

論文 / 著書情報
Article / Book Information

Title	Analysis of Implicit Conditions in Database Search Dialogues
Authors	Shun-ya Fukunaga, Hitoshi Nishikawa, Takenobu Tokunaga, Hikaru Yokono, Tetsuro Takahashi
Citation	Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018), pp. 2741-2745
Pub. date	2018, 5

Analysis of Implicit Conditions in Database Search Dialogues

Shun-ya Fukunaga Hitoshi Nishikawa
Takenobu Tokunaga

Tokyo Institute of Technology

fukunaga.s.ab@m.titech.ac.jp, {hitoshi, take}@c.titech.ac.jp

Hikaru Yokono Tetsuro Takahashi

Fujitsu Laboratories Ltd.

{yokono.hikaru, takahashi.tet}@jp.fujitsu.com

Abstract

This paper reports an annotation to a corpus of database search dialogues on real estate, and the analysis on implicit information in the utterances for constructing database queries. Two annotators annotated 50 dialogues with a set of database field tags, resulting in a high inter-annotator agreement (Cohen's $\kappa=0.79$), and the analysis revealed that 10% of the utterances included non-database-field information. We further investigated these utterances to find that more than 93% of them included useful information for figuring out search conditions, which we call the *implicit condition*. The contribution of this paper is to present the existence and importance of the implicit conditions in the database search dialogues and both qualitative and quantitative analysis of them. Our corpus can provide a fundamental language resource for constructing a dialogue system which can utilise the implicit conditions. The paper concluded with possible approaches to achieve our long-term goal, extracting the implicit conditions in the database search dialogues and utilising them to construct queries.

Keywords: database search dialogue, implicit condition, corpus annotation

1. Introduction

The information that a dialogue system should extract from user utterances highly depends 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 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, 2016) 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 corresponding to the backend database fields (e.g. departure city, arrival date, etc.) were annotated to the expressions in the user utterances. However, the utterances in real dialogues include information that does not always directly correspond to a database field but provides useful information for constructing database queries. It will be more efficient and natural 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. The number of family members is an attribute of the customer rather than that of houses. We call this type of information the *implicit condition*.

Our long-term goal is to establish a method that is capable of extracting implicit conditions from the user utterances in the database search dialogue and utilising them to construct queries. As the first step in achieving this goal, this paper reports a corpus construction in which we annotated user utterances with a set of fields of a real estate database. We further analysed the utterances that were not assigned a database field and found that the majority of those utter-

ances included the implicit conditions.

2. Related Work

There have been several attempts to annotate semantic information in database search dialogues. He and Young (2005) annotated the domain-specific lexical classes (e.g. “city name”, “airport name”) and concepts (e.g. ARRIVE, FROMLOC) in the ATIS corpus. The annotated tags were derived from the ATIS database schema. They considered only information corresponding to the database fields. The implicit conditions are out of their scope.

Asri et al. (2017) annotated a frame structure to each utterance in dialogues for searching package tours. A frame consists of a dialogue act and a set of slot-value pairs representing the content of the dialogue act. The most slots correspond to the database field, but some do not. These non-database-field slots mainly describe relations to other frames. In this respect, they are meta-level information than the same level information as the filed-slot information; therefore they are different from the implicit condition.

Dinarelli et al. (2009) annotated a semantic frame of FrameNet (Baker, 2012) in task-oriented dialogues. Since the FrameNet frames are designed for general purpose, the annotated frame are not necessarily useful for the database search task. In contrast, we annotated the information that is relevant for constructing database queries regardless whether it corresponds to the database field or not.

The above three studies annotated only predefined tags in the dialogues, i.e. they did not explore the information that does not fit into the predefined tags even though it is useful for the task. In contrary, Mitsuda et al. (2017) explored what kind of information can be extracted and also should be extracted from a chat-oriented dialogue system. We follow the same line, but we focus on exploring the useful information for constructing queries in the database search dialogues even though it does not directly correspond to a database field.

The most similar annotation scheme to ours is the one of the TourSG corpus (Kim et al., 2015) used in the Dialog State Tracking Challenge (DSTC) 4 and 5 (Kim et al., 2016b; Kim et al., 2016a). In this corpus, dialogue states are annotated in the form of slot-value pairs to each segment consisting of multiple utterances in touristic domain dialogues. A special slot called “INFO” is prepared for the information in the segment that does not directly correspond to any predefined regular slots. Their scheme is similar to ours in this respect, i.e. considering the information not corresponding to predefined slots. However, they did not analyse the details of such information nor its relation to the predefined slots.

