e. "people", sentences are wrongly extracted for the sense S_2.

It is rather difficult to distinguish senses based on sense definitions in such cases, which tend to happen if differences among senses are subtle.

5.3 WSD Results on Development Data

Table 5.6: Results on Development Data D_d

	RTW			RCW			RTW +ROB	RCW +ROB
T	**P**	**R**	**F**	**P**	**R**	**F**	**A**	**A**
0.0	0.72	0.48	0.58	0.66	0.53	0.59	0.67	0.66
0.3	0.76	0.35	0.48	0.68	0.45	0.54	0.67	0.66
0.6	0.83	0.26	0.39	0.73	0.36	0.48	0.66	0.66
0.9	0.90	0.16	0.28	0.79	0.29	0.42	0.64	0.66
E+								
0.0	0.70	0.57	0.63	0.62	0.59	0.60	0.68	0.61
0.3	0.71	0.55	0.62	0.63	0.55	0.58	0.67	0.61
0.6	0.74	0.45	0.56	0.67	0.49	0.56	0.65	0.60
0.9	0.77	0.36	0.49	0.68	0.40	0.51	0.62	0.59

	ROB
	A
	0.62
E+	0.60

	BL
	A
	0.62
E+	0.60

Table 5.6 reveals that the precision (P), recall (R) and F-measure (F)[1] of two example based classifiers RTW and RCW as well as accuracy (A) of ROB, baseline BL and two combined models RTW+ROB and RCW+ROB on the development data. Baseline (BL) here is the system which always selects the sense which has the highest number of example sentences. If more than one senses have same number of example sentences, it randomly chooses a sense. This is typically the baseline model when using only example sentences for WSD. Since ROB, BL, RTW+ROB and RCW+ROB always choose a sense, not precision and recall but accuracy (A) (ratio of agreement between gold sense and predicted sense) is shown for these systems.

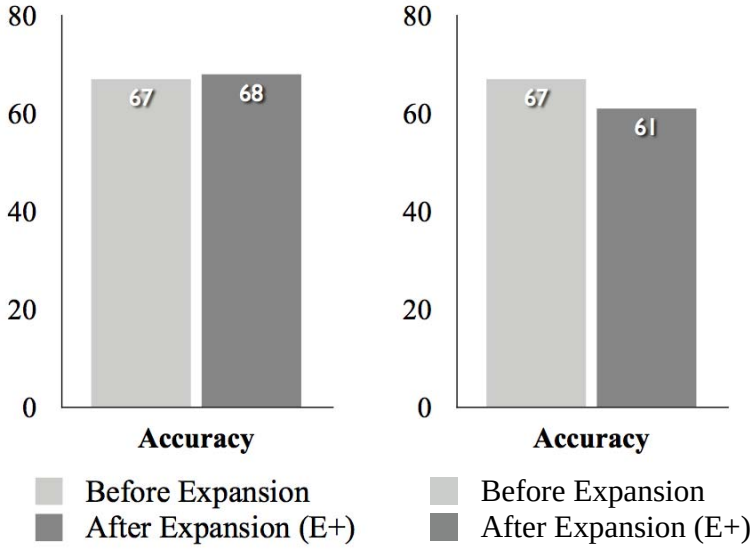
[1] $F=\frac{P+R}{PR}$

As expected, when the threshold (T) is set high (in Table 5.6 and also shown in Figure 5.3), precision of RTW and RCW increases but recall is dropped. Combination of precision oriented example based method with a robust ROB classifier is effective to improve the performance of WSD, since the accuracy of combined model outperforms both F-measure of RTW (or RCW) and accuracy of ROB.

Comparing RTW and RCW, RTW is better than RCW for precision and vice verse for recall and F-measure. For example, recall of RTW at T=0.3 and RCW at T=0.6 is around 0.35, while precision of RTW and RCW are 0.76 and 0.73, respectively. When they are combined with Robinson classifier, RTW+ROB is better than RCW+ROB. This is because more precision oriented classifier RTW is preferable for combination with Robinson classifier. However it is ok that RTW+ROB is better than RCW+ROB because the development data only consists of 330 test sentences from 17 target words. Also prior to the expansion both RTW and RCW's accuracies are comparable. Appropriately, it would be better to justify which classifier is better from the results on the evaluation data. Before we present the WSD results on the evaluation data in Section 5.4, we first discuss results on the development data in the following subsections.

5.3.1 Effectiveness of Example Sentences Expansion

By expanding example sentences, recall and F-measure are improved for both RTW and RCW, while precision is comparable. Example sentence expansion seems not contribute to a gain in the precision, although sentences are expanded with a high accuracy (85%). But it is not sure whether expansion has a positive impact on precision because the development data only consists of 17 target words.



(a) RTW+ROB (b) RCW+ROB

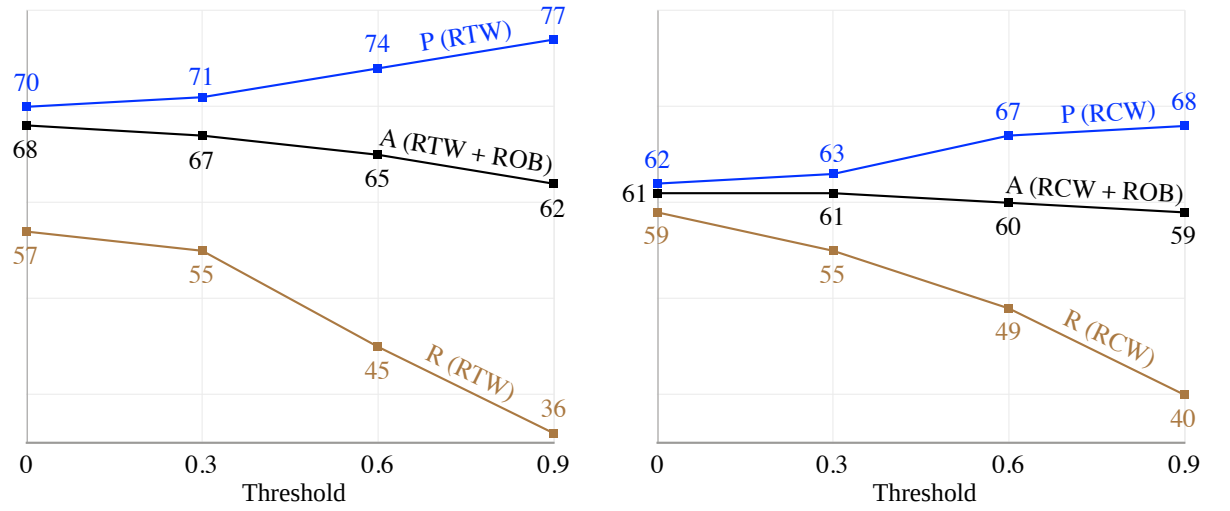
Figure 5.2: Accuracy of RTW(RCW)+ROB before and after Examples Expansion (E+), Development Data

Figure 5.2 clearly shows the performance of the combined model before and after expansion. RTW+ROB is comparable after expansion, while RCW+ROB is worse. Regardless of recall improvements, there is a drop in precision of RTW and RCW at same thresholds caused by expansion, which negatively influences the performance of combined models, especially RCW+ROB.

