Discrimination of Task-Related Words for Vocabulary Design of Spoken Dialog Systems

Akinori Ito¹, Toyomi Meguro¹, Shozo Makino¹ and Motoyuki Suzuki²

¹Graduate School of Engineering, Tohoku University, Japan
²Institute of Technology and Science, The University of Tokushima, Japan

{aito,meguro-t,makino}@makino.ecei.tohoku.ac.jp, moto@m.ieice.org

Abstract

This paper describes a method used to determine if a specific word is related to a certain spoken dialog task. In most ordinary spoken dialog systems, only the words that are actually used to achieve the task are included in the vocabulary. Therefore, the system cannot recognize utterances that contain OOV words that are related to the task. Therefore, we developed a method for determining the words that are related to a specified task in order to augment the system’s vocabulary. Our method is based on word similarity. We examined three similarities: word occurrence frequency on the Web, distance in a thesaurus and word similarity using LSA. The experiment revealed that the thesaurus-based and LSA-based methods have an OOV problem. To solve the problem, we developed a way to combine these two methods with the Web-based method. In addition, we tried combining the methods using the AdaBoost algorithm.

Index Terms: word similarity, thesaurus, web search engine, LSA

1. Introduction

User interfaces that employ spoken dialog have been actively developed. In an ordinary spoken dialog system that performs a certain task, the system accepts only words that are concerned with the task. For example, most of flight information systems have only the names of the cities with airports. Therefore, while the utterance “From New York to Tokyo” can be accepted by the system, the utterance “From New York to Saitama” cannot be accepted because Saitama has no airport. In this case, there are two possibilities. If the system is able to reject the unknown words, it would answer “I could not understand you. Please speak again.” Or, the system would misrecognize the utterance and respond “OK, I will reserve a ticket from New York to San Diego.” In either case, the system cannot inform the user of the most important thing—Saitama has no airport. Therefore, the user will not understand why the recognition failed.

The traditional way of treating this problem is to detect the out-of-vocabulary (OOV) words[1, 2, 3]. The OOV detection methods use a model of OOV words, which are composed of sub-word units. By using the OOV detection, we can know that the user’s utterance contains unknown words. However, we cannot know what kind of word the OOV word is, because the detected OOV words are just a sequence of sub-word units. Therefore, we must estimate the class of the OOV word[4], which requires the training data for the words that belong to the class in which we are interested.

Considering a spoken dialog system with a relatively small vocabulary, a more straightforward solution to this problem is to register all related words to the vocabulary of the speech recognizer. After recognition, if the recognized word is inappropriate according to the current task, the dialog system can prompt the user to select another choice. However, it is not always easy to manually select words that are concerned with the task. The recognition performance deteriorates if we add too many words that are not related to the task. If we could measure the semantic relevance of a word to the task, it would be a great help in finding words to be added to the vocabulary.

Figure 1 shows a schematic diagram of the vocabulary augmentation. First, we have a vocabulary of manually selected words that are used in a specific task. Then, we choose the relevant words from a large vocabulary created from another language source, then we merge the original vocabulary and the relevant words to create the augmented vocabulary.

In this paper, we investigate several methods for measuring the semantic relevance of a word to a specific task. First, we compare several methods to measure the relevance, and then we develop a method for combining the relevance scores obtained from multiple methods.

2. Measurement of Relevance

2.1. Word similarity and relevance

In this work, we assume only nouns as the words to be added to the vocabulary because of data availability. In principle, our method can also be applied to words other than nouns.

