Towards Spoken Language Interfaces for Mobile Applications

Yun-Nung Chen
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213, USA
yvchen@cs.cmu.edu

Ming Sun
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213, USA
mings@cs.cmu.edu

Alexander I. Rudnicky
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213, USA
air@cs.cmu.edu

CHI'16 Workshop, May 7–8, 2016, San Jose, CA, USA.

Abstract

People are able to interact with domain-specific intelligent assistants (IA) via spoken language interfaces to access mobile applications. However, sometimes user goals are complex and may require interactions with multiple applications/domains. However current IAs are limited to specific domains and users have to directly manage execution spanning multiple applications in order to engage in more complex activities. An ideal personal agent would be able to model user intents at different levels (single-domain to cross-domain dialogues). This paper addresses several challenges about hierarchical language understanding in the context of spoken dialogue systems (SDS). Our experiments show that language understanding at different levels allows an agent to actively suggest apps relevant to pursuing particular user goals and reduce the cost of users' self-management.

Author Keywords

Spoken dialogue system, user modeling.

ACM Classification Keywords

I.2.1 [Applications and Expert Systems]: Natural language interfaces; I.2.7 [Natural Language Processing]: Language Parsing and Understanding; I.2.11 [Distributed Artificial Intelligence]: Intelligent agents

Introduction

Smart devices, such as phones or TVs, now host applications (apps) from different domains. In recent, spoken dialogue systems (SDS) are appearing on smart-phones and allow users to launch apps via spontaneous speech. Each app is designed to handle a limited number of domains (usually one) so that a typical SDS needs predefined task domains to support the corresponding functions, such as `setting_an_alert` (CLOCK) and `navigation` (MAPS). However, an SDS is unable to dynamically support functions provided by newly installed or not yet installed apps.

Conventional intelligent assistants (IA) passively select *one* domain from multiple domains according to a user input, treating each domain (e.g. restaurant search, messaging, etc.) independent of each other and ignoring the relationships between domains and the ultimate user intention behind cross-domain behaviors [1, 2, 3, 4, 10, 11, 13, 14, 15]. However, given different context and the same user utterance, intended apps may differ. On the other hand, users can mentally arrange apps and seamlessly coordinate the information among them. However, manually launching apps one by one may be time-consuming and difficult, especially for elder users and users with (visual) disabilities, although vocabularies of a touch-screen or gestures have been enriched significantly over the past decade [8].

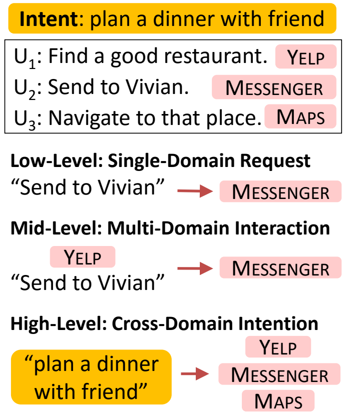

Intent: plan a dinner with friend
U_1: Find a good restaurant.
YELP
U_2: Send to Vivian.
MESSENGER
U_3: Navigate to that place.
MAPS
Low-Level: Single-Domain Request
"Send to Vivian"
MESSENGER
Mid-Level: Multi-Domain Interaction
YELP
"Send to Vivian"
MESSENGER
High-Level: Cross-Domain Intention
"plan a dinner with friend"
YELP
MESSENGER
MAPS

Figure 1: Illustration of language understanding at different levels.

To reduce manual annotations for developing app-based language interactive interfaces, this paper discusses challenges and research directions about spoken language understanding (SLU), and mainly focuses on app prediction of IAs at different intent levels— 1) low-level: single-domain requests, 2) mid-level: multi-domain interactions, and 3) high-level: cross-domain intentions. Figure 1 illustrates these three tasks using a dialogue example. The research summaries contribute to multidisciplinary goals in terms of speech processing, natural language processing and human-computer interaction.

Low-Level SLU: Single-Domain Requests

Challenge

There are two main challenges of single-domain requests.

