Generating More Specific Questions for Acquiring Attributes of Unknown Concepts from Users

Tsugumi Otsuka[†], **Kazunori Komatani**[†], **Satoshi Sato**[†], **Mikio Nakano**[‡]

† Graduate School of Engineering, Nagoya University, Nagoya, Aichi 464–8603, Japan
‡ Honda Research Institute Japan Co., Ltd., Wako, Saitama 351–0114, Japan
{t_ootuka,komatani,ssato}@nuee.nagoya-u.ac.jp, nakano@jp.honda-ri.com

Abstract

Our aim is to acquire the attributes of concepts denoted by unknown words from users during dialogues. A word unknown to spoken dialogue systems can appear in user utterances, and systems should be capable of acquiring information on it from the conversation partner as a kind of self-learning process. As a first step, we propose a method for generating more specific questions than simple wh-questions to acquire the attributes, as such questions can narrow down the variation of the following user response and accordingly avoid possible speech recognition errors. Specifically, we obtain an appropriately distributed confidence measure (CM) on the attributes to generate more specific questions. Two basic CMs are defined using (1) character and word distributions in the target database and (2) frequency of occurrence of restaurant attributes on Web pages. These are integrated to complement each other and used as the final CM. We evaluated distributions of the CMs by average errors from the reference. Results showed that the integrated CM outperformed the two basic CMs.

1 Introduction

In most spoken dialogue systems, knowledge bases for the systems are constructed off-line. In other words, they are not updated during dialogues. On the other hand, humans update their knowledge not only by reading books but also through interaction with other people. When they encounter an unknown word during conversations, humans notice that it is new to them and acquire knowledge about it by asking their conversational partner. This self-learning process is one of the most intelligent features of humans. We think that applying this intelligent feature to spoken dialogue systems will make them more usable.

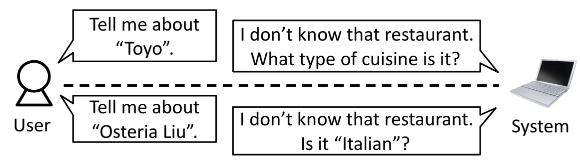

Figure 1: Example of simple and specific questions.

We present a method that generates appropriate questions in order to acquire the attributes of a concept that an unknown word denotes when it appears in a user utterance. Here, we define unknown words as those whose attributes necessary for generating responses were not defined by the system developer; that is, unknown to the response generation module in the spoken dialogue system. The system cannot reply to user utterances including such words even if they are correctly recognized by its automatic speech recognition (ASR) module.

Questions to the user to acquire the attribute should be specific. In spoken dialogue systems, specific questions are far preferable to wh-questions because they can narrow down variations of the following user response. Such questions lead to a better ASR performance of the response and reduce the risk that it includes new other unknown words.

Two example dialogues are shown in Figure 1. Since our target task is restaurant database retrieval, we set the unknown words as restaurant names and the attribute as their cuisine in our restaurant database. In the examples shown, the system uses a simple wh-question (the upper part) and a specific Yes-No question (the lower part) to obtain cuisine types. Here, "Toyo" and "Osteria Liu" are restaurant names. We assume that the

Proceedings of the SIGDIAL 2013 Conference, pages 70–77,
Metz, France, 22-24 August 2013. ©2013 Association for Computational Linguistics

Table 1: Question types according to the number of cuisines (num).

num	Question form	Example
1	Yes-No question	Is it cuisine c_1?
2	Alternative question	Which cuisine is it, c_1 or c_2?
3	3-choice question	Which cuisine is it, c_1, c_2, or c_3?
$\geq$4	Wh-question	What cuisine is it?

system already knows these are restaurant names but does not know its attributes such as its cuisine type. The system uses a wh-question for "Toyo" since no clue is obtained for it. In contrast, since "Osteria Liu" contains information on cuisines in the name itself, a concrete Yes-No question is used to ask whether the cuisine is "Italian".

We propose a method for providing a well-distributed confidence measure (CM) to generate more specific questions. For this purpose, we estimate the cuisine type of a restaurant from its name, which is assumed to be unknown to the system. There have been many previous studies that estimate word and character attributes using Web information (Pasca et al., 2006; Yoshinaga and Torisawa, 2007). Our two estimation methods are relatively simpler than these studies, since our main focus is to generate more concrete questions on the basis of appropriate CMs. That is, the CMs should be high when the system seems to correctly estimate a cuisine type and low when the estimation seems difficult.