3. Target Corpus: A Real Estate Corpus

We annotate a Japanese dialogue corpus developed by Takahashi and Yokono (2017). The corpus includes 986 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 house or an apartment room that fulfils the customer needs. The agent does not search in a real database but completes the dialogue when having acquired necessary information for search.

In a 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.”

4. Annotation

4.1. Database Field Tags

Referring to the search conditions in the real estate search site SUUMO¹, we defined 38 database (DB) field tags as shown in Table 1. We adopted the search conditions of SUUMO as it is one of the established search sites for real estate in Japan. In addition to these 38 tags, we defined the Other tag that does not fit into any of the database fields, expecting that utterances with the Other tag contain the implicit conditions.

4.2. Annotation Guidelines

The annotator annotated the corpus following the guidelines below.

- The minimal annotation unit is an utterance, i.e. the annotator assigns the DB field tag(s) to a whole utterance instead of a part of the utterance. We expect that this scheme reduces annotator’s load.
- The annotator assigns the DB field tag(s) to an utterance by referring to only the target utterance and its preceding utterances; the annotator cannot refer to the succeeding context.

Location

- available_railway_lines
- walking_distance_to_a_station
- nearest_station_facility
- zone
- surrounding_facilities
- land_characteristics
- distance_to_a_specific_place

Building

- building_age
- floor_plan
- floor_area
- room_placement_in_the_building
- room_size
- building_structure
- sunlight
- building_facilities
- security_system
- number_of_storeys
- number_of_households
- renovation

Facilities

- room_facilities
- air_conditioning
- storage
- bathroom
- kitchen
- TV_and_Internet

Property

- property_type
- rent
- price
- conditions_for_rent
- available_date
- target_demographic
- status
- appearance
- ownership
- available_discount
- subsidy
- certificate
- warranty

Table 1: Database (DB) field tags

- When the annotator cannot find any appropriate DB field tag for the utterance, they annotate the Other tag and describe its content in the free format.
- When an utterance mentions to multiple database fields, the multiple tags concatenated with “+” are assigned. When an utterance is ambiguous in DB field tags, the candidate tags concatenated with “/” are assigned.

Table 2 shows an example of an annotated dialogue, which is a translation of the original Japanese dialogue. The speaker (SP) “A” denotes the real estate agent and “C” denotes the customer. The content description for the Other tag is shown in the parentheses. In utterance 2, the customer states that they are looking for a room close to their workplace, and they live alone. Therefore the distance_to_a_specific_place tag and the Other tag are assigned to the utterance. The annotator has added “number of persons (to live in the room)” to the Other tag as its content. Since the utterance includes both tags, they are concatenated with “+”. Utterance 10 mentions “one-bedroom apartment” (floor_plan), “rent” (rent) and “location”. Since the expression “location” has two interpretations: surrounding_facilities and zone, they are conjoined by “/” and put together with other two tags with “+”.

4.3. Agreement Analysis

Two annotators A and B annotated 50 dialogues in the corpus introduced in the previous section with the DB field tags defined in Table 1. Five dialogues were randomly selected from the dialogues of each customer profile to constitute the 50 dialogues of the annotation target. The total number of utterances is 1,305. As we focus on the content-level analysis of the dialogues, we excluded the utterances at the dialogue management level, e.g. greeting utterances as utterance 1 in Table 2, prompting utterances like “Do you have any other requests?” and concluding utterances at the end of the dialogue such as “Thanks for your coming

