Interpretation of Implicit Conditions in Database Search Dialogues

Shun-ya Fukunaga[1] **Hitoshi Nishikawa**[1] **Takenobu Tokunaga**[1]
Hikaru Yokono[2] **Tetsuro Takahashi**[2]
[1]Tokyo Institute of Technology [2]Fujitsu Laboratories Ltd.
{fukunaga.s.ab@m, hitoshi@c, take@c}.titech.ac.jp
{yokono.hikaru, takahashi.tet}@jp.fujitsu.com

Abstract

Targeting the database search dialogue, we propose to utilise information in the user utterances that do not directly mention the database (DB) field of the backend database system but are useful for constructing database queries. We call this kind of information *implicit conditions*. Interpreting the implicit conditions enables the dialogue system more natural and efficient in communicating with humans. We formalised the interpretation of the implicit conditions as classifying user utterances into the related DB field while identifying the evidence for that classification at the same time. Introducing this new task is one of the contributions of this paper. We implemented two models for this task: an SVM-based model and an RCNN-based model. Through the evaluation using a corpus of simulated dialogues between a real estate agent and a customer, we found that the SVM-based model showed better performance than the RCNN-based model.

1 Introduction

The information that a dialogue system needs to extract from user utterances highly depends on its backend application. When being used as a natural language interface of a database system, the dialogue system should be able to extract pieces of information corresponding to the record fields of the database (DB) for constructing a query. There have been several attempts to extract this type of information from user utterances in database search dialogues. For instance, several studies (Raymond and Riccardi, 2007; Mesnil et al., 2015; Liu and Lane, 2016b) tried to extract values for the database field defined in the ATIS (The Air Travel Information System) corpus (Hemphill et al., 1990; Dahl et al., 1994) from the user utterances. The ATIS corpus includes a set of dialogues between users and an air travel system that were collected through the Wizard-of-Oz method. The tags that correspond to the backend database fields, e.g. departure city, arrival date, are annotated to the expressions in the user utterances. However, the utterances in real dialogues include information that does not always directly correspond to database fields but provides useful information for constructing database queries. It will be natural and more efficient if the dialogue system can utilise this type of information for retrieving the database. For instance, in real estate search dialogues, which is our target domain, the number of family members provides useful information for deciding the size of a house, but it is rarely a field of the real estate database as the number of family members is an attribute of the customer rather than that of houses. We call this type of information *implicit conditions* (Fukunaga et al., 2018).

To realise a dialogue system that can utilise the implicit conditions for the database search, we need to tackle the following two tasks.

(1) extracting pairs of a DB field and its value from user utterances that include the implicit conditions

(2) identifying the span in the utterance that represents the evidence for the DB field and value pair extraction

The task (1) provides useful information for efficiently constructing database queries from utterances that include the implicit conditions. The identified utterance span by the task (2) is helpful in constructing

Proceedings of the 27th International Conference on Computational Linguistics, pages 477–486
Santa Fe, New Mexico, USA, August 20-26, 2018.

clarification utterances by the system. As the task (1) requires a kind of inference to derive the DB field and value pair, the system interpretation is not necessarily correct. It will be safer and more natural to confirm the system interpretation with providing the evidence of that interpretation through clarification utterances. For instance, in the following dialogue fragment, the system inferred the DB field "number of rooms" and its value "one" from the user utterance. To ensure that this interpretation is correct, the system can add the evidence of the interpretation in the clarification utterance. This system utterance with the evidence is more natural than that without the evidence, i.e. "One bedroom place is large enough, isn't it?".

...

U: I will live alone.

S: As you live alone, one bedroom place is large enough, isn't it?

...

In such case, identifying that the span "live alone" is the evidence for extracting the DB field "number of rooms" and its value is worthwhile for constructing the clarification utterance.

We tackle the task (1) in two steps: extracting the DB field and extracting its value. In this paper, we first formalise the DB field extraction as a classification problem of user utterances into the DB fields. As we can derive multiple DB fields from a single utterance, the DB field extraction can be formalised as a multi-label classification. Furthermore, we solve the DB field classification together with the task (2), the evidence identification by adopting the method proposed by Lei et al. (2016). We put the DB field value extraction for the future work, as it requires the information of the database structure and its contents.