5.3.2 Robinson Classifier

The performance of ROB is not so improved from BL on both with and without expanded examples. One of the reasons may be that example sentences in EDICT are used as training data. Especially, distribution of appearance of senses, which is known as effective statistics for WSD, can be trained from a sense tagged corpus, but not from example sentences in the dictionary, since it is not guaranteed that the numbers of examples for senses follow the real distribution. Further it makes more difficult to obtain sense distribution from automatically extracted sentences from the parallel corpus, since not all but only reliable sentences are extracted.

5.3.3 Threshold Optimization



(a) Results on RTW and RTW+ROB on Development Data with Expanded Example Sentences

(b) Results on RCW and RCW+ROB on Development Data with Expanded Example Sentences

Figure 5.3: Threshold Optimization on Combined Models RTW (RCW) + ROB

Figure 5.3 shows the effect of increasing threshold on precision (P), recall (R) and accuracy (A) of the classifiers RTW(RCW) and its combination with ROB. Considering optimization of the threshold, T=0 seems the best parameter for both RTW+ROB and RCW+ROB, although the accuracy does not highly depend on T in the combined model. We set this threshold on the development and evaluate the performance of the system at T=0 on the evaluation data. Results on the evaluation data are shown in the next section.

5.4 WSD Results on Evaluation Data

Table 5.7 shows results on the evaluation data D_e on two example based classifiers RTW, RCW and respective combinations with the Robinson Classifier RTW (RCW) + ROB. Results of ROB and BL are also shown for comparison. Numbers under the label E+ shows the results after the injection of automatically acquired examples from parallel corpora.

Table 5.7: Results on Evaluation Data D_e

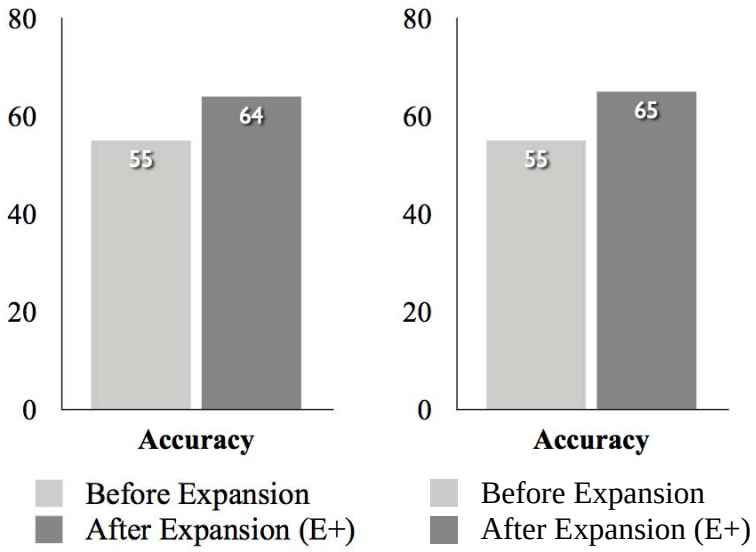
	RTW			RCW			RTW +ROB	RCW +ROB
T	**P**	**R**	**F**	**P**	**R**	**F**	**A**	**A**
0.0	0.63	0.43	0.51	0.60	0.48	0.54	0.55	0.55
0.3	0.65	0.31	0.43	0.64	0.42	0.51	0.53	0.54
0.6	0.73	0.21	0.33	0.67	0.33	0.44	0.53	0.53
0.9	0.80	0.12	0.21	0.71	0.24	0.36	0.53	0.53
E+								
0.0	0.66	0.55	0.60	0.65	0.65	0.65	0.64	0.65
0.3	0.67	0.52	0.58	0.66	0.59	0.63	0.63	0.66
0.6	0.68	0.45	0.54	0.68	0.50	0.50	0.64	0.66
0.9	0.70	0.33	0.45	0.71	0.40	0.40	0.64	0.66

	ROB
	A
	0.51
E+	0.60

	BL
	A
	0.51
E+	0.58

From Table 5.7 and also from Figure 5.5, as expected upon constraining the threshold (T) the precision (P) increases while recall (R) drops. From the accuracies in the table at T=0, both RTW+ROB (0.64) and RCW+ROB (0.65) are comparable. Threshold T=0, optimized from the development data, seems appropriate for example based classifiers RTW and RCW as it has the highest F-measure, although it has minimal effect on the combined models. If compare results from the baseline system (0.58), the proposed system's (RCW+ROB) accuracy (0.65) is 7% greater. Accuracies of both are comparable, but we prefer RTW+ROB over RCW+ROB. Next in the following subsections we talk further on the results in contrast with the expanded example sentences, sentence similarity measures and threshold.