¹<http://suumo.jp>

SP utterance	assigned tag (content of the Other tag if any)
1 A What kind of property are you looking for?	Other(general)
2 C I am looking for a room for a single life that is close to my workplace.	distance_to_a_specific_place+Other(number of persons)
3 A Where is your workplace?	Other(workplace)
4 C Ikebukuro.	Other(workplace)
5 A How do you commute?	Other(transportation)
6 C I prefer a walking distance place.	Other(transportation)
7 A How much do you pay for a place within walking distance of Ikebukuro?	distance_to_a_specific_place+rent
8 C Around 100,000 yen.	rent
9 A Do you have any preference in the floor plan?	floor_plan
10 C I prefer a one-bedroom apartment. I put importance on the location and rent.	floor_plan+(surrounding_facilities/zone)+rent
11 A Do you need a parking lot?	building_facilities
12 C Yes, I need a parking lot. I would like to have a bicycle parking space as well.	building_facilities
13 A Do you have any particular request on facilities?	building_facilities/room_facilities
14 C Since I come home late, I need a tight security system and prefer a place populous and bright even at night.	security_system+Other(living environment)

Table 2: Annotation example

annotator A		annotator B	
Other	176	Other	152
building_facilities	128	building_facilities	121
property_type	118	rent	107
rent	105	property_type	100
floor_type	94	floor_type	98

Table 3: Number of frequently assigned tags (top 5)

annotator A		annotator B	
number of persons	62	number of persons	56
living environment	38	transportation	14
resident type	8	purpose of the property	13
accessibility	8	family structure	13
transportation	7	ambient noise	12

Table 4: Content descriptions of the Other tag

to our agency.” We used remaining 1,194 utterances for the following analysis. The average number of the target utterances in a dialogue is 23.9, and the average length of the utterances is 19.8 characters in Japanese.

Out of 1,194 utterances, 964 utterances (80.7%) were assigned completely the same tag(s) by both annotators; that measures 0.79 in Cohen’s κ (Cohen, 1960). We have a quite high inter-annotator agreement, concluding that we have no serious problem in our annotation scheme. In remaining 230 utterances that were assigned different tags across the annotators, there were 77 cases (33.5%) where an annotator assigned the tags in Table 1 but the other assigned the Other tag to the utterance. As we will discuss in the next section, such inconsistency comes from the fact that there are many utterances with the Other tag containing information tightly related to the search condition, i.e. it is difficult to decide they should be assigned the DB field tags or the Other tag. In Table 2, for instance, annotator A assigned the Other tag with the description “living environment” to utterance 14 for the expression “a place populous and bright even at night”. In contrary, annotator B assigned the sur-

	annotator A		annotator B	
	type	token	type	token
implicit conditions	21	165	11	145
customer’s profile	17	114	9	121
vague requests	4	51	2	24
no relevant DB fields	4	11	2	7
potential DB fields	1	4	1	4
no relation to DB	3	7	1	3
total	25	176	13	152

Table 5: Classification of the content of the Other tags

rounding_facilities tag to the same utterance for the same expression. In reality, this expression does not directly correspond to the available database fields, but annotator B thought it corresponded to the DB field tag. This is because we can infer necessary values for the surrounding_facilities field through further clarification exchanges if necessary. To investigate this hypothesis quantitatively, we extracted utterances such that they are assigned a single tag by both annotators and these tags are inconsistent; one is the Other tag, and the other is a DB field tag. Out of such 32 utterances, 24 DB field tags (75.0%) can be easily derived from the content description of the Other tag through inference based on one of the authors subjective judgement. The annotator who assigned a DB field tag to these utterances presumably used some inference to derive the DB field tag regardless consciously or unconsciously.

5. Discussion

Table 3 shows the five most frequently assigned tags by each annotator. We counted multiple assigned tags to an utterance individually. The annotators assigned 1,464 (annotator A) and 1,467 (annotator B) tags respectively. Table 3 shows that the most frequently assigned tag is the Other tag that accounts for 12.0% (annotator A) and 10.4% (annotator B), indicating that the amount of information that does not directly correspond to the database fields is not negligible.

Table 4 shows the five most frequently mentioned content of the Other tag. Although the annotators were instructed to be consistent in surface expressions of the content description, as they were described in the free format, the expressions across annotators might be different for semantically the same content. The total numbers of the content description in type by the annotators are 25 (annotator A) and 13 (annotator B).

The utterances with the Other tag can be classified into two groups regarding if they include useful information for constructing a database search query.