2 Related Work

The traditional pipeline for task-oriented dialogue systems consists of four modules: Natural Language Understanding (NLU), Dialogue State Tracking, Policy Learning and Natural Language Generation (Chen et al., 2017). The NLU task can be further divided into two tasks: intent detection and slot-filling. The intent detection classifies user utterances into the categories of user intention, e.g. request, question, inform and so on. The slot-filling extracts the semantic contents of user utterances in the form of slot-value pairs, e.g. the slot-value pairs From-Location="New York" and To-Location="Chicago" are extracted from "I'm going to go to Chicago from New York.". Regarding this general framework, our task corresponds to the slot-filling task. The slot-filling task can be formalised as a sequential labelling problem where each word in the target utterance is assigned an IOB tag of semantic slots (Ramshaw and Marcus, 1995). Recently, many studies adopt Recurrent Neural Network (RNN)(Mesnil et al., 2013; Yao et al., 2013; Mesnil et al., 2015; Vu et al., 2016; Jaech et al., 2016; Liu and Lane, 2016b; Liu and Lane, 2016a; Bapna et al., 2017) and Long Short-Term Memory (LSTM) (Yao et al., 2014; Hakkani-t et al., 2016) for the sequential labelling to achieve better performance. These methods, however, cannot identify semantic slots from the implicit condition which we are targeting, because they capture only explicitly mentioned semantic slots through the sequential labelling.

There are very few studies which address the task of predicting the users implicit condition in the task-oriented dialogue system. Celikyilmaz et al. (2012) proposed the method which predicts the user's preferred movie genre in the movie search domain. They estimate the movie genre that the user prefers from the user utterances that do not necessarily mention the movie genre. For instance, they estimate the comedy genre from the utterance "I wanna watch a movie that will make me laugh". Although the motivation of their work is almost the same as ours, their method estimates only a single attribute, i.e. the movie genre, and does not present the evidence for that estimate. Our method provides the evidence of the estimate at the same time and we deal with multiple attributes simultaneously.

In recent years, the end-to-end task-oriented dialogue systems which directly generate system utterances from user utterances are proposed. Eric et al. (2017) proposed an end-to-end dialogue system which accepts queries for the underlying knowledge base and returns the entities related to the users

Location	*Building*	*Facilities*	*Property*
available_railway_lines	building_age	room_facilities	property_type
walking_distance_to_a_station	floor_plan	air_conditioning	rent
nearest_station_facility	floor_area	storage	price
zone	room_placement_in_the_building	bathroom	conditions_for_rent
surrounding_facilities	room_size	kitchen	available_date
land_characteristics	building_structure	TV_and_Internet	target_demographic
distance_to_a_specific_place	sunlight		status
	building_facilities		appearance
	security_system		ownership
	number_of_storeys		available_discount
	number_of_households		subsidy
	renovation		certificate
			warranty

Table 1: Database (DB) field tags

utterance. Their system works without an explicit NLU module. As is often the case for end-to-end systems, their system does not concern the reason why the system returns a certain result to a given query, i.e it does not provide any explanation for the response. Given a user utterance including implicit conditions, our system makes a kind of inference to construct a query to the DB. As there is no guarantee that the system always can make a correct inference, we need to prepare the explanation for the system inference or clarification to the user. Such functions are necessary for natural and efficient dialogues. They are, however, difficult to implement in the current end-to-end framework.

3 Data and Task Setting

We use a Japanese dialogue corpus developed by (Takahashi and Yokono, 2017) in this work. The corpus includes dialogues between pairs of crowd workers who play a real estate agent and their customer each. The dialogues were collected through a keyboard chat system. The goal of the dialogues is finding a dwelling that fulfils the customer needs. The agent does not search in a real database but completes the dialogue when having acquired the necessary information for search. In each dialogue, a customer was assigned one of ten predefined profiles and was instructed to interact with the agent regarding their assigned profile. The customer profile was not open to the agent. An example profile looks like "You are moving to a new place from your current studio apartment to live with your long-standing boyfriend. On this occasion, you want to improve your cooking skills, and thus you prefer a place equipped with an easy-to-use kitchen with a multi-burner range."