We assume a restaurant name as the input; that is, we suppose that the system can recognize the restaurant name in the user's utterance correctly by its ASR module and understand it is a restaurant name by its LU module. Nevertheless, it still remains unknown to its response generation module. This is a feasible problem when using a large vocabulary continuous speech recognition (LVCSR) engine containing over several million words (Jyothi et al., 2012) and a statistical named entity (NE) tagger (Tjong Kim Sang and Meulder, 2003; Zhou and Su, 2002; Ratinov and Roth, 2009).

The problem we tackle in this paper is different from trying to estimate the NE class of an unknown word (Takahashi et al., 2002; Meng et al., 2004). We assume the system already knows that it is a restaurant name. Rather, we try to acquire the attribute (e.g., cuisine type) of the concept of the unknown word, which is required for generating responses about the restaurant in subsequent dialogues.

2 Generating Questions Based on CM

The system determines a question type on the basis of CM. The CM is estimated for each cuisine type c_j in the target database. In this paper, the number of cuisine types is 16, all of which are in our restaurant database; that is, $c_j \in C$ and $|C| = 16$.

Table 1 shows the four question types and their examples. These are determined by parameter num, which is the number of cuisine types that should be included in the question. If the system obtains one cuisine type that it is very confident about and thus has a high CM, it should generate the most specific question, i.e., a Yes-No question; in this case, the number should be 1. In contrast, if unreliable cuisine types are obtained, which means lower CMs, the system generates questions including several cuisine types.

The num can be determined by Equation (1):

$$num = min(n) \text{ s.t. } \sum_{j=1}^{n} CM(c_j) > \theta, \qquad (1)$$

where $CM(c_j)$ is a confidence measure for cuisine type c_j in its descending order. θ is a constant and can be manually decided considering the distribution of $CM(c_j)$. This equation means that if only the $CM(c_1)$ is greater than θ (i.e., $n = 1$), the system generates a specific question including only cuisine type c_1, while if the total from $CM(c_1)$ to $CM(c_4)$ is smaller than θ (i.e., $n = 4$), the system does not use estimated cuisine types and instead generates a wh-question.

If the CM on the cuisine type is well-distributed, the system can generate appropriate questions. In the following section, methods to obtain such CMs are explained.

3 Estimating Cuisine Types and Calculating CM

The final CM is obtained by integrating two basic CMs. The system then uses this final CM to

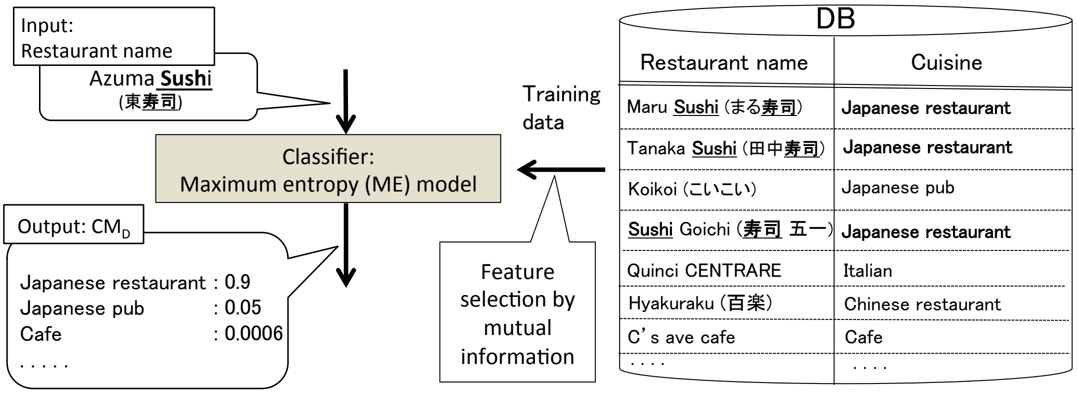

Figure 3: Overview of CM_D calculation.

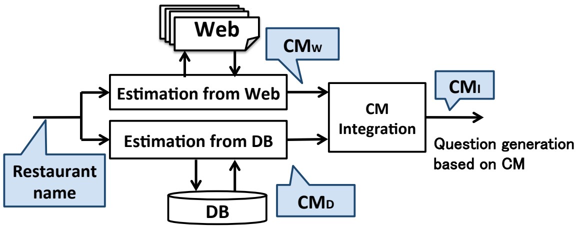

Figure 2: Process overview.

generate questions. The two basic CM estimation methods are:

1. Using word and character distribution in the target database

2. Using frequency of the restaurant attributes on the Web

A process overview of the proposed method is shown in Figure 2. Its input to the system is an unknown restaurant name and its output is the estimated CMs. The system generates questions on the basis of the estimated CMs, which are calculated for each cuisine type.

