UNDERSTANDING COMPUTER-DIRECTED UTTERANCES IN MULTI-USER DIALOG SYSTEMS

Dong Wang[*]

University of Texas at Dallas

Dilek Hakkani-Tür, Gokhan Tur

Microsoft Research

ABSTRACT

This work aims to understand user requests when multiple users are interacting with each other and a spoken dialog system. More specifically, we explore the use of multi-human conversational context to improve domain detection in a human-computer interaction system. We investigate the different effects of human-directed context and computer-directed context, and compare the impact of using different context window sizes. Furthermore, we employ topic segmentation to chunk conversations for determining context boundaries. The experimental results show that the use of conversational context helps reduce domain detection error rate, especially in some specific domains. And though computer directed context is more reliable, the results show that the combination of both computer and human addressed utterances within a reasonable window size performs the best.

Index Terms— spoken language understanding, domain detection, conversational systems, multi-party conversations.

1. INTRODUCTION

In the past decades, spoken language understanding (SLU), which serves as a key component in human-computer conversational interaction systems, has been studied in both commercial and academic communities [1]. Almost all of the existing spoken dialog systems set the scenario where a user speaks to the system and the system gives feedback in response to the user's request, such as making a call to check flight status [2] or find a restaurant [3]. However, none of them is able to be actively involved in multi-user conversations. In this work, we investigate SLU in multi-user spoken dialog systems, where the users can talk to each other as well as the computer. For example, imagine multi-user spoken dialog systems, where you can chat with your friend, maybe discuss where you are going for dinner or where you will have a vacation during spring break, in the meantime, the computer listens to both of you and identifies when you are talking to the computer and when you are talking to each other. When you say "find me Italian restaurants" to the computer, it responds by searching on the Internet and presenting you a list of Italian restaurants.

In such multi-user spoken dialog systems, users may *explicitly address the system with computer directed utterances* (**explicitly addressed**), for example, by pressing a button (i.e., "push-to-talk") or looking at the computer (i.e., "look-to-talk"), or using an addressing term, such as "computer", and then say their specific request. An example of such a scenario is the last utterance of the following conversation segment:

Speaker 1: ...
Speaker 2: ok, let's find a place then
Speaker 2: computer, are there any italian restaurants nearby?

Users may also *explicitly address the system with a turn, but without using an addressing signal* (**implicitly addressed**), such as by saying a specific term or pushing a button. For example:

Speaker 1: ...
Speaker 2: ok, let's find a place then
Speaker 2: are there any italian restaurants nearby?

Note that, in such cases, one can understand the machine is addressed by relying on a combination of speakers' gaze, words and prosody [4]. Users may also *just chat with each other, and the system may be a conversation participant and contribute to the conversation without being necessarily addressed, when it has any information to provide* (**conversation participant**). For example, in the following conversation segment, the machine is contributing to the conversation without being explicitly invoked:

Speaker 1: wanna eat lunch?
Speaker 2: ok, do you wanna walk to downtown and find an italian place?
Speaker 1: ok
Computer: Here are some italian restaurants in the downtown area (or just show restaurants on display).

In the first two scenarios, SLU involves detecting and interpreting utterances that are directed to the machine, where the multi-human conversation context can be used as additional information. In the third one, the main goal of SLU is interpreting the multi-human conversations. In this work, we are tackling the second scenario, however, components such as "determining the right conversational context to use", and "analyzing that context to find the utterance domain" can be useful in all three scenarios.