The first group, which we introduced as the *implicit condition* in section 1, includes the information that does not directly match the database field but provides a clue for the field information through some inference. Observing the content descriptions of the Other tags, we can further categorise implicit conditions into two subgroups: customer's profiles and vague requests on the property. For instance, a customer's utterance "We are planning to live with two of us." describes the number of persons to live in the property, which is categorised to the customer's profile. This information can provide useful information for deciding room_size and floor_plan because an agent can easily infer the number and the size of rooms required for living with two persons. Also, the customer's profile of "workplace" and "transportation" described by utterances from 3 to 6 in Table 2 is useful information for deciding zone and available_railway_lines. In the collected corpus, we observe that the agent often asks the customer's profile. This would be because the agent preferred a question that draws out the information for multiple search fields at a time instead of a sequence of questions asking individual search field values one by one.

The second subgroup of the implicit conditions concerns customer's vague requests, which are related to multiple database fields. For example, a customer's utterance "I prefer a safe neighbourhood.", which corresponds to "living environment" in Table 4, is categorised to this subgroup, since the safeness of the neighbourhood is decided by both zone and surrounding_facility. Also, a customer's utterance "I prefer a convenient place for transportation." is related to all database fields about transportation such as available_railway_lines, walking_distance_to_a_station and nearest_station_facility. Being triggered by these kinds of utterances, we might be able to infer the field values through further clarification exchanges.

Table 5 summarises the counts of the implicit conditions and its subgroups. It also shows the counts of utterances that provide no information for the database fields. The table indicates that the majority of the utterances with the Other tag includes the implicit condition, i.e. the useful information to construct database search queries but not having corresponding database fields. This suggests insufficiency of the database fields we adopted, but the further investigation reveals that this is not the case. We point out two reasons of difficulty in defining the implicit conditions as the database fields.

The first reason concerns the mismatch in the information type between user's needs and the database. The real estate database stores information of real properties, while

the user's needs realised as queries are tightly related to their objectives of a search and their backgrounds, which include diverse user's attributes. For instance, a family structure affects many database fields such as zone, surrounding_facilities and room_size, but the family structure is defined by many attributes such as the number of family members, their gender, their age and individual working style. It is impractical to add these attributes constituting the family structure in the real estate database. There is a gap in type between user's attributes and the real estate attributes.

Related to the first one, the second reason is that mapping from user's needs to the conditions on the database fields involves the user's evaluation of the values of the database fields. For instance, the implicit condition "living environment", which is the second most frequent by annotator A in Table 4, can be fulfilled differently in terms of the field values of properties depending on the user. In other words, a good property for a user is not always relevant for other users. There is a gap in evaluation of the field values by individual users. These reasons can apply to the database systems in general not being restricted the real estate database. The implicit condition triggers inference for bridging these kinds of gaps. Our corpus can provide a fundamental language resource for constructing a dialogue system which can utilise the implicit conditions.

We also have a minor group of utterances that does not provide any information regarding the current database fields. However, a few of them can be a potential field of the database ("potential DB fields" in Table 5). We found one such case where the customer asked if a preview of the property was available. The database we used does not include this information, but it would be useful for customers and is an attribute of real properties. Thus it can be a field of the database.

6. Conclusion

This paper reported an annotation to a corpus of database search dialogues on the real estate, and the analysis on implicit information in the utterances for constructing database queries. Two annotators annotated 50 dialogues with a set of database field tags, resulting in a high inter-annotator agreement (Cohen's $\kappa=0.79$). The analysis revealed that 10 % of the utterances included the information that did not directly match the database fields. We further investigated these utterances to find that more than 93% of them included useful information for constructing search queries by providing the database filed with a certain value, which we call the *implicit condition*. To realise a versatile dialogue system for database search that can naturally interact with users, we need to establish a method to extract the database field information from the implicit condition utterances. Our corpus can provide a fundamental language resource for constructing such a dialogue system. We plan to make it available for scientific communities.

Toward implementation of dialogue systems which can utilise implicit conditions, we need to challenge at least the following three tasks.

- (1) identification of the utterances including the implicit condition

- (2) interpretation of the implicit condition in the utterances as the corresponding database fields and their values
- (3) identification of the span in the utterance representing the implicit condition

We will tackle these tasks with the following approaches. For the task (1), we can construct a binary classifier for deciding if an utterance includes the implicit condition or not. The annotated corpus described in this paper is usable for the training data of the classifier.