First, we assume that we have a vocabulary \( V_i \) that contains
nouns that are used in the task \( t \). If we can calculate the similarity between two words, we can calculate the relevance score for a word \( w \) to the task \( t \) by the following formula.

\[
R_t(w) = \frac{1}{|V_t|} \sum_{u \in V_t} s(w, u) 
\]

where \( s(w, u) \) is the similarity between two words, \( w \) and \( u \). We examined the following three similarity measures:

1. **Web-based method**
   The first one is based on a Web search engine[5]. Let \( N(w_1, w_2, \ldots) \) be the number of Web pages in which the words \( w_1, w_2, \ldots \) appear. Then, the similarity between words \( w \) and \( u \) is calculated using the Dice coefficient:

\[
s(w, u) = \frac{2N(w, u)}{N(w) + N(u)}
\]

2. **Thesaurus-based method**
   This method is based on a thesaurus. As words in a thesaurus are organized as a network structure (mostly a tree structure), we can measure the distance of two words as the average length of all paths between the two words in the network. Using the distance in a thesaurus \( d(w, u) \), we define

\[
s(w, u) = -d(w, u).
\]

3. **LSA-based method**
   The third method is based on latent semantic analysis (LSA), which converts a word into a vector in a vector space. By using LSA, we can calculate the similarity of two words as the cosine similarity of two vectors[7]. Let \( v(w) \) be the vector corresponding to the word \( w \). Then, we can define

\[
s(w, u) = \frac{v(w) \cdot v(u)}{|v(w)||v(u)|}.
\]

After calculating the relevance score, we use a threshold \( \theta \) to determine if the word is relevant to the task. If \( R(w) > \theta \), the word \( w \) is regarded as a task-related word.

### 2.2. Experiment

We carried out an experiment to compare the discrimination rate for each method. In the experiment, we used Japanese as the target language.

We used Goi-Taikei[8] and Gendai Nihongo Meishi Thesaurus[9] for the thesaurus-based method. Goi-Taikei has about 300,000 words including nouns, proper nouns, verbs and adjectives. Gendai Nihongo Meishi Thesaurus has 70,000 nouns organized as a tree structure. We used 11 years of newspaper articles from the Mainichi Shimbun as training data for the LSA-based method. Infomap-NLP[10] was used to perform LSA. The number of singular values was set to 100.

We examined three tasks: fruits, clothes and beverages. We prepared a 30-word vocabulary \( V_t \) for each task \( t (t = 1, 2, 3) \). In addition, we prepared a 50-word vocabulary \( V_O \) that are not relevant to any of the three tasks. Table 1 shows examples of the words involved in the tasks. When measuring the relevance score of a task-related word, the word to be measured was excluded from the vocabulary for the relevance calculation, i.e.:

\[
R_t(w) = \frac{1}{|V_t|-1} \sum_{u \in V_t} s(w, u).
\]

For each task \( t (t = 1, 2, 3) \), we measured the discrimination rate

\[
D(\theta, t) = \frac{\sum_{w \in V_t} T(w, t, \theta) + \sum_{w \in V_O} (1 - T(w, t, \theta))}{|V_t| + |V_O|}
\]

where

\[
T(w, t, \theta) = \begin{cases} 1 & R_t(w) > \theta \\ 0 & R_t(w) \leq \theta \end{cases}
\]

Figures 2, 3, 4 and 5 show the discrimination rates for all methods with respect to the threshold value. Note that the lowest discrimination rate is 0.5 because of the two-class discrimination problem. We found it possible to achieve a high discrimination rate by using the appropriate threshold, and the optimum threshold values were not very different among the different tasks.

Figure 6 shows the discrimination rates for all methods when the optimum thresholds are used, calculated as

\[
\bar{D} = \frac{1}{3} \sum_{t=1}^{3} \max \theta D(\theta, t).
\]

We discriminate a word as either ‘task-related’ or ‘not task-related’, but we cannot determine the relevance of the word if it is not included in the thesaurus or vocabulary of the LSA.

### Table 1: Examples of words in the tasks

<table>
<thead>
<tr>
<th>Vocab</th>
<th>Task</th>
<th>Examples</th>
<th># Words</th>
</tr>
</thead>
<tbody>
<tr>
<td>V1</td>
<td>fruits</td>
<td>apple, orange, …</td>
<td>30</td>
</tr>
<tr>
<td>V2</td>
<td>clothes</td>
<td>shirts, trousers, …</td>
<td>30</td>
</tr>
<tr>
<td>V3</td>
<td>beverages</td>
<td>tea, coffee, …</td>
<td>30</td>
</tr>
<tr>
<td>V_O</td>
<td>not relevant</td>
<td>medicine, space, …</td>
<td>50</td>
</tr>
</tbody>
</table>
Thus, we have the third category ‘unknown’ in the figure. No
unknown words were observed in the Web-based method.
The discrimination rate for the Web-based method was the highest
among the methods, but the rate of incorrect decision was also
the highest. The thesaurus-based method using Goi-Taikei gave
the fewest incorrect decisions, which means that the thesaurus-
based method is accurate as long as the word is included in the
vocabulary.