The corpus includes 968 dialogues and 29,058 utterances consisting of 14,571 agent utterances and 14,487 customer utterances. The average number of utterances in a dialogue is 29.5.

We assigned the DB field tag shown in Table 1 to each utterance in our past research (Fukunaga et al., 2018). The DB field tag set was designed based on the search conditions that are available for one of the established search sites for real estate in Japan[1]. In addition to these 38 tags, we defined the Other tag that does not fit into any of the DB fields, expecting that utterances with the Other tag contain the implicit conditions. When the annotators assigned the Other tag to an utterance, they were instructed to describe its content in the free format.

The task of the present work is to categorise customer utterances with the Other tag into the 38 DB field tags and to identify the span representing the evidence of that classification at the same time. The target customer utterances are coupled with their preceding consecutive agent utterances[2]. The customer utterance with the Other tag sometimes contains only confirmation, e.g. "Yes, please" for the agent question "You live with your wife?". To collect necessary information for classification in such cases, we add the preceding agent utterances to the target customer utterance. We call this a target utterance chunk. There are 2,642 target utterance chunks in total.

[1] http://suumo.jp

[2] When the target customer utterance with the Other tag is preceded by customer utterances without the Other tag, they are also included in the chunk.

4 Methods

We propose two models for our current task: a Support Vector Machine (SVM)-based model and a Recurrent Convolutional Neural Network (RCNN)-based model.

4.1 SVM-based model

We use SVM with a linear kernel for the first model. We build an SVM binary classifier for each DB field tag that judges whether the input target utterance chunk includes the information for that DB field or not. Thus we have 38 classifiers in total. Given an utterance chunk, the final output of the system is the list of the DB field tags of which classifier returns a positive judgement. As features for SVM, we use a bag-of-words in the utterance chunk. We ran the Japanese morphological analyser MeCab[3] on all utterance chunks and extracted nouns, verbs, adjectives and adverbs appearing more than once in the corpus. We replaced the occurrences of numbers and proper nouns with their abstraction symbol NUM and PROP. The dimension of the feature vector is 1,730.

To identify the evidence span in the utterance chunk, we collect the words of which learnt weight exceeds a certain threshold.

4.2 RCNN-based model

We extend the method proposed by Lei et al. (2016) to solve our task. Given a product review text as input, their method estimates the user rating for each aspect of the product by using regression and specify the text span for the evidence of that regression result. Their system consists of two components: a neural network for regression (Encoder) and the other neural network for identifying the evidence (Generator). These two components are trained simultaneously. The encoder is trained through supervised training being given correct ratings while the generator is trained through unsupervised training. They designed a loss function for the generator so that it prefers a shorter and consecutive span for the evidence. Assuming that the generator correctly identifies the evidence span, the encoder uses only words in the evidence span to estimate the rating. They aim at improving the generator performance indirectly through improving the encoder performance that is trained by supervised training.

To adopt Lei et al. (2016)'s method for our current task, we redesign the loss function for the encoder. We use the cross-entropy instead of the square error for the encoder loss function, as our task is the binary classification while Lei et al. (2016)'s is the regression. Let $\tilde{y}_i = \mathrm{enc}_i(\boldsymbol{z}_i, \boldsymbol{x}) \in \{0, 1\}$ is the result of classification for the i-th DB field tag where $\boldsymbol{x}$ is the input word sequence with length m and $\boldsymbol{z}_i = (z_{i1}, z_{i2}, \ldots, z_{im})$ is the binary vector denoting if the words are selected as the evidence. Note that $\mathrm{enc}_i(\cdot)$ uses only words in $\boldsymbol{x}$ of which corresponding values in $\boldsymbol{z}_i$ is one, and $\boldsymbol{z}_i$ is the output of the generator. Our loss function $\mathcal{L}_i(\boldsymbol{z}_i, \boldsymbol{x}, y_i)$ is defined in equation (1),

$$\mathcal{L}_i(\boldsymbol{z}_i, \boldsymbol{x}, y_i) = -(y_i \log(\tilde{y}_i) + (1 - y_i) \log(1 - \tilde{y}_i)), \tag{1}$$

where y_i is the correct classification. The generator loss function is the same as the original as shown in equation (2).