5.4.1 Effectiveness of Example Sentences Expansion



(a) RTW+ROB (b) RCW+ROB

Figure 5.4: Effect of Examples Expansion (E+) on WSD, Evaluation Data

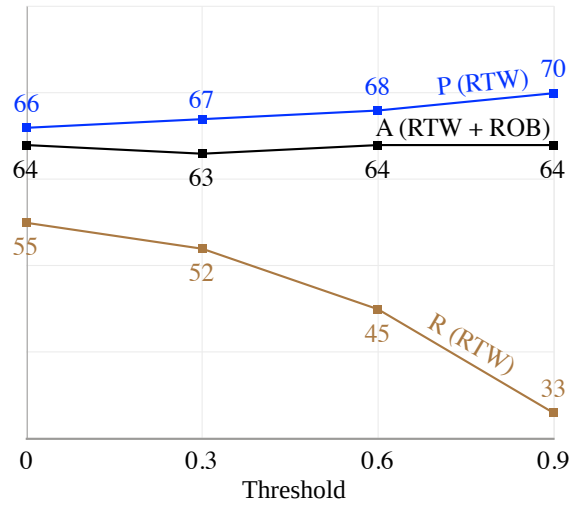
Figure 5.4 clearly compares the results of RTW + ROB and RCW + ROB before and after example expansion. From Figure 5.4 and Table 5.7 unlike results on D_d, the expansion of example database gives remarkable impacts for all classifiers on D_e. Especially, not only recall and F-measure but also precision of RTW and RCW is improved by expansion. Since D_e consists of more target words and test instances than D_d, results on D_e might be more reliable than D_d. Therefore we can say that the example sentences expansion is effective for WSD.

Nevertheless, if more corpora of Japanese-English parallel sentences the accuracy is expected to increase, however the domain of the corpora plays a very important role in the performance. Currently we have extracted sentences from a News domain and Wikipedia editorial articles. Incorporating more data from different domains naturally will increase the recall but may or may not effect the precision. Because our target words are only 49 which might not be enough to tackle the domain adaptation problem and is out of scope of this research. We discuss briefly about domain effect on proposed WSD in Subsection 5.7.5.

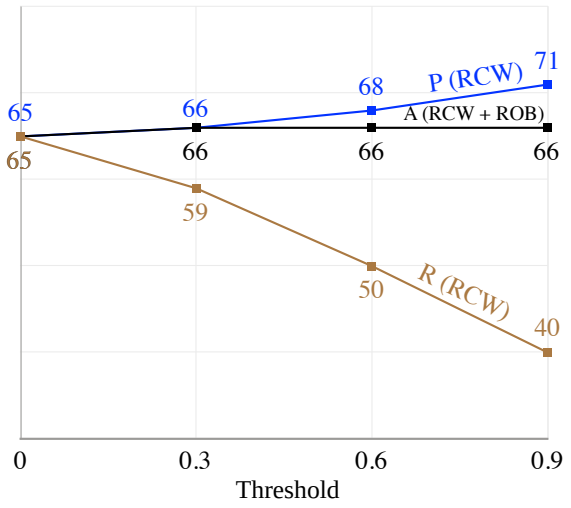
5.4.2 Effectiveness of Syntactic and Collocation Similarities

In previous researches on Japanese example based WSD, Fujii et al. and Shirai et al. calculate a syntactic similarity between two sentences by exploiting a case frame dictionary and Japanese language dependency structure respectively [21][8]. Our proposed example based WSD classifiers are inspired by these two researches, where syntactic similarity $syn(I, E)$ measured between two sentences seems to be precision oriented. Both previous researches classifiers face a low recall problem, therefore on top of syntactic similarity, we further calculate a collocation similarity between two sentences. In order to evaluate the contribution of collocation feature alone, we implemented RTW without collocation score $coll(I, E)$ on the evaluation data. Comparing RTW in Table 5.7 (T=0, with expanded example sentences), its precision was the same but recall was 4% lower.

5.4.3 Threshold Validation



(a) Results on RTW and RTW+ROB on Evaluation Data with Expanded Eg Sentences



(b) Results on RCW and RCW+ROB on Evaluation Data with Expanded Eg Sentences

Figure 5.5: Threshold Validation on Combined Models RTW (RCW) + ROB

Next we validate the optimized Threshold T=0 which we obtained from the development data from the most accurate classifier RCW+ROB. Figure 5.5 shows the effect of threshold on the

evaluation data. As expected on increasing the threshold the precision increases while the recall decreases. Moreover, T=0 seems appropriate as it has the highest accuracy on RTW+ROB, but not RCW+ROB.

5.5 Comparing WSD Results on Development and Evaluation Data

5.5.1 Before Examples Expansion

From Table 5.6 (WSD results on development D_d) and Table 5.7 (WSD results on evaluation data D_e), the performance of all classifiers on D_e is worse than that on D_d. Some of the possible reasons is that WSD of target words in D_e might be more difficult, because

1. There are more senses per target word

   4.7 for D_e, while 3.2 for D_d, as shown in Table 5.3.

2. Less example sentences per sense

   48.23 for D_e, while 73.31 for D_d, as shown in Table 5.3

3. The precision of ROB and BL is worse.

   0.51 on D_e, while 0.60 on D_d as shown in Tables 5.6 and 5.7.

5.5.2 After Examples Expansion

After the examples expansion the difference in the accuracies on both data is not huge. Although results on the evaluation data are short by 4% on RTW+ROB. But this is natural as evaluation data consists of more target words and less example sentences as compared to the development data, as we explained in the previous subsection. On the same hand, accuracy of RCW+ROB has increased by 4% from development to evaluation data. Nevertheless, expansion of example sentences proves to be efficient to bring up the accuracy significantly.

5.6 Comparison of Machine Learning Algorithms

Table 5.8: Results on Seven Classifiers, Development Data

Classifiers	F-Measure		Accuracy	
			RTW + X	
		E+		**E+**
Decision Tree	0.57	0.60	0.67	0.68
Maximum Entropy	0.57	0.64	0.62	0.68
Naive Bayes	0.54	0.45	0.65	0.68
SVM	0.65	0.46	0.66	0.68
Baseline	0.59	0.61	0.66	0.68
ROB	0.62	0.61	0.67	0.68

In order to check the performance of popular machine learning algorithms, we trained the following classifiers from the same training data (i.e. example sentences). Table 5.8 summarizes the results on the development data. Without automatic expansion, Support Vector Machine (SVM) outperformed Robinson (ROB) by 3%. When both example sentences in EDICT and automatically acquired examples are used Maximum Entropy is the best. When they are combined with RTW, however the accuracies are comparable.

RTW+X represents the classifier combined with RTW in order. Because RTW+ROB has the highest accuracy 0.67 before expansion and equal to others 0.68 after expansion, we choose it as the best combined classifier. By looking at the results from Decision Tree (DT) and ROB, RTW+DT and RTW+ROB has same accuracies, 0.67 & 0.68 (E+). But F-measure of ROB is greater than DT and further more Robinson classifier is much faster than training a Decision Tree. Conclusively, expansion of examples together with the combination of RTW has positive effect on accuracy on all classifiers.