There have been many widely used human-computer dialog systems available, most of which are oriented in a specific application thus have very limited domains. On one side this constraint makes the system more accurate, but on the other side it limits their purpose, so the user has to resort to different resources for different tasks. In order to build a personal digital assistant (PDA) for general purpose, we need a more complex SLU system to accommodate the need of different users for different intentions from multiple domains. The human-computer interaction system we describe in this paper, although still supports limited domains, is general enough to respond most of the queries the users may have on Internet. It covers 38 domains such as "search", "music", "movie", "restaurant", and "shopping". For each domain, a semantic ontology that includes domain-specific "intents" and "slots" are defined to describe pos-

[*]This work was done when the first author was an intern at Microsoft

sible user requests specific to each domain. In such multi-domain systems, the process of interpreting user's request to the computer, domain detection is usually performed as the first step. Formally, the problem of domain detection is defined as, for the given computer-addressed utterance u, the goal is to estimate its domain c':

$$c' = argmax_{c \in C} p(c|u)) \quad (1)$$

where C is the set of all domains.

Previous work on domain detection [5] studied this problem in a single-user scenario, where there is only one user speaking to the computer and in that dataset, the utterances were independently collected without context, i.e., the user speaks a random query to the computer. This paper explores domain classification in a multi-human scenario like in the examples above. The goal is to find out whether the conversational context, including human addressed and computer addressed, is helpful for domain detection.

The contributions of our work are:

- an effective approach to select useful context from noisy conversations,
- a demonstration that topic segmentation helps context selection in spontaneous conversations, and
- a comparison of the effect of computer directed and human directed contexts as well as different context window sizes on domain detection.

In the next section we briefly present the related work. Then in Section 3 we describe the approach for exploiting conversational context. Then in Section 4 we provide an overview of the multi user dialog system used in this study. Section 5 shows experimental results with discussion.

2. RELATED WORK

In a multi-human dialog system like we describe in the beginning of Section 1, while the utterances directed to the computer are the main focus of the SLU system, many problems initially defined on human-human conversation understanding are closely related to this task. For example, dialog act labeling [6, 7] defines the function of each dialog act and their relationship, which helps us better understand the user intent; addressee detection [4] aims to differentiate the utterances addressed to the other speaker or to the computer, so that the user can speak to the computer naturally without any intervention.

Our work follows the line of work by [5, 8, 9]. [5] shows exploiting web query click logs using a semi-supervised method outperforms the fully supervised approach using limited annotated data on domain classification. [9] employs a joint model to mine the domain, intent and slots simultaneously instead of using separate steps.

The goal of this paper is to explore the conversational context in multi-human conversations. To the best of our knowledge there is no prior research that utilizes context for domain detection, but context is exploited in some other applications: [10] uses some simple context such as previous dialog act and speaker labels to help dialog act labeling, and [11] employs the topic shift/continue detection to help find relevant answers in a question answering system.

3. SELECTION OF CONVERSATIONAL CONTEXT

In a multi-user, multi-domain dialog system, where the users can switch topics of their utterances, parts of the conversational context may not be relevant for interpreting the computer-addressed utterances. Hence, we propose and investigate methods for picking the previous conversational context that may help the detection of the domain of the computer-addressed utterances.

3.1. Incorporating Prior Knowledge

In this work, we propose to use the utterance domain detection trained from computer addressed utterances (from single user interactions with the computer) to assign a domain category to the human-human parts of the conversations. In our previous work [5, 12], we have collected a dataset with domain annotation on single-user utterances, which can be utilized as prior knowledge to better mining the context. This single-user utterance dataset contains 21 categories that include domains such as "movies", "restaurants", in addition to "command" and "conversational" utterances (such as, "oh yes", etc.). We train a statistical domain classification model using ICSIboost [13] on this dataset with lexical features, then perform classification on each utterance (including both computer addressed and human addressed) in multi-user sessions. For utterance u_i, the output of ICSIboost is a vector of scores $< score_{i1}, score_{i2}, ..., score_{iT} >$ $(T = 21)$, we use the sigmoid function to calibrate the scores to a vector of real values $< p_{i1}, p_{i2}, ..., p_{iT} >$, then normalize it to a probability distribution $< t_{i1}, t_{i2}, ..., t_{iT} >$.

$$p_{ik} = \frac{1}{1 + exp(-2 \times n \times score_{ik})} \quad (2)$$

$$t_{ik} = \frac{p_{ik}}{\sum_{j \in [1,M]} p_{ij}} \quad (3)$$

Here n is the number of training iterations of ICSIboost (i.e., the number of weak learners).