$$\Omega_i(\boldsymbol{z}_i) = \lambda_1 \|\boldsymbol{z}_i\| + \lambda_2 \sum_{t=1}^{m} |z_{it} - z_{i(t-1)}|. \tag{2}$$

We set z_0 to zero. The first factor promotes the evidence span to be shorter and the second promotes the span to be consecutive. We have two hyper-parameters λ_1 and λ_2 to adjust the balance between these two factors. The total cost function for the i-th DB field is

$$\mathrm{cost}_i(\boldsymbol{z}_i, \boldsymbol{x}, y_i) = \mathcal{L}_i(\boldsymbol{z}_i, \boldsymbol{x}, y_i) + \Omega_i(\boldsymbol{z}_i). \tag{3}$$

[3] http://taku910.github.io/mecab/

5 Experiments

5.1 Data

We divided 986 dialogues of the corpus into ten sets with carefully keeping the distribution of the customer profiles uniform in each set. We use the nine sets for training and the rest one set for testing. We further extracted the utterance chunks as described in 3. We removed the chunks that include the target utterances at the dialogue management level, e.g. greeting, prompting and concluding utterances. The training set has 2,379 utterance chunks and the test set has 263 utterance chunks.

The user utterance in each target utterance chunk has been annotated with the Other tag and its content description in our past work as explained in 3. In the present work, we annotated each utterance chunk with the correct answer of the classification task by mapping its content description onto some of the DB field tags. One utterance chunk can be assigned with more than one tag. For instance, a content description, "the number of people to live", which is assigned to an utterance, "I will live alone.", is mapped onto floor_plan and floor_area. The evidence span for each assigned DB field tag is also annotated in the chunk. These annotation tasks are done by one of authors.

5.2 Evaluation Measures

We use precision, recall and F-measure for the evaluation of the DB field classification. To evaluate the evidence span identification, we calculate F-measure on words, and BLEU (Papineni et al., 2002) and ROUGE (Lin and Hovy, 2003). In the evaluation of the evidence span identification, we evaluated only cases where the DB field was correctly classified. Let binary vectors $\tilde{\boldsymbol{z}} = (\tilde{z_1}, \tilde{z_2}, \cdots, \tilde{z_m})$ and $\boldsymbol{z} = (z_1, z_2, \cdots, z_m)$ be the estimated and true evidence spans respectively, where the vector element one represents that the corresponding word is selected for constituting the evidence. Given $\tilde{\boldsymbol{z}}$ and $\boldsymbol{z}$, the F-measure on words can be calculated by equation (5).

$$\mathrm{TP} = \sum_{t=1}^{m} \tilde{z}_t z_t, \quad \mathrm{FP} = \sum_{t=1}^{m} \tilde{z}_t (1 - z_t), \quad \mathrm{FN} = \sum_{t=1}^{m} (1 - \tilde{z}_t) z_t \tag{4}$$

$$\mathrm{F} = \frac{2\mathrm{TP}}{2\mathrm{TP} + \mathrm{FP} + \mathrm{FN}} \tag{5}$$

For calculating BLEU and ROUGE, we define the sets of n-grams in the estimated and true evidence span as $\tilde{G_n}$ and G_n respectively in equation (6) and (7).

$$\tilde{G}_n = \left\{ \{j\}_{j=t}^{t+n-1} \middle| \prod_{j=t}^{t+n-1} \tilde{z}_j = 1,\ 1 \le t \le m - n + 1 \right\} \tag{6}$$

$$G_n = \left\{ \{j\}_{j=t}^{t+n-1} \middle| \prod_{j=t}^{t+n-1} z_j = 1,\ 1 \le t \le m - n + 1 \right\} \tag{7}$$