3.1 Attribute Estimation Using Word and Character Distribution in Database

We estimate the cuisine types of an unknown restaurant by using the word and character distribution in the target database. The target database contains many pairs of restaurant names and cuisine types. The estimation is performed by using supervised machine learning trained with the pairs. The overview of calculating CM_D is shown in Figure 3. This approach is based on our intuition that some cuisine types can be estimated from restaurant names on the basis of their character types or typical character sequences they contain. For example, a restaurant name composed of only katakana[1] is probably a French or Italian restaurant because words imported from other countries to Japan are called "katakana loanwords" and are written in katakana characters (Kay, 1995).

We use the maximum entropy (ME) model (Berger et al., 1996) as a classifier. Its posterior probability $p(c_j|s_i)$ is used as a CM_D denoting the CM estimated using a database. CM_D is calculated as

$$\begin{aligned} CM_D(s_i, c_j) &= p(c_j|s_i) \\ &= \frac{1}{Z} \exp\left[\vec{\lambda}(c_j) \cdot \vec{\phi}(s_i)\right], \quad (2) \end{aligned}$$

where s_i is a restaurant name, c_j ($\in C$) is a cuisine type, $\vec{\phi}(s_i)$ is a feature vector obtained from a restaurant name, $\vec{\lambda}(c_j)$ is a weight vector, and Z is a normalization coefficient that ensures $\sum_{c_j} CM_D(s_i, c_j) = 1$.

We use three types of feature vectors obtained from each restaurant name:

- Character n-grams (n = 1, 2, 3)
- Words
- Character types

The feature values of the character n-gram and the word are scored as 1 if such features are contained in the restaurant name. The Japanese morphological analyzer Mecab (Kudo et al., 2004) with the IPADIC dictionary is used to segment restaurant names into word sequences. The character type

[1] Katakana is a Japanese syllabary. There are three kinds of characters in Japanese. Kanji (Chinese character) are logograms and hiragana and katakana are syllabaries. Katakana is mainly used for writing imported words and hiragana is used for writing original Japanese words.

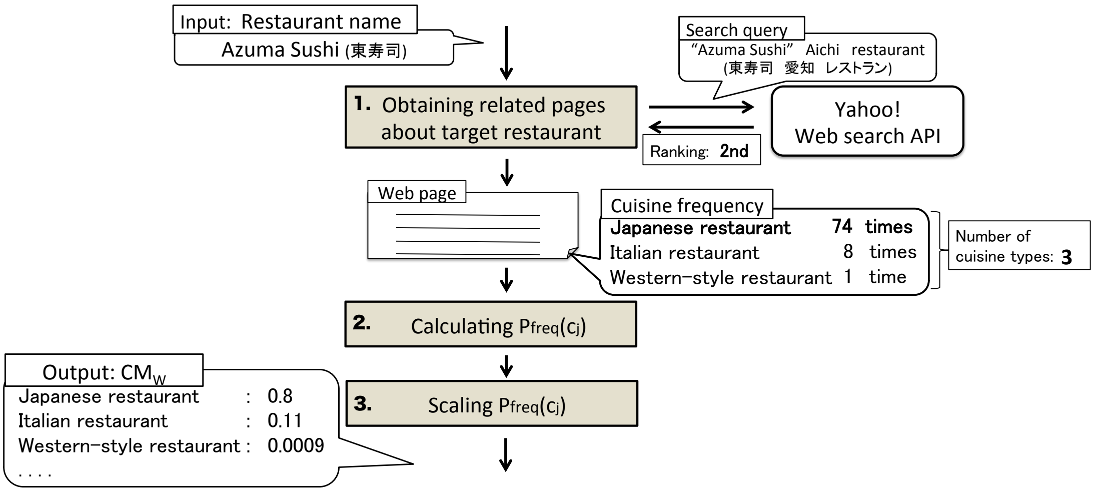

Figure 4: Overview of CM_W calculation.

is represented by the four character types used in the Japanese writing system: hiragana, katakana, kanji (Chinese characters), and romaji (Roman letters). For example, the restaurant name "Maru Sushi (まる寿司)" includes two character types: "Maru (まる)" is written in hiragana and "Sushi (寿司)" is written in kanji. Therefore, the feature values for hiragana and kanji are both 1, while those for katakana and romaji are 0. Another example is shown using the restaurant "IB cafe (IB カフェ)", in which the "IB" part is romaji and the "cafe (カフェ)" part is katakana. Therefore, in this case, the feature values of katakana and romaji are 1 and those of hiragana and kanji are 0.