- Supporting Unexplored Apps
  Because a typical SDS requires predefined domain ontology to understand corresponding functions, we ask the following question: with open-domain requests, how can a system dynamically and effectively provide corresponding functions to fulfill users' requests?
- Hidden Semantics Inference
  Another challenge of SLU is the inference of hidden semantics. Given a user utterance "i would like to contact Alex", its surface patterns include explicit semantic information about "*contact*"; however, it also includes hidden semantic information such as "*message*" and "*email*", since the user likely intends to launch apps like MESSENGER (message) or OUTLOOK (email) even though they are not directly observed in the surface patterns. Such hidden semantics was shown to be useful for learning better SLU models and can be captured by matrix factorization (MF) techniques [3, 4, 5].

Proposed Approach

To tackle the above problems, an SLU model takes account of app descriptions and spoken utterances along with enriched semantic knowledge in a joint fashion in order to predict intended apps [1, 4]. More specifically, structured knowledge resources (e.g. Wikipedia, Freebase, etc.) are utilized to locate identified entities in a given utterance and then enrich the semantics of utterances. Then applying

	MAP		P@10	
Feature	**LM**	**MF-SLU**	**LM**	**MF-SLU**
(a) Lexical	25.1	29.2 (+16.2%)	28.6	29.5 (+3.4%)
(b) Lex+Knowl	32.0	34.2 (+6.8%)	31.2	32.5 (+4.3%)

Table 1: Low-level SLU on mean average precision (MAP) and precision at 10 (P@10) (%).

unsupervised approaches such as matrix factorization enables an SDS to dynamically support non-predefined domains based on the semantics-enriched models [3, 4]. We evaluate the performance by examining whether predicted apps are capable of fulfilling users' requests.

Experiments

A total of 13 domains are defined, including "navigation", "email writing", "music playing", etc [1]. Then each subject was shown with images corresponding to domain-specific tasks and asked to voice 3 different ways for making requests in order to fulfill the task implied by the images. The corpus contains 195 utterances, and the word error rate (WER) is reported as 19.8% using Google Speech API.

Table 1 shows the performance of a baseline language modeling approach (LM) and the proposed MF method (MF-SLU), where LM models explicit semantics and MF-SLU additionally considers implicit semantics. The features for app prediction include lexical observations (row (a): Lexical) and leveraging lexical observation and structured knowledge (row (b): Lex+Knowl). Enriching features with structured knowledge improves the performance, and the MF-SLU approach outperforms the baseline LM. Applying MF-SLU on enriched features for low-level SLU achieves about 34% on MAP, showing the feasibility of dynamically supporting unexplored mobile apps given single-domain requests.

Mid-Level SLU: Multi-Domain Interactions

Challenge

Typically each domain (e.g. restaurant search, messaging, etc) is independent of other domains, so only current utterances are considered to decide the desired apps in SLU. Some IAs modeled user intents by keeping the contexts from the previous utterances, but they did not consider behavioral patterns of individual users [1]. To improve understanding, some studies utilized the nonverbal contexts like eye gaze and head nod as cues to resolve the referring expression ambiguity and to improve driving performance respectively [7, 9]. The main challenge is language ambiguity, which often makes prediction difficult, for example, two apps EMAIL and MESSAGE are both plausible by hearing an utterance "Send to Alex". To disambiguate the understanding, the following cues should be considered:

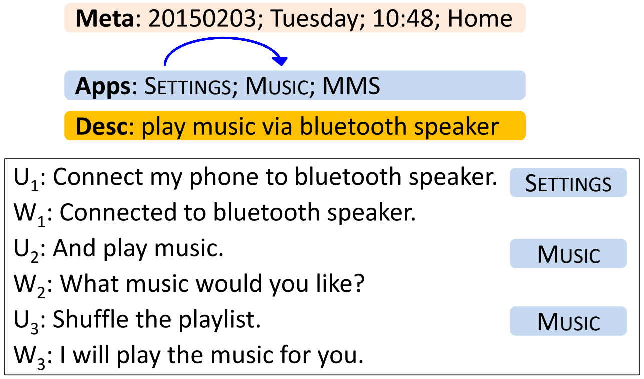
- User Preference
  Some people prefer MESSAGE to EMAIL even given the same input utterance.
- App-Level Contexts
  If a user always texts his friend via MESSENGER instead of GMAIL right after finding a restaurant via YELP, such contexts would help disambiguate the above utterance. Similarly, MESSENGER may be more likely to follow CAMERA, and OUTLOOK may be more likely to follow EXCEL.