Using these n-gram sets, we can calculate BLEU and ROUGE as the geometric means of P_n and Q_n as shown in equation (8) and (9). We use uni-grams and bi-grams for calculating BLEU and ROUGE, i.e. $n \le 2$.

$$\mathrm{BLEU} = \left(\prod_{n=1}^{2} P_n \right)^{1/2}, \quad P_n = \begin{cases} \dfrac{|\tilde{G}_n \cap G_n|}{|\tilde{G}_n|} & (|\tilde{G}_n| > 0 \wedge n = 1) \\ \dfrac{|\tilde{G}_n \cap G_n| + 1}{|\tilde{G}_n| + 1} & (\text{otherwise}) \end{cases} \tag{8}$$

$$\mathrm{ROUGE} = \left(\prod_{n=1}^{2} Q_n \right)^{1/2}, \quad Q_n = \begin{cases} \dfrac{|\tilde{G}_n \cap G_n|}{|G_n|} & (|G_n| > 0 \wedge n = 1) \\ \dfrac{|\tilde{G}_n \cap G_n| + 1}{|G_n| + 1} & (\text{otherwise}) \end{cases} \tag{9}$$

	surrounding_facilities			floor_plan			floor_area		
Model	Precision	Recall	F-measure	Precision	Recall	F-measure	Precision	Recall	F-measure
SVM	**0.789**	**0.796**	**0.793**	**0.918**	**0.865**	**0.891**	**0.891**	**0.874**	**0.882**
RCNN	0.777	0.770	0.773	0.881	0.856	0.868	0.873	0.864	0.868

Table 2: Result of the DB field classification

	surrounding_facilities			floor_plan			floor_area		
Model	F-measure on words	BLEU	ROUGE	F-measure on words	BLEU	ROUGE	F-measure on words	BLEU	ROUGE
SVM_1	0.514	0.768	0.467	0.440	**0.808**	0.364	0.420	0.763	0.345
SVM	**0.530**	**0.787**	0.552	**0.533**	0.802	0.467	**0.507**	**0.773**	0.462
RCNN	0.458	0.534	**0.576**	0.452	0.625	**0.475**	0.436	0.521	**0.485**

Table 3: Result of the evidence span identification

In equation (8) and (9), we add a smoothing constant. We do not introduce the brevity penalty into BLEU as we also use ROUGE in this evaluation.

5.3 Evaluation Settings

Due to the limitation of the data size, we evaluate the three DB fields: surrounding_facilities, floor_plan and floor_area, which are the three most dominant tags. The numbers of utterance chunks in the training set that are assigned these tags are 1,033 for surrounding_facilities, 974 for floor_plan and 964 for floor_area.

The RCNN-based model has two hyper-parameters λ_1 and λ_2 that control the balance between the span length and the constraints on the word sequence. To decide the values for these hyper-parameters, we randomly chose 200 utterance chunks from the training set of surrounding_facilities to make a development set, and trained the model with the rest of the training set and evaluated it with the development set. As a result, we found that $\lambda_1 = 0.021$ and $\lambda_2 = 0.003$ made the best performance on the development set. We use these hyper-parameter values for the other two DB fields. We also used the parameter values of the neural networks learnt in the development set as the initial parameter values. We used the existing word embeddings made from the Japanese Wikipedia articles[4]. The dimension size is 200.

The SVM-based model also has a hyper-parameter, i.e. the threshold weight for the evidence span identification. The words that have a higher weight than the threshold constitute the evidence. We used the same development set as the RCNN-based model to estimate this threshold. The word weight is normalised by equation (10). The estimated threshold was 0.58.

$$\hat{w} = \frac{w - w_{\min}}{w_{\max} - w_{\min}} \tag{10}$$

5.4 Experimental Results and Discussion

Table 2 and Table 3 shows the results of the DB field classification and the evidence span identification. Table 3 also includes the result of the SVM-based model that adopts a single word with the highest weight for the evidence (SVM_1). In both tasks, the SVM-based model outperformed the RCNN-based model.