We perform feature selection for the obtained features set (Guyon and Elisseeff, 2003). The classifier needs to be built without overfitting because we assume that a restaurant name as the input to this module is unknown and does not exist in the database. We use the mutual information (Peng et al., 2005; Yang and Pedersen, 1997) between each feature and the set of cuisine types as its criterion. This represents how effective each feature is for the classification. For example, in the features obtained from the restaurant name "まる寿司", which is a Japanese restaurant, the 2-gram feature "寿司" frequently co-occurs with the cuisine type "Japanese restaurant". This is an effective feature for the cuisine type estimation. In contrast, the 2-gram feature "まる" is not effective because its co-occurrence with cuisine types is infrequent. Mutual information is calculated as

$$I(f_k; C) = \sum_{c_j \in C} p(f_k, c_j) \log \frac{p(f_k, c_j)}{p(f_k)p(c_j)}, \quad (3)$$

where $p(f_k)$ is an occurrence probability of feature f_k in the database, $p(c_j)$ is an occurrence probability of cuisine type c_j ($\in C$), and $p(f_k, c_j)$ is a joint probability of the feature and the cuisine type.

Features having lower mutual information values are removed until we deem that overfitting has been avoided, specifically, when the estimation accuracies become almost the same between the closed and open tests. We confirm this by cross-validations (CV) instead of open tests.

3.2 Estimation Using the Web

We estimate a restaurant's cuisine type and calculate CMs by using its frequency on the Web as CM_W. This is based on an assumption that a restaurant's name appears with its cuisine type on Web pages. CM_W is calculated in the following steps, as shown in Figure 4.

1. Obtaining related Web pages:
   Twenty pages per search query were obtained, as this was the limit of the number of pages when this experiment was performed. We used the Yahoo! Web search API[2]. The query is formed with the target restaurant name and the following two words: "Aichi (愛知)" and "restaurant (レストラン)". The two are added to narrow down the search result since our domain is a restaurant search in Aichi prefecture. For example, the query is "`<rest>` 愛知 レストラン" for the target restaurant name `<rest>`.

[2] http://developer.yahoo.co.jp/webapi/search/websearch /v2/web search.html

2. Calculating $P_{freq}(c_j)$:
We count the frequency of each cuisine type c_j in the i-th Web pages, which are ranked by the Web search API. We then sum up the frequency through all the obtained pages and calculate its posterior probability.

$$P_{freq}(c_j) = \frac{\sum_i w_i \cdot freq_i(c_j)}{\sum_{c_j} \sum_i w_i \cdot freq_i(c_j)} \quad (4)$$

Here, $freq_i(c_j)$ is the frequency of c_j in the i-th page. Weight w_i is calculated using two factors, $rank(i)$ and $cuisine(i)$:

$$w_i = \frac{1}{rank(i) \cdot cuisine(i)} \quad (5)$$

   (a) $rank(i)$: The ranking of pages in the Web search API
   We assume that a Web page is more related to the target restaurant if the Web search API ranks it higher.
   (b) $cuisine(i)$: The number of cuisine types in the i-th Web page
   We assume that a Web page containing many different cuisine types does not indicate one particular cuisine. For example, a page on which only "Chinese restaurant" appears is more reliable than that on which more cuisine types ("Chinese restaurant", "Japanese restaurant", "Japanese pub", and "Western-style restaurant", for example) appear, as a page indicating a "Chinese restaurant".