Proposed Approach

In terms of mid-level SLU performance, the proposed approach focuses on leveraging app behavioral history to model user preference and app-level contexts in a bottom-up way. An MF-based approach that models speech and app usage patterns is utilized to predict intended apps by inferring implicit relations between lexical and contextual features [2, 3].

	#User	Age
Male	4	23.0
Female	10	34.6
Age < 25	6	21.2
Age ≥ 25	8	38.9
Native	12	31.8
Non-native	2	28.5

Table 2: Age distribution for subject characteristics. Age informally indicates young and old. A native Korean and Spanish speaker participated; both were fluent in English.


Meta: 20150203; Tuesday; 10:48; Home
Apps: SETTINGS; MUSIC; MMS
Desc: play music via bluetooth speaker
SETTINGS
MUSIC
MUSIC

U_1: Connect my phone to bluetooth speaker.

W_1: Connected to bluetooth speaker.

U_2: And play music.

W_2: What music would you like?

U_3: Shuffle the playlist.

W_3: I will play the music for you.

Figure 2: User connected SETTINGS and MUSIC and noted that these two apps were used to *play music via bluetooth speaker*. Wizard-of-Oz dialogues were collected and manually transcribed (U: user; W: wizard).

Experiments

We logged real-life interactions at app-level from users' smart phones. Meta information such as date, time, location was recorded. Participants were presented with episodes (segmentated by idle interval) and asked to group events into sequences corresponding to individual activities [12] (which we will also refer to as intents); We then requested two types of user annotations: 1) what apps were used for a particular goal; and 2) what the goal was (i.e., intent description). The upper part of Figure 2 illustrates an annotation example, where SETTINGS was linked to MUSIC and served a high-level intent—*play music via bluetooth speaker*. Users were also asked to re-enact the smart phone interaction by talking with a Wizard-of-Oz system. A transcribed dialogue is shown in the lower part of Figure 2.

We had 14 participants and collected 533 sessions, where there are 455 dialogues involving multiple user turns and the average number of used apps per user is 19.4 [18]. The corpus characteristics is displayed in Table 2. The WER is 22.7% using Google Speech API.

	MAP		ACC	
Feature	**MLR**	**MF-SLU**	**MLR**	**MF-SLU**
(a) Lexical	52.1	52.7 (+1.2%)	48.2	48.3 (+0.2%)
(b) Lex+Cxt	53.9	55.7 (+3.3%)	50.1	51.9 (+3.6%)

Table 3: Mid-level SLU on mean average precision (MAP) and turn accuracy (ACC) (%).

Table 3 shows the results of a baseline multinomial logistic regression (MLR) method and the proposed MF-SLU on lexical features (row (a): Lexical) and multimodel features (row (b): Lex+Cxt) The proposed multi-model MF-SLU system achieves about 56% of MAP and 52% of turn accuracy for app prediction, showing that behavioral contexts help better inference between diverse features for language disambiguation.

High-Level SLU: Cross-Domain Intentions

Challenge

User goals are complex and may require interactions with multiple apps, but current IAs are limited to specific domains and users have to directly manage execution spanning multiple apps in order to fulfill more complex activities. The main challenge is to allow our personal IAs to help organize apps/domains automatically given user requests expressed at the level of intentions. For example, upon receiving "can you help me plan an evening out with my friends?", the agent may respond "Okay, to *plan a dinner event*, I need to know *where*, *when* and *who*". The information collectively constructs a shared context across app boundaries. Thus, a unified interaction could be provided, instead of concatenating individual domains managed by the user.