DB field classification

Table 2 indicates that the SVM-based model is superior to the RCNN-based model in all evaluation measures. The RCNN-based model is trained to classify the utterance into a DB field by utilising only the fragments in the utterance identified as the classification evidence. That means the performance of the DB field classification depends on that of the evidence span identification. As shown in Table 3,

[4] http://www.cl.ecei.tohoku.ac.jp/~m-suzuki/jawiki_vector/

DB field	surrounding_facilities		floor_plan		floor_area	
overlap	TP	FN	TP	FN	TP	FN
Yes	73	17	86	10	75	11
No	14	9	3	5	14	3

Table 4: Categorisation of cases based on word overlap between the correct and predicted evidence spans

the word-wise F-measure of the evidence span identification is less than 0.5. This suggests that the RCNN-based model has to solve the DB field classification by using insufficient information, i.e. wrong evidence span.

To investigate the relation between the classification performance and the evidence spans, we counted the number of true-positive (TP) and false-negative (FN) cases in the result of the DB field classification by the RCNN-model. Table 4 shows the breakdown of the cases according to whether the predicted and correct evidence spans share at least one word. The row "Yes" shows the numbers of cases where at least one word is shared by both predicted and correct evidence spans and "No" shows that for no word overlap. For instance, in surrounding_facilities, 73 out of 90 (81%) "Yes" cases are correct (TP) while 14 out of 23 (60%) "No" cases are correct. This tendency is the same as in the other two DB fields. Table 4 suggests that the performance of the evidence span identification has a significant impact on that of the DB field classification in the RCNN-model.

However, Table 4 also shows that there are cases where the RCNN-model can correctly classify the DB fields even though it fails to identify the correct evidence spans. We investigated such 14 cases for the surrounding_facilities field (the cell intersecting "No" and "TP") to find words that are useful for the field classification but not included in the evidence spans. For instance, the word "near" appears in the positive examples twice as frequent as in the negative examples in our training data. This word can be a clue for the field classification although not being included in the evidence span. To improve the performance of the DB field classification, we need to extend the current model to incorporate such useful information on top of the information from the evidence spans.

We further investigated the individual error cases of the surrounding_facilities classification. We found errors due to the annotation inconsistency in 10 out of 25 FP cases in the RCNN-based model result and 11 out of 24 FP cases in the SVM-based model result. For instance, an utterance chunk in FP cases, "Do you have any preferences for location? — I prefer a suburb.", is annotated as a negative example, while the utterance chunk which has almost the same content, "What kind of property are you looking for? — I plan to move to a suburb in future.", is annotated as a positive example. When we assign the DB field to the utterance with the Other tag we regarded only the description for the Other tag, without referring to the original utterances. In the above example, the content description of the former utterance chunk is "rough area", while that of the latter is "living environment". Although the content of the utterances is the same, the difference of their description caused such inconsistent field assignment. Our future work includes implementing more solid annotation guideline that enables stable and consistent assignment of the correct DB field.

Evidence span identification

Table 3 shows that the SVM-based model is superior to the RCNN-based model in all DB fields and measures except for ROUGE.

Table 5 shows the average number of words and spans in the predicted and correct evidence of the evidence span identification.

As shown in Table 5, the number of words in the correct evidence spans is more than two on average. Therefore the SVM-based model selecting only one word as the evidence (SVM_1 in Table 3) cannot select enough words for the evidence. Setting an appropriate threshold on the learnt weight of each word (feature) improves both ROUGE and BLUE values. This implies that the learnt word weight can be a clue to choose words to constitute the evidence.