3. Scaling $P_{freq}(c_j)$:
CM_W is calculated by scaling each $P_{freq}(c_j)$ with the corresponding α_j. α_j is a scaling coefficient that emphasizes the differences among CM_W: α_j is equal to or smaller than 1 and becomes smaller as j increases.

$$CM_W(c_j) = \frac{\alpha_j P_{freq}(c_j)}{\sum_{c_j} \alpha_j P_{freq}(c_j)} \quad (6)$$

$$\alpha_j = P_{freq}(c_j)/P_{freq}(c_1) \quad (7)$$

3.3 Integration of CMs

We define CM_I by integrating the two basic CMs: CM_D and CM_W. Specifically, we integrate them by the logistic regression (Hosmer Jr. et al., 2013) shown in Equation (8). The optimal parameters, i.e., weights for the CMs, are determined using a data set with reference labels. The teacher signal is 1 if the estimated cuisine type is correct and 0 otherwise.

$$CM_I(c_j) = \frac{1}{1 + \exp(-f(c_j))} \quad (8)$$

$$f(c_j) = w_D CM_D(c_j) + w_W CM_W(c_j) + w_0$$

Here, w_D and w_W are the weights for CM_D and CM_W, and w_0 is a constant.

4 Experiment

We evaluate our method to obtain the CMs from three aspects. First, we evaluate the effect of feature selection based on mutual information. Second, we evaluate how the CMs were distributed and whether they were appropriate measures for question generation. Third, we determine the effectiveness of integrating the two basic CMs. In this paper, we used a restaurant database in Aichi prefecture containing 2,398 restaurants with 16 cuisine types.

4.1 Effect of Feature Selection Based on Mutual Information

We determined whether overfitting could be avoided by feature selection based on mutual information in the estimation using a database. We regard overfitting to be avoided when estimation accuracies become almost the same between the closed and open tests. For the closed test, estimation accuracy was calculated for all 2,398 restaurants in the database by using a classifier that was trained with the same 2,398 restaurants. For the open test, it was calculated by 10-fold CV for the 2,398 restaurants. This experiment is not for determining a feature set but rather for determining a feature selection ratio. That is, the feature selection result is kept not as a feature set but as a ratio. The resulting ratio is applied to the number of features appearing in another training data (e.g., that in Section 4.2) and then the feature set is determined.

Figure 5 shows the estimation accuracy of the closed test and the 10-fold CV when the feature selection was applied. The horizontal axis denotes ratios of features used to train the classifier out of 20,679 features in total. They were selected in descending order of mutual information. The vertical axis denotes the estimation accuracy of the

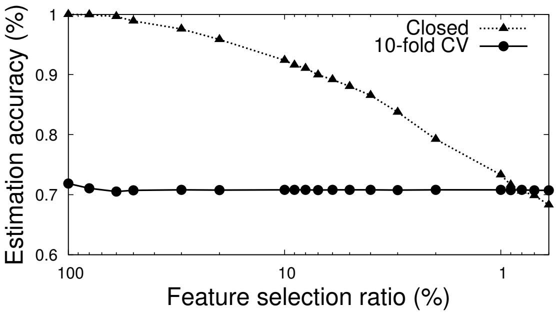

Figure 5: Estimation accuracies of closed test and 10-fold CV.

cuisine types. Figure 5 shows that, at first, overfitting occurs if all features were used for training; that is, the feature selection ratio = 100%. This can be seen by the difference in estimation accuracies, which was 28.1% between the closed test and the 10-fold CV. The difference decreased as the number of used features decreased, and almost disappeared at feature selection ratio = 0.8%. In these selected features, as an example, the 2-gram "gyoza (餃子)", which seems intuitively effective for cuisine type estimation is, included[3].

4.2 Evaluation for Distribution of CMs

We evaluate the distribution of CMs obtained with the estimation results. Specifically, we evaluated three types of distributions: CM_D, CM_W, and CM_I. We extracted 400 restaurants from the database and used them as evaluation data. The remaining 1,998 restaurants were used as training data for the classifier to calculate CM_D. In all features obtained from these 1,998 restaurants, the ME classifier uses 0.8% of them, which is the feature selection ratio based on the mutual information determined in Section 4.1. That is, the feature set itself obtained in the feature selection is not delivered into the evaluation in this section.

We used average distances between each CM score and its reference as the criterion to evaluate the distribution of the CMs. Generally, CMs should be as highly scored as possible when the estimation is correct and as lowly scored as possible otherwise. We calculate the distances over the 400 estimation results.

$$eval(CM_x) = \frac{\sum_i^N |CM_x^i - \phi_x^i|}{N} \quad (9)$$

Here, N is the total number of the estimation result, so $N = 400$ in this paper. ϕ_x^i for CM_x^i is defined as

$$\phi_x^i = \begin{cases} 1, & \text{If estimation result } i \text{ is correct} \\ 0, & \text{Otherwise} \end{cases} \quad (10)$$

Note that ϕ_x depends on CM_x because estimation results differ depending on the CM_x used.