Performance shown in Table 5.8, is much worse than previous work of supervised learning of WSD classifiers. A collection of example sentences in a dictionary or automatically extracted examples seems less appropriate for supervised learning than a sense tagged corpus. One of the reasons is that the frequency of senses cannot be trained.

5.7 Discussion

In this section we discuss about additional experiments that we conducted in order to improve WSD accuracy. Motivation behind these experiments came upon error analysis, previous work and observations on our results from the proposed method.

5.7.1 Combination of Classifiers

Classifier combination has been studied intensively (such as (Brill et al. 1998), (Halteren et al. 1998) and (Pedersen et al. 200)) in the last decade, and has been shown to be successful in improving performance on diverse applications [3, 23, 18]. The intuition behind classifier combination is that individual classifiers have different strengths and perform well on different subtypes of test data. These researches show that combinations of classifiers with different schemas such as average probability classifiers and hierarchical classification combination yields better accuracies.

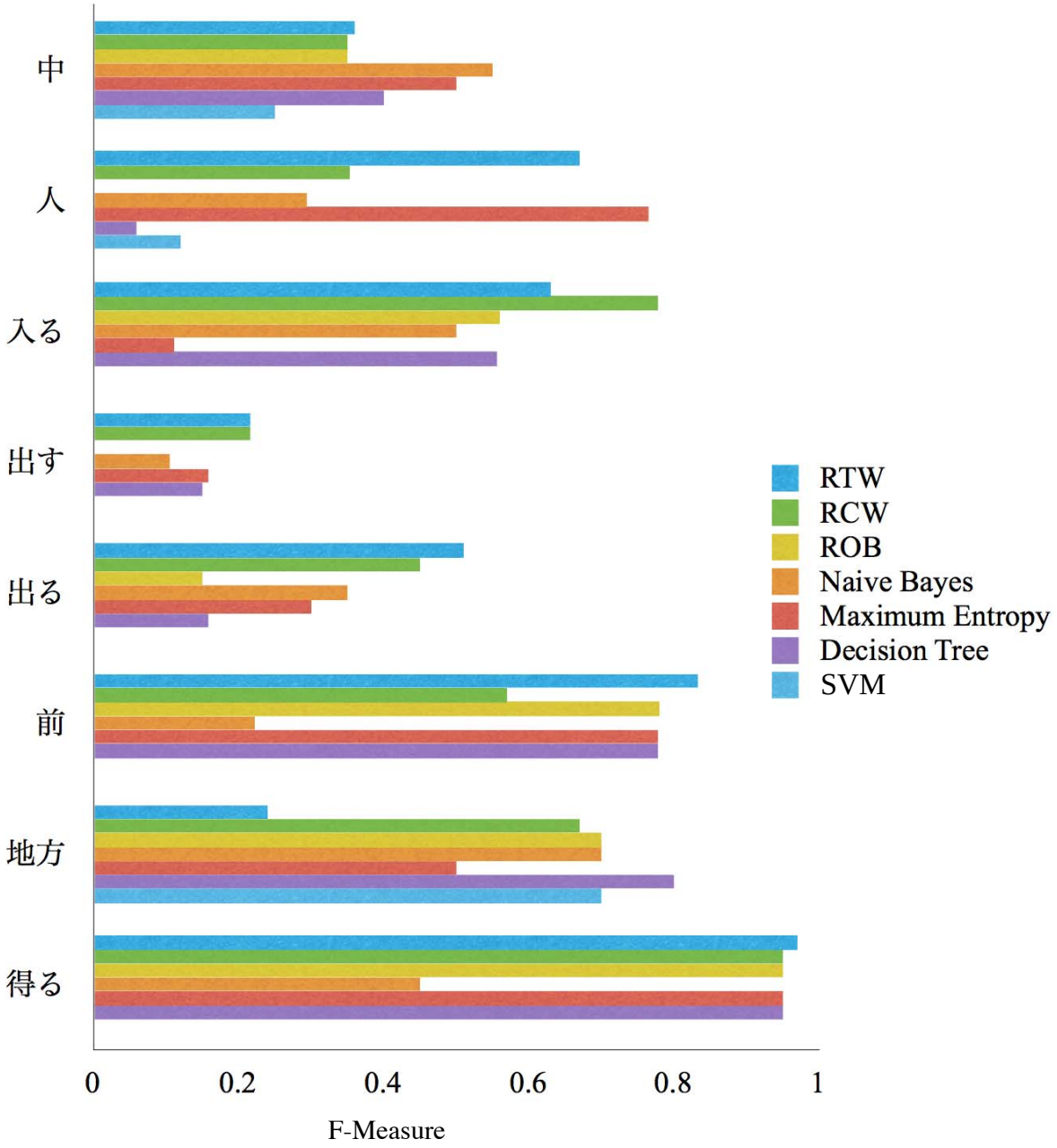


Figure 5.6: F-Measure on each Target Word from different classifiers, Development Data E+-1/2

In the rest of this section, we restrict our analysis on the target words from the development set. As we mentioned earlier that the training is data is not a tagged corpus but are the exam-

ple sentences for senses. Therefore classifier combination approaches like average probability combination might not be effective, because probability distribution is not given. We analyzed the performance of each target word on the development data per classifier.