We formalised the evidence span identification as a binary classification of whether each word in the

	surrounding_facilities			floor_plan			floor_area		
	#words	#spans	#w/#s	#words	#spans	#w/#s	#words	#spans	#w/#s
SVM	1.60	1.54	1.04	2.48	2.33	1.07	2.49	2.32	1.07
RCNN	3.21	2.39	1.34	2.97	2.06	1.44	3.57	2.12	1.69
Correct Answer	3.42	1.23	2.78	4.58	1.20	3.81	4.58	1.20	3.81

Table 5: Size of evidence spans

			Correct Answer	
			Positive	Negative
Prediction	Positive	SVM	102	64
		RCNN	92	144
	Negative	SVM	193	2,353
		RCNN	165	2,038

Table 6: Confusion matrix of the evidence span identification

utterance chunks belongs to the evidence or not. Table 6 shows the confusion matrix of the number of words constituting the evidence for the surrounding_facilities field. This matrix indicates that there are more false-negative cases than false-positive cases in both models. A dominant word type in the false-negative cases is Japanese particles, e.g. ga, ni, which indicate case marking. In other words, both models tend to skip these particles and to select the content words such as nouns and verbs. Although the particles are not important features of the classification since they appear in any class, it is desirable for dialogue systems to include them into the evidence so that they are used in the later stage of response generation, e.g. generating clarification utterances.

Through the investigation of the individual extracted evidence spans for the surrounding_facilities field by the RCNN-based model, we found that the model tends to extract the word just after the question mark "?" for all test cases. The question mark is the last word of the question by the real estate agent in most cases. The word "safety" which is the strong clue for surrounding_facilities appears just after the question mark at a rate of 41.3%. The model seems to have learnt this collocation. However, there are only 40.4% of test cases where the word just after a question mark belongs to the evidence span. This over-tuning degrades the performance of the evidence span identification.

The RCNN-based model is inferior to the SVM-based model in both tasks. Probably we need to have more data to train the neural network-based model; we used 2,379 utterance chunks for training classifiers while Lei et al. (2016) used about 80,000 to 90,000 reviews in the experiment.

6 Conclusion

Targeting the database search dialogue, we proposed to utilise information in the user utterances that do not directly correspond to the DB field of the backend database system. We call this kind of information *implicit conditions*. We formalised the interpretation of the implicit conditions as classifying user utterances into the related DB field while identifying the evidence for that classification at the same time. We implemented two models: an SVM-based model and an RCNN-based model. Through evaluation experiments on a corpus of simulated dialogues between a real estate agent and a customer, we found that the SVM-based model showed better performance than the RCNN-based model. This implies that the size of the corpus is too small to train the latter model. Furthermore, the error analysis on the DB field classification revealed that some errors occurred due to the annotation inconsistency. Our future work includes implementing more solid annotation guideline that enables stable and consistent assignment of the correct DB field.

While this paper focused on only finding DB field and its evidence from the utterance, we need to further extract the value for the field in order to generate database queries from the utterances.

References

Ankur Bapna, Gokhan Tur, Dilek Hakkani-tur, and Larry Heck. 2017. Sequential Dialogue Context Modeling for Spoken Language Understanding. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue (SIGDIAL 2017)*, pages 103–114.

Asli Celikyilmaz, Dilek Hakkani-tür, and Gokhan Tur. 2012. STATISTICAL SEMANTIC INTERPRETATION MODELING FOR SPOKEN LANGUAGE UNDERSTANDING WITH ENRICHED SEMANTIC FEATURES. In *Spoken Language Technology Workshop (SLT), 2012 IEEE*, pages 216–221.

Hongshen Chen, Xiaorui Liu, Dawei Yin, and Jiliang Tang. 2017. A survey on dialogue systems: Recent advances and new frontiers. *SIGKDD Explor. Newsl.*, 19(2):25–35, November.

Deborah A. Dahl, Madeleine Bates, Michael Brown, William Fisher, Kate Hunicke-Smith, David Pallett, Christine Pao, Alexander Rudnicky, and Elizabeth Shriberg. 1994. Expanding the scope of the ATIS task: The ATIS-3 corpus. In *Proceedings of the Workshop on Human Language Technology*, HLT '94, pages 43–48.