We also set the majority baseline as Equation (11). Here, all CMs are regarded as 0 or 1 in Equation (9). Because there were more correct estimation results than incorrect ones, as shown in Table 2, we used 1 for the majority baseline, as

$$MB(CM_x) = \frac{\sum_i^N |1 - \phi_x^i|}{N}. \quad (11)$$

Table 3: Evaluation against each CM.

	$eval(CM_x)$	$MB(CM_x)$
CM_D	0.31	0.37
CM_W	0.28	0.32
CM_I	0.25	0.28

The results are shown in Table 3. A comparison of the three $eval(CM_x)$ demonstrates that the integrated CM_I is the most appropriate in our evaluation criterion because it is the lowest of the three. The relative error reduction rates from CM_I against CM_D and CM_W were 16% and 37%, respectively. Each $eval(CM_x)$ outperformed the corresponding majority baseline.

4.3 Effectiveness of Integrated CM

We verify the effectiveness of the CM integration from another viewpoint. Specifically, we confirm whether the number of correct estimation results increases by integration.

First, we show the distribution of the three CMs and whether they were correct or not in Table 2. The bottom row of the table shows that CM_I obtained correct estimation results for 297 restaurants, which is the highest of the three CMs.

More specifically, we investigated how many estimation results changed by using the three CMs. Here, an estimation result means the cuisine type that is given the highest confidence. This result is shown in Table 4, where C denotes a case

[3]"Gyoza (餃子)" is a kind of dumplings and one of the most popular Chinese foods. It often appears in Chinese restaurant names in Japan.

Table 2: Distribution of estimation results by CM values.

	CM_D		CM_W		CM_I	
CM range	Correct	Incorrect	Correct	Incorrect	Correct	Incorrect
0.0 – 0.1	0	0	0	32	2	10
0.1 – 0.2	0	0	0	11	9	15
0.2 – 0.3	1	16	14	22	15	18
0.3 – 0.4	6	19	28	19	10	8
0.4 – 0.5	11	25	29	21	13	12
0.5 – 0.6	21	29	56	9	13	12
0.6 – 0.7	22	28	85	7	15	7
0.7 – 0.8	41	16	42	3	17	6
0.8 – 0.9	21	9	19	1	19	9
0.9 – 1.0	131	4	1	1	184	10
Total	254	146	274	124	297	103

Table 4: Estimation results by three CMs.

		CM_D / CM_W			
		I / I	I / C	C / I	C / C
CM_I	C	0	51	33	213
	I	85	10	8	0

C: correct, I: incorrect

when a cuisine type was correctly estimated and I denotes that it was not. The four columns with '/' denote the numbers of estimation results for CM_D and CM_W. For example, the C/I column denotes that estimation results based on the database were correct and those using the Web were incorrect, that is, the I/C and C/I columns mean that the two estimation results differed. The table shows that 102 of 400 restaurants corresponded to these cases, that is, either of the two estimation results was incorrect. It also shows that estimation results for 84 of the 102 (82%) restaurants became correct by the integration.

Two examples are shown for which the estimation results became correct by the integration. First, "Kaya (加屋)" is a restaurant name whose cuisine type is "Japanese-style pancake". Its cuisine type was correctly estimated by CM_W while it was incorrectly estimated as "Japanese pub" by CM_D. This was because, in Japanese, "Kaya (加屋)" has no special meaning associated with specific cuisine types. Thus, it is natural that its cuisine type was incorrectly estimated from the word and character distribution of the name. On the other hand, when Web pages about it were found, "Japanese-style pancake" co-occurs frequently in the obtained pages, and thus it was correctly estimated by CM_W. Second, "Tama-Sushi Imaike (玉寿司 今池)" is a restaurant name whose cuisine type is "Japanese restaurant". Its cuisine type was estimated correctly by CM_D while it was incorrectly estimated as "Japanese pub" by CM_W. CM_D was effective in this case because the part of "Sushi (寿司)" indicates a Japanese cuisine. No Web pages for it were found indicating its cuisine type correctly, and thus CM_W failed to estimate it.