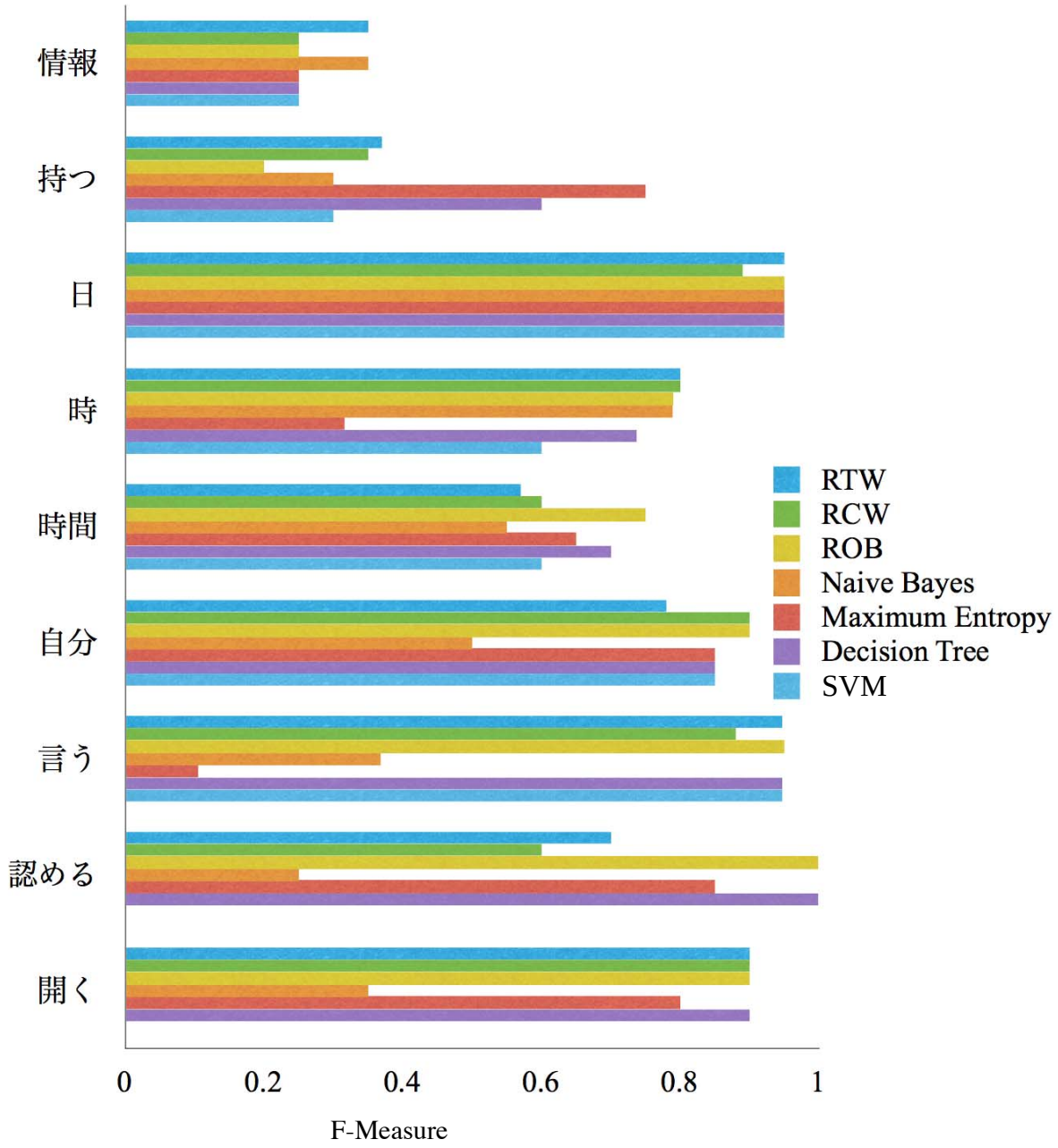


Figure 5.7: F-Measure on each Target Word from different classifiers, Development Data E+ - 2/2

We showed the F-Measure of classifiers used in previous work in Table 5.8 and here we show in more detail the F-measure on each target word divided into Figure 5.6 and 5.7. Based on the bars we put the following notable points:

1. Target Words 中 (*naka*), 出す (*dasu*), 出る (*deru*), 情報 (*jōhō*) have the lowest F-measure and these are the target words which decrease the overall accuracy on the complete set. For these target words, accuracy from all classifiers is less than 0.60, i.e. any combination of classifiers applied will not help to boost the accuracy.

2. Maximum Entropy classifier performs better than RTW for target words 中 (*naka*), 人 (*hito*), 地方 (*chihō*), 持つ (*motsu*), 時間 (*jikan*), 自分 (*jibun*), 認める (*mitomeru*). Alongside performs worse than RTW for rest of the target words. The similar scenario goes if we mutually see the results from the decision tree classifier. It is rather hard to guess which classifier is suitable for which target word based on development data as they may behave otherwise on a new evaluation data with different target words.

Since the best classifiers are different for individual target words, the system that uses different classifiers for each target word may improve the performance of WSD. However, how to choose best one for each target word is still an open question.

5.7.2 WSD Results on Modified Dice

In this subsection, we present the results of the Modified Dice approach presented in Subsection 3.6.1. Briefly recalling, in Modified Dice, we use the example sentences in English and calculate a score for each sense by measuring an overlap of English tokens. The results on the development and evaluation data of the Modified Dice (MD hereafter) and its combined model with example-based classifier RTW and RCW are shown in Table 5.9.

Table 5.9: WSD Results on Modified Dice (MD)

(a) Development Data

	F-Measure
MD	**0.65**
ROB	0.62
RTW	0.63
RCW	0.60
	Accuracy
RTW+MD	0.68
RTW+ROB	**0.68**
RCW+MD	0.63
RCW+ROB	0.61

(b) Evaluation Data

	F-Measure
MD	0.63
ROB	0.64
RTW	0.60
RCW	**0.65**
	Accuracy
RTW+MD	0.64
RTW+ROB	0.64
RCW+MD	0.64
RCW+ROB	**0.65**

Note that Modified Dice (MD) always choose a sense. The results didn't improve but unexpectedly aren't worst either especially on the evaluation data. MD has highest F-measure of 0.65 on the development data, however the accuracies of combined models, RTW+MD and RTW+ROB are same. On the evaluation data, example based classifier RCW has the highest F-measure and further when combined with Robinson (ROB), RCW+ROB has the highest accuracy of 0.65. In fact if we compare this result with the results on the other machine learning classifiers presented in Table 5.8, we could say that it is worth looking more into this approach in future on a different evaluation data.

5.7.3 WSD results on Bootstrapped Examples

Till now we have presented the WSD results on the combined classifiers RTW+ROB, where example sentences are excerpted from parallel corpora. RTW+ROB is faster than RCW+ROB and has comparable accuracies on the evaluation data from Table 5.7.Therefore further in this subsection, we discuss the performance of RTW+ROB, where the example sentences are taken from a monolingual corpus in a bootstrapping manner as we showed in the Section 4.3.