For the task (2), we plan to annotate the corresponding database fields to the implicit conditions in the utterances manually. As the number of database fields is limited, we believe this manual annotation is feasible according to our preliminary annotation trial. Being trained on this additional annotation results, a classifier can be constructed which classifies the implicit condition to the database fields. As an implicit condition can correspond to multiple database fields, e.g. “number of persons” can be classified to both `room_size` and `floor_plan`, this classifier should be a multi-label classifier.

To confirm that the interpretation of the system is correct, it might need clarification to the user. For constructing a clarification utterance by the system, the surface expression corresponding to the implicit condition in the user utterance is useful. For instance, having obtained the number of persons in the family, the system interprets it as `floor_plan` and might confirm the interpretation by saying “Then, a four-room apartment is necessary for three persons, isn’t it?”. The task (3) is, therefore, important for natural interaction. For this task, we plan to use the machine learning techniques which can provide the rationales for the output (Lei et al., 2016). We aim at identifying the span in the utterance (rationale) for the interpretation of the implicit condition (output).

7. Bibliographical References

- Asri, L. E., Schulz, H., Sharma, S., Zumer, J., Harris, J., Fine, E., Mehrotra, R., and Suleman, K. (2017). Frames: a corpus for adding memory to goal-oriented dialogue systems. In *Proceedings of the 18th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL 2017)*, pages 207–219.
- Baker, C. F. (2012). FrameNet, current collaborations and future goals. *Language Resources and Evaluation*, 46(2):269–286.
- Cohen, J. (1960). A coefficient of agreement for nominal scales. *Educational and psychological measurement*, 20(1):37–46.
- Dahl, D. A., Bates, M., Brown, M., Fisher, W., Hunicke-Smith, K., Pallett, D., Pao, C., Rudnicky, A., and Shriberg, E. (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.
- Dinarelli, M., Quarteroni, S., Tonelli, S., Moschitti, A., and Riccardi, G. (2009). Annotating spoken dialogs: From speech segments to dialog acts and frame semantics. In *Proceedings of the 2Nd Workshop on Semantic Representation of Spoken Language, SRSL ’09*, pages 34–41.
- He, Y. and Young, S. (2005). Semantic processing using the hidden vector state model. *Computer speech & language*, 19(1):85–106.
- Hemphill, C. T., Godfrey, J. J., and Doddington, G. R. (1990). The ATIS spoken language systems pilot corpus. In *Proceedings of the Workshop on Speech and Natural Language, HLT ’90*, pages 96–101.
- Kim, S., D’Haro, L. F., Banchs, R. E., Williams, J. D., and Henderson, M., (2015). *Dialog State Tracking Challenge 4 Handbook v.3.0*. http://www.colips.org/workshop/dstc4/Handbook_DSTC4.pdf.
- Kim, S., D’Haro, L. F., Banchs, R. E., Williams, J. D., and Henderson, M. (2016a). The fourth dialog state tracking challenge. In *Proceedings of the 7th International Workshop on Spoken Dialogue Systems (IWSDS 2016)*.
- Kim, S., D’Haro, L. F., Banchs, R. E., Williams, J. D., Henderson, M., and Yoshino, K. (2016b). The fifth dialog state tracking challenge. In *Proceedings of the 2016 IEEE Workshop on Spoken Language Technology (SLT)*.
- Lei, T., Barzilay, R., and Jaakkola, T. (2016). Rationalizing neural predictions. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP 2016)*, pages 107–117.
- Liu, B. and Lane, I. (2016). 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.
- Mesnil, G., Dauphin, Y., Yao, K., Bengio, Y., Deng, L., Hakkani-Tur, D., He, X., Heck, L., Tur, G., Yu, D., and Zweig, G. (2015). Using recurrent neural networks for slot filling in spoken language understanding. *IEEE/ACM Trans. Audio, Speech and Lang. Proc.*, 23(3):530–539.
- Mitsuda, K., Higashinaka, R., and Matsuo, Y. (2017). What information should a dialogue system understand?: Collection and analysis of perceived information in chat-oriented dialogue. In *Proceedings of the 8th International Workshop on Spoken Dialogue Systems (IWSDS 2017)*. 9 pages.
- Raymond, C. and Riccardi, G. (2007). Generative and discriminative algorithms for spoken language understanding. In *Eighth Annual Conference of the International Speech Communication Association*, pages 1605–1608.
- Takahashi, T. and Yokono, H. (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.