We use this domain distribution for several purposes:

1. When selecting context, if the domain distribution indicates the utterance is categorized as a "command" or a "conversational" utterance we skip it, because such utterances do not bear any topic information;
2. For each utterance u_i, we calculate its domain confidence as $max_{k \in [1,T]} t_{ik}$. When the confidence is lower than a threshold, we consider it ambiguous and exclude it;
3. In topic segmentation, domain distribution is used to calculate similarity between user utterances (see next subsection);
4. We use this prior domain distribution as topic distribution in context and as classification features.

3.2. Topic Segmentation

Usually a conversation is composed of multiple topics/themes, for example, users would talk about their future vacation and then check the weather there, then probably check flights in the same conversation. In order to select context within the same topic segment as the current utterance, we use topic segmentation to divide a conversation into segments, each belonging to a single topic. We apply an approach similar to TextTiling [14], based on domain detection scores, for topic segmentation. Instead of using words, we use the average domain distribution in each block to calculate the cosine similarity between blocks (window size = 3).

When calculating the window size, similar to context selection, we skip the "command" and "conversational" utterances as well as the ones with a low confidence score. We set the threshold of topic boundary as 0.35, i.e. when the similarity between two windows is lower than 0.35, we consider there is a topic boundary. When we search for previous context, we stop at the topic boundary.

ID	prior label	utterance transcription
u_{i-5}	search	*see we have this thing where we go to*
	movie	*I know it's annoying.*
	command	**start over.**
u_{i-4}	music	*ohh. try turning it. ohh no. it's okay.*
u_{i-3}	movie	**look for Avengers reviews on Rotten Tomatoes DOT com.**
u_{i-2}	navigation	**go to Rotten Tomatoes DOT com.**
u_{i-1}	movie	*you also can't get new movies. like anything.*
	command	**go back.**
u_i	music	**Marvel's The Avengers.**

Table 1. Feature extraction example.

3.3. Features

Table 1 shows a conversation fragment from our dataset. The utterances addressed to the computer are in bold font, these are the ones we need to perform domain detection on. For utterance u_i, the context we exploit is its previous utterances u_{i-1} to u_{i-w}, w is the window size. The utterances without any nouns or noun phrases (such as, "hmm", "yes") are also skipped, as our preliminary study shows that the contexts with nouns or noun phrases are more informational. We experiment with various window sizes w from 1 to 10 to investigate how much context needed to boost the performance.

The features we use in domain detection are listed below:

- Lexical features: Word n-grams in current utterance u_i.
- Contextual features: we explore two methods to integrate the features from several context utterances:
  - C_AVG: The label of majority vote by all context utterances within a window size, and their average topic distribution.

$$T_f = \frac{\sum_{j \in [i-1, i-w]} T_j}{w} \quad (4)$$

    where T_f is the context feature vector, T_j is the topic distribution of utterance u_j. e.g., in Table 1, if the window size is 5, the selected label is "movie", and the context feature vector is an average of u_{i-1} to u_{i-5}.
  - C_TOP: The label of majority vote by all context utterances within window size, which is the same as in C_AVG, but use the topic distribution of the utterance with this top label nearest to the current utterance as features. e.g., in Table 1, if the window size is 5, the selected label is still "movie" but we use the topic distribution of u_{i-1} as the feature vector.