Feature	Pers.	Gene.
(a) Lexical	50.8	23.8
(b) +QryEnr	54.9	26.2
(c) +AppSim	50.8	30.7
(d) +QryEnr +AppSim	**54.9**	**32.7**

Table 4: High-level SLU on weighted average F-measure across 14 participants (%). (Pers.: Personalized; Gene.: Generic)

Proposed Approach

We build a layer above individual apps, which links multiple apps to a specific intention underlying user activities, so an agent would be able to manage interactions at the level of intentions, mapping intents into multiple existing apps/functionality [17, 18, 19]. For example, upon receiving "can you help me plan an evening out with my friends?", we would like our agent to find a restaurant with good reviews (YELP), reserve a table (OPENTABLE) and contact friends (MESSENGER). Figure 3 illustrates the pipeline of high-level SLU, where two components are proposed to process the mapping from a high-level intent description to corresponding apps in a top-down way.

1. Intent understanding
   Given a high-level intent description, we apply K-nearest neighbors (KNN) to find the K most similar past interactions for deciding the current intent. The features can be enriched with semantically related words, for example {shoot, photo} $\rightarrow$ {shoot, take, photo, picture, selfie} (QryEnr) [16].
2. Intent realization
   Based on the decided intent, we take a representative app sequence, which is extracted by collapsing multiple app sequences into one using ROVER, to generate multiple apps [6]. Recommended apps can be further personalized to the ones installed, e.g., BROWSER $\rightarrow$ CHROME (AppSim).

Experiments

The intent descriptions and associated apps from multi-app interactions are utilized to perform this task. Table 4 shows the performance of app prediction for high-level SLU in terms of personalized and generic models. The features include lexical observations (row (a): Lexical), incorporating semantically similar words by query enrichment (row (b): +QryEnr), similar apps (row (c): +AppSim), and combination of all features (row (d)). It is found that query expansion improves performance for both personalized and generic models. The app similarity can improve the generic model performance to 32.7% on F-measure. The gap between personalized and generic model can be reduced by adopting our proposed techniques.

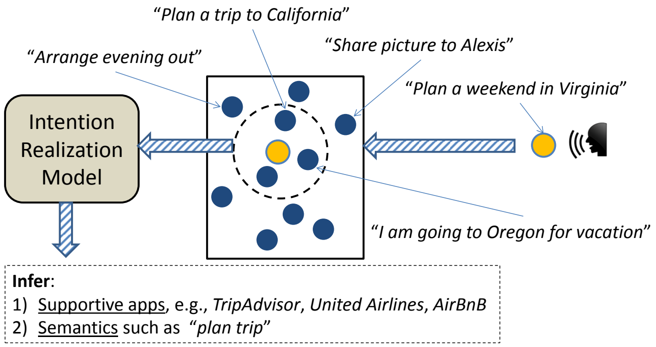

"Plan a trip to California"
"Arrange evening out"
"Share picture to Alexis"
"Plan a weekend in Virginia"
Intention Realization Model
"I am going to Oregon for vacation"
Infer:
1) Supportive apps, e.g., TripAdvisor, United Airlines, AirBnB
2) Semantics such as "plan trip"

Figure 3: User connected SETTINGS and MUSIC and noted that these two apps were used to *play music via bluetooth speaker*. Wizard-of-Oz dialogues were collected and manually transcribed.

Conclusion and Future Work

This paper discusses the challenges of hierarchical language understanding for mobile apps and presents some approaches to guide potential research directions of speech and multi-model interactive interfaces for intelligent assistants. Our long-term goal is to create agents that observe recurring human activities, figure out the underlying intentions and then provide active support through language-based interaction (in addition to allowing the user to explicitly teach the agent about complex tasks). The ideal agent can learn to manage activities on a level more abstract than provided by app-specific interfaces and would allow users to build their personalized agents that combine the functionality of existing apps according to their usage.

References

[1] Y.-N. Chen and A. I. Rudnicky. 2014. Dynamically supporting unexplored domains in conversational interactions by enriching semantics with neural word embeddings. In *Proc. of SLT*. 590–595.