3. Combination of Classifiers

3.1. Cascade combination

As shown in Fig. 6, the Web-based method has the advantage
of a near absence of out-of-vocabulary (OOV) problems, but its
discrimination rate for known words is not high. On the other
hand, the thesaurus-based and LSA-based methods have OOV
problems, but their discrimination rates for the known words are
high. Considering these results, it makes sense to combine the
Web-based and other methods to compensate for each other’s
weak points.

The basic idea for combining them is to use the thesaurus-
based or LSA-based method if the word is a known word; other-
wise, the Web-based method is employed. In general, when
we have two classifiers $C_0(w)$ and $C_1(w)$, where

$$C_0(w) = \begin{cases} 
1 & \text{if } w \text{ is determined as task-related} \\
0 & \text{if } w \text{ is an unknown word} \\
-1 & \text{if } w \text{ is not determined as task-related} 
\end{cases}$$

and

$$C_1(w) = \begin{cases} 
1 & \text{if } w \text{ is determined as task-related} \\
-1 & \text{if } w \text{ is not determined as task-related} 
\end{cases}$$

then we can consider a new classifier that is a cascade combina-
tion of the two classifiers:

$$C(w) = \begin{cases} 
C_0(w) & \text{if } C_0(w) \neq 0 \\
C_1(w) & \text{if } C_0(w) = 0 
\end{cases}$$

We examined the thesaurus-based method using the two
thesauri and the LSA-based method as $C_0(w)$. As for $C_1(w)$,
we examined the following three classifiers:

1. The web-based method
2. Always task-related (i.e. $C_1(w) = 1$)
3. Always non-task-related (i.e. $C_1(w) = 0$)
Thresholds for the discrimination were the same as the optimum ones in the previous experiment.

The experimental results are shown in Fig. 7. ‘Web’ indicates a result obtained by the Web-based method only. The combination of Goi-Taikei and Web-based method gave the best results, proving the effectiveness of combining the classifiers was effective.

3.2. Parallel combination by AdaBoost
Next, we examined combining multiple classifiers using the AdaBoost algorithm[11]. We combined the six combination methods and the Web-based method as classifiers for combination. The six classifiers were a cascade combination of the following $C_0(w)$ and $C_1(w)$: the thesaurus-based method using the two thesauri and the LSA-based method as $C_0(w)$, and ‘always task-related’ and ‘always non-task-related’ as $C_1(w)$. We did not use methods for a combination that was a cascade combination of any method and the Web-based method, because there were two thresholds to optimize, making it difficult to apply an ordinary AdaBoost algorithm. In principle, of course, it is possible to use a method with two parameters as a weak classifier for boosting. Combining such methods is our future work.

Figure 8 shows the results. ‘Web’ indicates the result of the Web-based method, ‘Goi-Taikei+Web’ indicates the result of the cascade combination of Goi-Taikei and the Web-based method, and ‘AdaBoost’ is the result based on the AdaBoost algorithm. From these results, we can confirm the AdaBoost-based method combination did improve the accuracy of discrimination. However, the performance was not better than that of the cascade combination of Goi-Taikei and Web-based method.

4. Conclusions
In this paper, we investigated several methods for determining whether or not a word is related to a specific task. The proposed method is based on word similarity. We examined three kinds of similarities using a Web-based method, thesaurus-based method and LSA-based method. The experimental results revealed that the thesaurus-based and LSA-based methods have OOV problems. To solve the OOV problem, we developed a cascade combination of Web-based method and other method, which improved the accuracy of discrimination. In addition, we examined combining multiple classifiers using the AdaBoost algorithm.

5. References