4. DATA COLLECTION

We collected the data in a typical living room setting. In each session, two users who know each other were invited to have a chat. They sit in a sofa facing a screen which is connected to a computer. During their conversation, they may ask computer to do some tasks related to what they are discussing, such as searching for flights, playing music, shopping on-line, etc. Before each session the users are presented with the capabilities of the system, and a set of "commands" that may be used to navigate the contents (such as "go back" and "start over"). A Kinect in front of sofa records people's speech and gestures and then passes the data to a server, which has an automatic speech recognition system that transcribes the speech to text, then the system first identifies if the utterance is addressed to it. If the user utterance is intended as a query to computer, the system processes the utterance and presents users the found results on the screen and provides feedback with text and spoken prompts. After seeing the results, the users can either continue chatting with each other, or continue conversation with the computer, and select an item from the screen (i.e., "click" on a link on the current page), identify new constraints to their search, initiate a new request, or navigate the results.

baseline	**feature**	**error rate (%)**	
	LEX	38.8	
window size	**feature**	**Cs**	**Cs&Hs**
1	LEX+C_TOP	35.6	38.2
5	LEX+C_AVG	38.2	36.3
	LEX+C_TOP	36.0	35.0
10	LEX+C_AVG	37.0	38.1
	LEX+C_TOP	37.2	36.7

Table 2. System performance (error rate) of baseline and using different context feature C_AVG and C_TOP under varied window size.

Our data set consists of 36 conversations, with an average length of 26 minutes. The number of total utterances is 12,983 and 4,752 out of them are addressed to the computer. The continuous stream of user utterances are also captured by individual close-talking headset microphones. These are manually transcribed and annotated as computer-addressed or not. The computer-addressed utterances are further annotated with utterance domain, intent and slot values. In this work we exclude the "command" and the "click" utterances, as these two categories are the majority classes in our dataset and are easy to detect. For example, our current "click" detector shows only 1% error rate in previous experiments. Then we have 1,458 computer-addressed user utterances left that cover 25 domains.

5. EXPERIMENTS

We perform 12-fold cross validation on the 36 conversations, each time using 3 of them as the test set. We compare our system performance with a baseline approach which only uses lexical features from the current turn (LEX).

We evaluate the contextual features with different window size 1, 5 and 10, using two different settings: one uses only computer addressed utterances (Cs); the other use both computer addressed (Cs) and human addressed utterances(Hs). In these experiments, we use manual annotations for Cs and Hs. The results measured by error rate are shown in Table 2. It shows that using context feature is able to reduce the error rate from 38.8% in baseline to 35.0% by using both Cs and Hs with window size 5. This last approach does not require the use of correct detection of computer addressed utterances for the context.

The results show that context feature C_TOP is consistently better than C_AVG in different window size and different context types, except when using Cs only with window size 10. This is due to two reasons: one is the property of people's conversation, which can be demonstrated by the example in Table 3: a couple is discussing where they are going to eat for dinner. The first column is the prior label given by single-human training set and the second column is speaker ID. Computer directed utterances are labeled in bold font. They talk about weather, then transportation, then restaurant. So it is natural to divert to other sub-topics even in the same theme (finding restaurant). Sometimes there is a topic boundary (like in this example), but sometimes not, if there are also some "restaurant" utterances involved in the first part. Another reason is the inaccurate prior label, such as the utterances labeled as "movie" and "music" in this example. These errors occur because "movie" and "music" instances have a large vocabulary in our single-human dataset, and that data set doesn't include any human-human interaction examples.

prior label	speaker	utterance transcription
weather	A	**how is the weather tonight?**
movie	B	*ohh looks nice.*
music	A	*aww it's a bit cold.*
transportation	B	*you wanna walk to castro?*
calendar	A	*okay good idea well let me call Adam.*
==========topic boundary==========		
restaurant	A	*well okay before we go outdoor before let's decide on what we are gonna eat.*
restaurant	A	*what do you wanna eat?*
restaurant	B	*let's check.*
events	A	*you wanna go to this new place on at the corner of california and castro?*
music	A	*you remember the name?*
restaurant	B	*let's check places with outdoor dining first*
conversational	A	*okay let's do that*
restaurant	B	*find restaurants with outdoor dining along the castro*

Table 3. Example of topic shifts in context.