5 Conclusion

Our aim is to acquire the attributes of an unknown word's concept from the user through dialogue. Specifically, we set restaurant cuisine type as the attribute to obtain and showed how to generate specific questions based on the estimated CM. We use two estimation methods: one based on the target database and the other on the Web. A more appropriate CM was generated in terms of its distribution and estimation accuracy by integrating these two CMs.

There is little prior research on obtaining and updating system knowledge through dialogues, with the notable exception of the knowledge authoring system of (Knott and Wright, 2003). Their system also uses the user's text input for constructing the system knowledge from scratch, which is used to generate simple stories. Our study is different in two points: (1) we focus on generating several kinds of questions because we use ASR, and (2) we try to handle unknown words, which will be stored in the target database to be used in future dialogues.

We should point out that these kinds of questions can be generated only when the types of unknown concepts are given. We assume the type of unknown concepts is already known and thus the attributes to be asked are also known. More specifically, we assume that the concept denoted by an unknown word is a restaurant name and its attributes are also known. The cuisine type has been estimated as one of the attributes. However, when the type is unknown, the system first needs to identify its attributes to ask. That is, the system first needs to ask about its supertype and then to ask about attributes that are typical for objects of this type. This issue needs to be addressed in order for the system to acquire arbitrary new concepts. This paper has shown the first step for obtaining concepts through dialogues by generating questions. Many issues remain in this field for future work.

References

Adam L. Berger, Vincent J. Della Pietra, and Stephen A. Della Pietra. 1996. A maximum entropy approach to natural language processing. *Computational Linguistics*, 22(1):39–71.

Isabelle Guyon and André Elisseeff. 2003. An introduction to variable and feature selection. *The Journal of Machine Learning Research*, 3:1157–1182.

David W. Hosmer Jr., Stanley Lemeshow, and Rodney X. Sturdivant. 2013. *Applied logistic regression*. Wiley. com.

Preethi Jyothi, Leif Johnson, Ciprian Chelba, and Brian Strope. 2012. Large-scale discriminative language model reranking for voice search. In *Proceedings of the NAACL-HLT 2012 Workshop: Will We Ever Really Replace the N-gram Model? On the Future of Language Modeling for HLT*, pages 41–49.

Gillian Kay. 1995. English loanwords in Japanese. *World Englishes*, 14(1):67–76.

Alistair Knott and Nick Wright. 2003. A dialogue-based knowledge authoring system for text generation. In *AAAI Spring Symposium on Natural Language Generation in Spoken and Written Dialogue, Stanford, CA*, pages 71–78.

Taku Kudo, Kaoru Yamamoto, and Yuji Matsumoto. 2004. Applying conditional random fields to Japanese morphological analysis. In *Proceedings of Conference on Empirical Methods in Natural Language Processing (EMNLP-2004)*, pages 230–237.

Helen Meng, P. C. Ching, Shuk Fong Chan, Yee Fong Wong, and Cheong Chat Chan. 2004. ISIS: An adaptive, trilingual conversational system with interleaving interaction and delegation dialogs. *ACM Transactions on Computer-Human Interaction*, 11(3):268–299.

Marius Pasca, Dekang Lin, Jeffrey Bigham, Andrei Lifchits, and Alpa Jain. 2006. Organizing and searching the World Wide Web of facts - step one: the one-million fact extraction challenge. In *Proceedings of the 21st National Conference on Artificial intelligence - Volume 2*, AAAI '06, pages 1400–1405. AAAI Press.

Hanchuan Peng, Fuhui Long, and Chris Ding. 2005. Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(8):1226–1238.

Lev Ratinov and Dan Roth. 2009. Design challenges and misconceptions in named entity recognition. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning*, pages 147–155.

Yasuhiro Takahashi, Kohji Dohsaka, and Kiyoaki Aikawa. 2002. An efficient dialogue control method using decision tree-based estimation of out-of-vocabulary word attributes. In *Proc. Int'l Conf. Spoken Language Processing (ICSLP)*, pages 813–816.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4*, pages 142–147.

Yiming Yang and Jan O Pedersen. 1997. A comparative study on feature selection in text categorization. In *ICML*, volume 97, pages 412–420.

Naoki Yoshinaga and Kentaro Torisawa. 2007. Open-domain attribute-value acquisition from semi-structured texts. In *Proceedings of the Workshop of OntoLex07 - From Text to Knowledge: The Lexicon/Ontology Interface*, pages 55–66.

Guo Dong Zhou and Jian Su. 2002. Named entity recognition using an HMM-based chunk tagger. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 473–480.