Mihail Eric, Lakshmi Krishnan, Francois Charette, and Christopher D Manning. 2017. Key-Value Retrieval Networks for Task-Oriented Dialogue. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue (SIGDIAL 2017)*, pages 37–49.

Shun-ya Fukunaga, Hitoshi Nishikawa, Takenobu Tokunaga, Hikaru Yokono, and Tetsuro Takahashi. 2018. Analysis of implicit conditions in database search dialogues. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, pages 2741–2745.

Dilek Hakkani-t, Gokhan Tur, Asli Celikyilmaz, Yun-nung Chen, Jianfeng Gao, Li Deng, and Ye-yi Wang. 2016. Multi-Domain Joint Semantic Frame Parsing Using Bi-Directional RNN-LSTM. In *INTERSPEECH-2016*, pages 715–719.

Charles T. Hemphill, John J. Godfrey, and George R. Doddington. 1990. The ATIS spoken language systems pilot corpus. In *Proceedings of the Workshop on Speech and Natural Language*, HLT '90, pages 96–101.

Aaron Jaech, Larry Heck, and Mari Ostendorf. 2016. Domain Adaptation of Recurrent Neural Networks for Natural Language Understanding. In *INTERSPEECH-2016*, pages 690–694.

Tao Lei, Regina Barzilay, and Tommi Jaakkola. 2016. Rationalizing neural predictions. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP 2016)*, pages 107–117.

Chin-Yew Lin and Eduard Hovy. 2003. Automatic evaluation of summaries using n-gram co-occurrence statistics. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pages 71–78.

Bing Liu and Ian Lane. 2016a. Attention-Based Recurrent Neural Network Models for Joint Intent Detection and Slot Filling. In *INTERSPEECH-2016*, pages 685–689.

Bing Liu and Ian Lane. 2016b. Joint Online Spoken Language Understanding and Language Modeling with Recurrent Neural Networks. In *Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL 2016)*, pages 22–30.

Grégoire Mesnil, Xiaodong He, Li Deng, and Yoshua Bengio. 2013. Investigation of Recurrent-Neural-Network Architectures and Learning Methods for Spoken Language Understanding. In *INTERSPEECH-2013*, pages 3771–3775.

Grégoire Mesnil, Yann Dauphin, Kaisheng Yao, Yoshua Bengio, Li Deng, Dilek Hakkani-tur, Xiaodong He, Larry Heck, Gokhan Tur, Dong Yu, and Geoffrey Zweig. 2015. Using Recurrent Neural Networks for Slot Filling in Spoken Language Understanding. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 23(3):530–539.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318.

Lance Ramshaw and Mitch Marcus. 1995. Text chunking using transformation-based learning. In *Third Workshop on Very Large Corpora*.

Christian Raymond and Giuseppe Riccardi. 2007. Generative and discriminative algorithms for spoken language understanding. In *Eighth Annual Conference of the International Speech Communication Association*, pages 1605–1608.

Tetsuro Takahashi and Hikaru Yokono. 2017. Two persons dialogue corpus made by multiple crowd-workers. In *Proceedings of the 8th International Workshop on Spoken Dialogue Systems (IWSDS 2017)*. 6 pages.

Ngoc Thang Vu, Pankaj Gupta, Heike Adel, and Hinrich Schutze. 2016. BI-DIRECTIONAL RECURRENT NEURAL NETWORK WITH RANKING LOSS FOR SPOKEN LANGUAGE UNDERSTANDING. In *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*, pages 6060–6064.

Kaisheng Yao, Geoffrey Zweig, Mei-yuh Hwang, Yangyang Shi, and Dong Yu. 2013. Recurrent Neural Networks for Language Understanding. In *INTERSPEECH-2013*, pages 2524–2528.

Kaisheng Yao, Baolin Peng, Yu Zhang, Dong Yu, Geoffrey Zweig, and Yangyang Shi. 2014. SPOKEN LANGUAGE UNDERSTANDING USING LONG SHORT-TERM MEMORY NEURAL NETWORKS. In *Spoken Language Technology Workshop (SLT)*, pages 189–194.