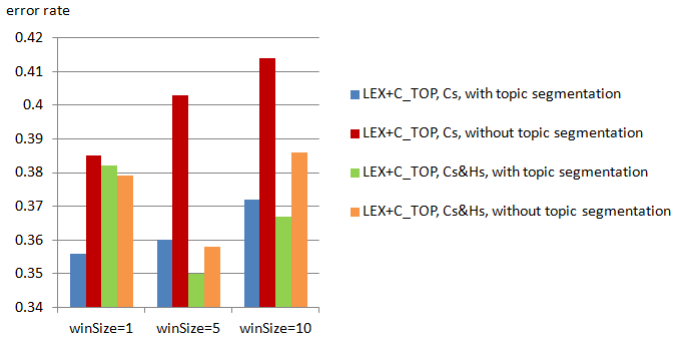

Fig. 1. Comparison of system performances with topic segmentation vs. without topic segmentation

From the experimental results, we observe a drop of performance from window size 1 to window size 5 when use only Cs context, but when use both Cs and Hs context, the performance improves from window size 1 to 5. It shows that if only one previous context is used, Cs is more reliable, but if more context can be included, it is better to use both of them. It is because human-to-human conversations contain off-topic noise, and longer context may contain more topic-related utterances. In both settings longer window size (10) fails to bring any gain, which shows that only limited context is needed to get better performance.

Though the best performance in our experiments is achieved by using both Cs and Hs with window size 5, it only has marginal gain compared to using only Cs with just one context utterance. This is against our expectation that people usually give the query related to what they are discussing. After analyzing the data, we find that the main reason is not that Hs does not contain as much information as Cs, but domain estimation on Hs is very noisy so it is too hard to mine the topic information from them.

Topic segmentation confines the context selection to a limited scope so it will not pick the context with a different topic/theme from current utterance. We compare the system performance with topic segmentation and without segmentation where we do not stop at a topic boundary in Figure 1, based on LEX+C_TOP features, as it performs better. The result shows using topic segmentation is always better except when using both Cs and Hs with window size 1. It demonstrates that topic segmentation can effectively avoid the noisy context by context selection.

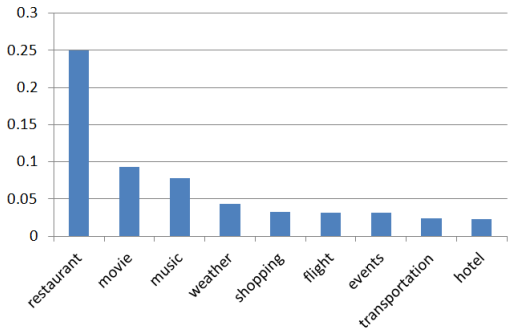

Fig. 2. Percentage of each domain.

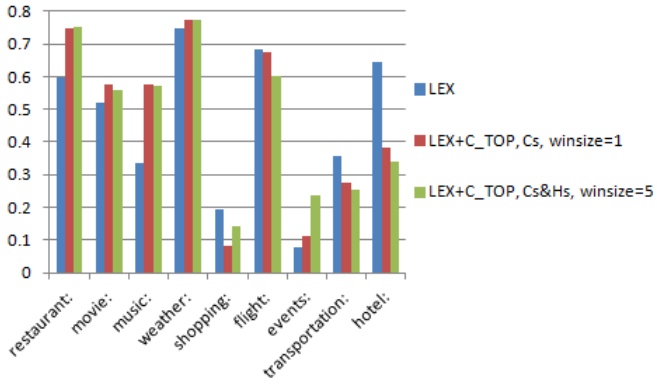

Fig. 3. F-measure on major domains.

We measure the precision, recall or F-measure on 25 domains individually, and find an interesting phenomenon that using context improves performance in some domains like "events", "restaurants", "music" and "movies" in both precision and recall, while hurts the performance in domains like "flight" and "transportation". Figure 2 shows the relative frequency of each domain in our dataset, and Figure 3 compares the F-measure of baseline and two best system performances using context feature on some major domains, where the infrequent domains and the ones that have no obvious variance between systems are omitted. The reason could be because people would talk for a while on some topics like "events" or "restaurant", but shortly in other topics like "transportation".