[2] Y.-N. Chen, M. Sun, and A. I. Rudnicky. 2015a. Leveraging Behavioral Patterns of Mobile Applications for Personalized Spoken Language Understanding. In *Proc. of ICMI*.

[3] Y.-N. Chen, M. Sun, and A. I. Rudnicky. 2015b. Matrix Factorization with Domain Knowledge and Behavioral Patterns for Intent Modeling. In *Proc. of NIPS-SLU*.

[4] Y.-N. Chen, M. Sun, A. I. Rudnicky, and A. Gershman. 2016. Unsupervised User Intent Modeling by Feature-Enriched Matrix Factorization. In *Proc. of ICASSP*.

[5] Y.-N. Chen, William Yang Wang, A. Gershman, and A. I. Rudnicky. 2015. Matrix Factorization with Knowledge Graph Propagation for Unsupervised Spoken Lnaguage Understanding. In *Proc. of ACL-IJCNLP*.

[6] J. G Fiscus. 1997. A Post-Processing System to Yield Reduced Word Error Rates: Recognizer output voting error reduction (ROVER). In *Proc. of ASRU*.

[7] D. Hakkani-Tür, M. Slaney, A. Celikyilmaz, and L. Heck. 2014. Eye gaze for spoken language understanding in multi-modal conversational interactions. In *Proc. of ICMI*.

[8] C. Harrison, R. Xiao, J. Schwarz, and S. E. Hudson. 2014. TouchTools: leveraging familiarity and skill with physical tools to augment touch interaction. In *Proc. of SIGCHI on Human Factors in Computing Systems*.

[9] S. Kousidis, C. Kennington, T. Baumann, H. Buschmeier, S. Kopp, and D. Schlangen. 2014. A multimodal in-car dialogue system that tracks the driver's attention. In *Proc. of ICMI*.

[10] Q. Li, G. Tur, D. Hakkani-Tur, X. Li, T. Paek, A. Gunawardana, and C. Quirk. 2014. Distributed open-domain conversational understanding framework with domain independent extractors. In *Proc. of SLT*.

[11] B.-s. Lin, H.-m. Wang, and L.-s. Lee. 1999. A distributed architecture for cooperative spoken dialogue agents with coherent dialogue state and history. In *Proc. of ASRU*.

[12] C. Lucchese, S. Orlando, R. Perego, F. Silvestri, and G. Tolomei. 2011. Identifying task-based sessions in search engine query logs. In *Proc. of WSDM*.

[13] M. Nakano, S. Sato, K. Komatani, K. Matsuyama, K. Funakoshi, and H. G Okuno. 2011. A two-stage domain selection framework for extensible multi-domain spoken dialogue systems. In *Proc. of SIGDIAL*.

[14] A. I Rudnicky, Jean-Michel Lunati, and A. M Franz. 1991. Spoken language recognition in an office management domain. In *Proc. of ICASSP*.

[15] Seonghan Ryu, J. Song, S. Koo, S. Kwon, and G. G. Lee. 2015. Detecting Multiple Domains from UserâĂŹs Utterance in Spoken Dialog System. In *Proc. of IWSDS*.

[16] M. Sun, Y.-N. Chen, and A. I. Rudnicky. 2015a. Learning OOV through Semantic Relatedness in Spoken Dialog Systems. In *Proc. of INTERSPEECH*.

[17] M. Sun, Y.-N. Chen, and A. I. Rudnicky. 2015b. Understanding User's Cross-Domain Intentions in Spoken Dialog Systems. In *Proc. of NIPS-SLU*.

[18] M. Sun, Y.-N. Chen, and A. I. Rudnicky. 2016a. HELPR: A Framework to Break the Barrier across Domains in Spoken Dialog Systems. In *Proc. of IWSDS*.

[19] M. Sun, Y.-N. Chen, and A. I. Rudnicky. 2016b. An Intelligent Assistant for High-Level Task Understanding. In *Proc. of IUI*.