Table 5.10: WSD Results on Bootstrap

(a) Development Data

	F-Measure	
	E+	B+
RTW	0.63	0.65
ROB	0.62	0.66
	Accuracy	
RTW+ROB	0.67	0.71

(b) Evaluation Data

	F-Measure	
	E+	B+
RTW	0.60	0.49
ROB	0.64	0.59
	Accuracy	
RTW+ROB	0.64	0.59

Table 5.10 shows the results of our proposed classifier on both development and evaluation data. F-Measure and accuracy under the label B+ represents the results where only bootstrapped examples are used. In the bootstrapping approach, example sentences come only from monolingual corpora, while in results under the label E+ are the ones where example sentences come from EDICT database and automatically extracted examples from parallel corpora as explained in Section 4.2.

From the results, bootstrapped examples seem to improve accuracy on the development data however performed worse on the evaluation data. On a new note, bootstrapping faces a fundamental problem that sense frequency obtained from the bootstrapped examples is not reliable. Probability based machine learning relies on the frequency of senses/words and with our bootstrapped approach it is not guaranteed that the words follow the real frequency distribution. To overcome this issue instead of building a sense tagged corpus which involves human labor thereby expensive, we focused on automatically acquiring sense frequency, as also studied in the modern researches, such as Celina et al. (2010) used Wikipedia as the sense inventory and tackles a similar problem to automatically measure sense frequency [20].

5.7.4 Automatic Learning of Sense Frequency - Web Search Ranking

It would be appropriate to say that the example sentences are not appropriate for machine learning because sense frequency cannot be estimated. Moreover, when we expand the examples

sentences using parallel corpora, the estimation becomes more difficult because examples only in reliable cases are extracted. The distribution of number of example sentences per sense, shown in Table 5.11, non uniform, especially when the example sentences are expanded.

Lets pick one example from the Table 5.11. The first target word's, 中 (*naka*; in), sense S01 has 2 example sentences prior to the expansion and increased to 257. While S02 had 6 examples prior and increased to 103. Such a non uniform increase in the examples doesn't ensure that it is the real sense distribution. Next we explain about our attempt to acquire the sense frequency from the web search results.

Table 5.11: Statistics of Senses, Example Sentences of Target Words in Development Set

S.No.	Target Word	Num of Example Sentences per Sense (num in parenthesis is after E+)							
		S01	S02	S03	S04	S05	S06	S07	S08
1.	中	2(257)	6(103)	0(5)	0(12)				
2.	人	28(201)	2(770)	0(0)	0(0)	0(0)	0(0)		
3.	入る	4(906)	1(4)	5(69)	2(79)	1(1)			
4.	出す	14(26)	1(18)	2(2)	1(10)	33(47)	6(19)	21(30)	1(4)
5.	出る	232(434)	31(181)	8(8)					
6.	前	320(683)	1(435)	0(15)	3(10)	0(0)	1(1)		
7.	地方	19(1498)	4(19)	0(0)					
8.	得る	253(1302)	17(17)						
9.	情報	1(7)	3(43)						
10.	持つ	5(241)	7(43)						
11.	日	476(476)	62(63)	0(0)					
12.	時	2(820)	46(156)						
13.	時間	293(1009)	244(801)						
14.	自分	1906(1906)	20(90)						
15.	言う	3(2127)	8(16)						
16.	認める	1(343)	1(1)	23(546)					
17.	開く	31(31)	6(7)	95(870)					

Previous researches on Japanese WSD has shown good performances when a sense tagged corpus is provided [15, 17]. One of the possible solutions to automatically obtain a sense tagged corpus for EDICT would be to use an existing sense tagged corpora available for Japanese WSD task. One of the available sense tagged corpora is the EDR corpus annotated with senses defined by EDR Japanese dictionary. If we want to use this for EDICT senses, an alignment of senses in two dictionaries would be required. Upon previous work by Bond et al., tried to align these two dictionaries and found a very few hits of senses [2]. Therefore we shifted our focus away

from dictionary alignment.

Celina et al. (2010) used wikipedia as the sense inventory and tackles a similar problem to automatically measure sense frequency [20]. They estimated the sense frequency based on number of wikipedia articles combined with the number of search queries for the word. Additionally they also used web traffic stats[2] to accompany this estimation.

Based on this observation, we try to estimate the sense frequency using the same. For this experiment we pick the Naive Bayes classifier for testing because the sense frequency obtained from Web search engine can be easily incorporated into the overall probability. We trained two standard Naive Bayes classifiers NB and NB-WEB as the following equation.

$$\hat{s} = \underset{s \in S}{\text{argmax}}\, P(s) \prod_{i=1}^{n} P(f_i|s) \tag{5.1}$$

In NB the probability of the sense $P(s)$ is calculated using the example sentences database, while in NB-WEB we estimate the frequency of a sense from Google, Bing and Wikipedia search queries using a public tool[3]. For each sense, we form search queries from the target word and one content word in its sense definition. For example, 話 (*hanashi*) as two sense definitions, S1: "story, talk, conversation" and S2: "discussions, argument". For S1, we obtain three queries like 1) '話' and 'story' 2) '話' and 'talk' 3) '話' and 'conversation'. Similarly, two search queries are formed for S2. Then we obtain the count of number of web pages obtained from the combined web search engines using these queries. The public tool that we have used, outputs the value of $P(s)$ (between 0-1), for each sense (s). It is estimated as follows:

$$P(s) = \frac{\text{Number of web pages for queries from sense } s}{\text{Total number of web pages for queries from all senses}} \tag{5.2}$$

The current method to infer $P(s)$ by Web search engines still has several questions. One is that it is uncertain how effective the combination of Japanese and English word to disambiguate the sense of the target word in web pages. The other is the bias caused by the number of words in the sense definitions. More words the sense definition of the sense s contains, the greater $P(s)$ is estimated. In both two Naive Bayes approaches, $P(f_i|s)$ is calculated same with features

[2]http://stats.grok.se

[3]http://wow.clips.ua.ac.be/pages/pattern

(f_i) obtained from the example sentences. We show the results on NB and NB-WEB on the development data in Table 5.12.

Table 5.12: F-Measure on NB and NB-WEB on Development Data, E+

	NB	NB-WEB
	F	**F**
	0.54	0.33
E+	0.45	0.40

There is a big drop in the F-measure when $P(s)$ is estimated using web search queries. Although the results have been improved in previous research (Celina et al. (2010)) using this approach. Their published results were from a coarse grained sense inventory while EDICT's sense inventory [20] is fine grained. Moreover, we have searched the web by forming bilingual searches, which seems not appropriate. But we can say that the accuracy greatly vary on the test instances, sense inventory and different origins (i.e. domain) of test and training data.