6. CONCLUSION AND FUTURE WORK

This work proposes to exploit conversational context to improve domain detection. We use the prior topic distribution and topic segmentation to select informational context. The experimental results show that contextual feature is able to boost the performance in some specific domains.

In our current work, the topic distribution derived from prior single-user dataset plays an important role in context selection and is used as context feature, but the training set used to derive this distribution is too limited. In future work, we are going to improve this topic distribution by exploiting web data. In addition, given the observation that context feature is useful on some domains while not on others, we can limit the use of context only to some specific domains.

Acknowledgments: We would like to thank Ashley Fidler for her help in preparing the data sets and useful discussions.

7. REFERENCES

[1] G. Tur and R. De Mori, Eds., *Spoken Language Understanding: Systems for Extracting Semantic Information from Speech*, John Wiley and Sons, New York, NY, 2011.

[2] P. J. Price, "Evaluation of spoken language systems: The ATIS domain," in *Proceedings of the DARPA Workshop on Speech and Natural Language*, Hidden Valley, PA, June 1990.

[3] O. Lemon and X. Liu, "DUDE: a dialogue and understanding development environment, mapping business process models to information state update dialogue systems," in *Proceedings of the EACL*, Trento, Italy, April 2006.

[4] Elizabeth Shriberg, Andreas Stolcke, Dilek Hakkani-Tür, and Larry Heck, "Learning when to listen: Detecting system-addressed speech in human-human-computer dialog," in *Proceedings of Interspeech*, 2012.

[5] Dilek Hakkani-Tür, Larry Heck, and Gokhan Tur, "Exploiting query click logs for utterance domain detection in spoken language understanding," in *ICASSP*. IEEE, 2011, pp. 5636–5639.

[6] Andreas Stolcke, Klaus Ries, Noah Coccaro, Elizabeth Shriberg, Rebecca Bates, Daniel Jurafsky, Paul Taylor, Rachel Martin, Carol Van Ess-Dykema, and Marie Meteer, "Dialogue act modeling for automatic tagging and recognition of conversational speech," *Computational Linguistics*, vol. 26, pp. 339–373, 2000.

[7] Jeremy Ang, Yang Liu, and E. Shriberg, "Automatic dialog act segmentation and classification in multiparty meetings," in *Proceeding of Acoustics, Speech, and Signal Processing*, 2005, pp. 1061–1064.

[8] Dilek Hakkani-Tŕ, Gokhan Tur, and Asli Celikyilmaz, "Mining search query logs for spoken language understanding," in *NAACL-HLT Workshop on Future directions and needs in the Spoken Dialog Community: Tools and Data (SDCTD 2012)*, 2012.

[9] Asli Celikyilmaz and Dilek Hakkani-Tür, "A joint model for discovery of aspects in utterances," in *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2012, pp. 330–338.

[10] Kristy Elizabeth Boyer, Eun Young Ha, Robert Phillips, Michael D. Wallis, Mladen A. Vouk, and James C. Lester, "Dialogue act modeling in a complex task-oriented domain," in *SIGDIAL*, 2010, pp. 297–305.

[11] Manuel Kirschner and Raffaella Bernardi, "Towards an empirically motivated typology of follow-up questions: The role of dialogue context," in *SIGDIAL*, 2010, pp. 322–331.

[12] Dilek Z. Hakkani-Tür, Gokhan Tur, Larry P. Heck, and Elizabeth Shriberg, "Bootstrapping domain detection using query click logs for new domains.," in *INTERSPEECH*, 2011, pp. 709–712.

[13] Benoit Favre, Dilek Hakkani-Tür, and Sebastien Cuendet, "Icsiboost," http://code.google.come/p/icsiboost, 2007.

[14] Marti A. Hearst, "Texttiling: Segmenting text into multi-paragraph subtopic passages," *Computational Linguistics*, pp. 33–64, 1997.