5.7.5 Domain

The test instances come from News domain while the original example sentences in EDICT are different. Upon expanding example sentences, we used the parallel corpora domain of which is also news articles and Wikipedia. The accuracy of all the classifiers is increased upon injected the example sentences from such domain. It is reasonable to say that the domain of example sentences plays a crucial role in the accuracy.

In order to check the effect of domain, we performed experiments where input and example sentences were chosen specifically for the same domain. We prepared two test data for 17 target words from the development set.

Test 1 For 17 target words, we popped 20 example sentences for each target word from the EDICT's example sentences, on average 4 example sentences for each sense and used it a for testing the accuracy. Rest of the example sentences were used by RTW+ROB classifier. The accuracy of classifier in this test is 94%.

Test 2 As we explained earlier in Section 5.1, 17 target words from development data are also in the evaluation data. In test 2 for RTW+ROB, instead of using EDICT and expanded example sentences we used the sentences that we manually annotated in the development data (20 sentences for each target word) and evaluate the performance of classifiers on 17 target words on test instances from evaluation. We achieved the accuracy of 78% which is 10% higher when same test sentences were classified using EDICT and expanded example sentences. Note that in Test 2, there are only 20 sentences for all senses of each target word in the example database.

Good accuracy on these tests is because the training and test sentences come from same corpus (domain). Many state of the art WSD classifier's published results, are where the test data comes from the same training data. Because our reading assistant system should handle input sentences from any domain, it might be necessary to identify the domain first and then use only the domain specific example sentences. We don't handle such domain adaptation in this research because the test sentences for 49 target words, which we manually annotated, for development and evaluation are not enough.

5.8 Limitations

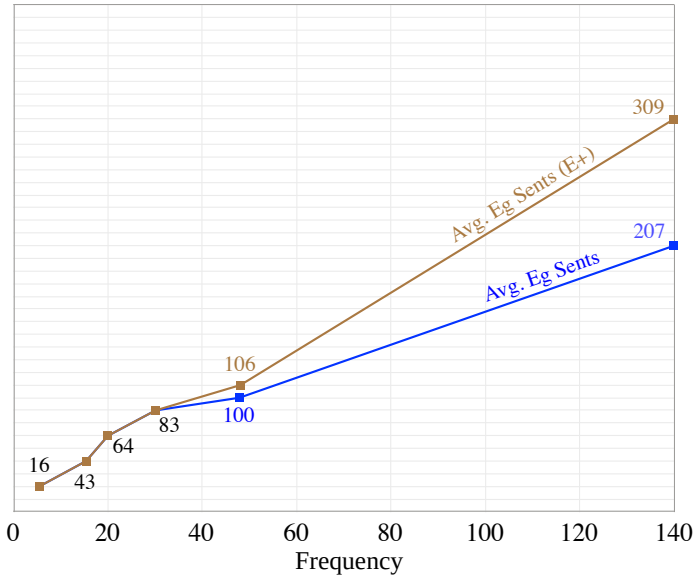


Figure 5.8: Statistics, Example Sentences Expansion and Word Frequency. Stats estimated on 200 words at each frequency

In this section, we present the detailed limitations of our proposed approach. Figure 5.8 shows the statistics of our examples expansion technique towards the frequency of words. Horizontal axis shows the frequency of word usage in the corpus and the average number of example sentences are estimated on 200 words at each frequency block. In this research the target words in the development (17) and evaluation data (49) come from high frequency words. Present expansion method performs well in extracting and increasing the average number of example sentences per sense towards high frequency however low frequency words remains the same. It was out of scope of this research, nevertheless this approach might not be feasible in such scenario.

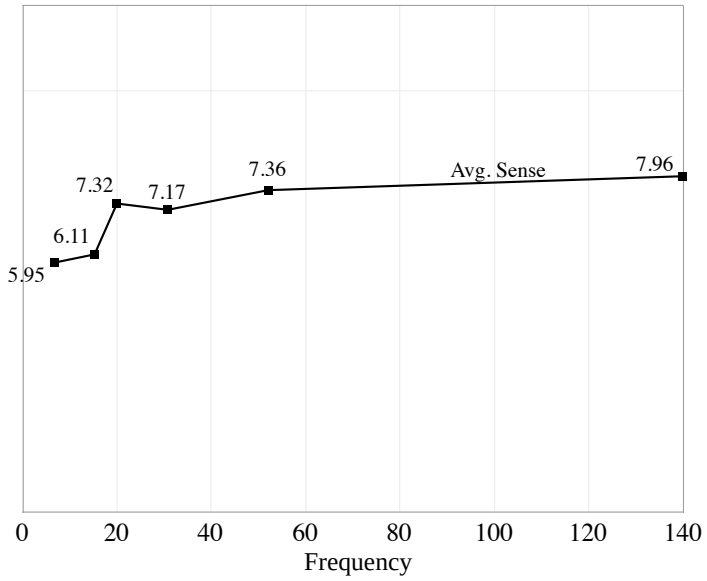


Figure 5.9: Statistics, Average number of Senses and Word Frequency. Stats estimated on 200 words at each frequency

One may argue that the disambiguation of low frequency words is easier than the higher ones because usually they contain at max 2~3 senses or in other words less ambiguous. This is not the case as shown by Figure 5.9. Even the words whose frequency is near to 5 consists on average 5.95 senses. Because there are almost no example sentences for such senses, it is difficult to disambiguate them using our proposed method. To tackle that problem, algorithm like Lesk (using only the sense definitions) [11] or approaches like online learning may compliment the proposed WSD for such words. We will look into incorporating these approaches in future.

Chapter 6

Conclusion

In this thesis, we proposed a precision oriented example based WSD method. Proposed sentence similarity measures compute a score by exploiting collocation information and comparing syntactic dependency relations for a target word, just by using example sentences from an MRD. We also showed the reliability of these measures towards increasing precision by constraining a threshold. Being a precision oriented approach, robustness to the system comes by combining with a Robinson classifier. Reliable bilingual example sentences are extracted from an automatically aligned parallel corpus to enlarge the example database. Injection of extracted examples substantially increases the performance of all classifiers. One of the advantages of our method is that it does not require sense tagged corpora. It achieved 65% accuracy, which is 7% better than the baseline.

For the future work, we would prepare an evaluation data for low frequency words and evaluate the performance of the proposed WSD method. Current example sentences expansion seems less effective for low frequency, therefore we will explore different methods such as paraphrasing to expand the examples database in future.

Selected Publications

1. Pulkit Kathuria, Kiyoaki Shirai. Example Based Word Sense Disambiguation towards Reading Assistant System. In *18th Annual NLP Meeting*. Japan, 2012 (Non-refereed)

2. Pulkit Kathuria, Kiyoaki Shirai. Example Based Word Sense Disambiguation with Automatically Acquired Examples from Parallel Corpus (in Japanese). In *IPSJ SIG Technical Report*. vol.207.No.3. Japan, 2012. (Non-refereed)

3. Pulkit Kathuria, Kiyoaki Shirai. Word Sense Disambiguation Based on Example Sentences in Dictionary and Automatically Acquired from Parallel Corpus. In *8th International Conference on Natural Language Processing*. JapTal 2012, LNAI 7614, pp.210-221, 2012. (Refereed)

Acknowledge

Foremost, I would like to express my sincere gratitude to Professor Shirai for his support towards my Masters, research and future career, for his patience, motivation and immense knowledge. Professor's guidance, teaching, problem solving and many more has helped me during all time of my research. I would like to thank him as my immediate supervisor, Professor and researcher, without him I couldn't have imagined such an inspirational mentor.

Besides my advisor, I would like to thank Professor Shimazu for his support, encouragement and insightful comments which lead to explore future directions in my research in Natural Language Processing.

I also want to sincerely thank Professor Yoshitaka for his support in my sub research theme during Masters.

I sincerely also want to thank JAIST for being the modest university to International Students and JASSO scholarship, financial support of which helped me to successfully finish my research.

Last but not the least, I would like to thank my life partner Chieko for standing patiently & firmly, with a strong belief in me, all the time. I also thank her as an artist who crafted a Japanese heart in me. And, greatly thankful to my beloved father for everything.

Bibliography

[1] Jeremy Blosser and David Josephsen. Awarded best paper! - scalable centralized bayesian spam mitigation with bogofilter. In *Proceedings of the 18th USENIX conference on System administration*, LISA '04, pages 1–20, Berkeley, CA, USA, 2004. USENIX Association.

[2] Francis Bond, Eric Nichols, and Jim Breen. Enhancing a dictionary for transfer rule acquisition.

[3] Eric Brill and Jun Wu. Classifier combination for improved lexical disambiguation. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics - Volume 1*, ACL '98, pages 191–195, Stroudsburg, PA, USA, 1998. Association for Computational Linguistics.

[4] Marine Carpuat and Dekai Wu. Improving statistical machine translation using word sense disambiguation. In *The 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 61–72, Prague, Czech Republic, 2007.

[5] Yee Seng Chan and Hwee Tou Ng. Word sense disambiguation improves statistical machine translation. In *45th Annual Meeting of the Association for Computational Linguistics (ACL-07)*, pages 33–40, Prague, Czech Republic, 2007.

[6] Ido Dagan and Alon Itai. Word sense disambiguation using a second language monolingual corpus. *Computational Linguistics*, 20(4):563–596, December 1994.

[7] Gerard de Melo and Gerhard Weikum. Extracting sense-disambiguated example sentences from parallel corpora. In *Proceedings of the 1st Workshop on Definition Extraction*, WDE '09, pages 40–46, 2009.

[8] Atsushi Fujii, Kentaro Inui, Takenobu Tokunaga, and Hozumi Tanaka. Selective sampling for example-based word sense disambiguation. *Computational Linguistics*, 24(4):573–597, 1998.

[9] Sanae Fujita and Akinori Fujino. Word Sense Disambiguation by Combining Labeled Data Expansion and Semi-Supervised Learning Method. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 676–685, 2011.

[10] Hyun Ah Lee and Gil Chang Kim. Translation selection through source word sense disambiguation and target word selection. In *Proceedings of the 19th International Conference on Computational Linguistics*, volume 1 of *COLING '02*, pages 1–7, 2002.

[11] Michael Lesk. Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone. In *Proceedings of the 5th annual international conference on Systems documentation*, SIGDOC '86, pages 24–26, New York, NY, USA, 1986. ACM.

[12] Rada F. Mihalcea. Bootstrapping large sense tagged corpora. In *In Proceedings of the 3rd International Conference on Language Resources and Evaluations (LREC), Las Palmas*, 2002.

[13] Guido Minnen, John Carroll, and Darren Pearce. Robust, applied morphological generation. In *Proceedings of the First International Natural Language Generation Conference*, pages 201–208, 2000.

[14] Natural Institute for Japanese Language and Linguistics, editor. *Bunrui Goi Hyo*. Dainippon Tosho, 2004.

[15] Roberto Navigli. Word sense disambiguation: A Survey. *ACM Computing Surveys*, 41(2):1–69, 2009.

[16] NIICT. http://alaginrc.nict.go.jp/wikicorpus/index.html, 2011.

[17] Manabu Okumura, Kiyoaki Shirai, Kanako Komiya, and Hikaru Yokono. SemEval-2010 task: Japanese WSD. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 69–74, 2010.

[18] Ted Pedersen. A simple approach to building ensembles of naive bayesian classifiers for word sense disambiguation. In *Proceedings of the 1st North American chapter of the Association for Computational Linguistics conference*, NAACL 2000, pages 63–69, Stroudsburg, PA, USA, 2000. Association for Computational Linguistics.

[19] Gary Robinson. A statistical approach to the spam problem. *Linux J.*, 2003(107), March 2003.

[20] Celina Santamaría, Julio Gonzalo, and Javier Artiles. Wikipedia as sense inventory to improve diversity in web search results. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ACL '10, pages 1357–1366, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

[21] Kiyoaki Shirai and Takayuki Tamagaki. Word sense disambiguation using heterogeneous language resources. In *First International Joint Conference on Natural Language Processing*, pages 614–619, 2004.

[22] Masao Utiyama and Hitoshi Isahara. Reliable measures for aligning Japanese-English news articles and sentences. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics*, volume 1 of *ACL '03*, pages 72–79, 2003.

[23] Hans van Halteren, Jakub Zavrel, and Walter Daelemans. Improving data driven wordclass tagging by system combination. In *Proceedings of the 17th international conference on Computational linguistics - Volume 1*, COLING '98, pages 491–497, Stroudsburg, PA, USA, 1998. Association for Computational Linguistics.