Leveraging Behavioral Patterns of Mobile Applications for Personalized Spoken Language Understanding

Yun-Nung Chen Ming Sun Alexander I. Rudnicky Anatole Gershman
School of Computer Science, Carnegie Mellon University
5000 Forbes Avenue, Pittsburgh, PA 15213, USA
{yvchen, mings, air, anatoleg}@cs.cmu.edu

ABSTRACT

Spoken language interfaces are appearing in various smart devices (e.g. smart-phones, smart-TV, in-car navigating systems) and serve as intelligent assistants (IAs). However, most of them do not consider individual users' behavioral profiles and contexts when modeling user intents. Such behavioral patterns are user-specific and provide useful cues to improve spoken language understanding (SLU). This paper focuses on leveraging the app behavior history to improve spoken dialog systems performance. We developed a matrix factorization approach that models speech and app usage patterns to predict user intents (e.g. launching a specific app). We collected multi-turn interactions in a WoZ scenario; users were asked to reproduce the multi-app tasks that they had performed earlier on their smart-phones. By modeling latent semantics behind lexical and behavioral patterns, the proposed multi-model system achieves about 52% of turn accuracy for intent prediction on ASR transcripts.

Categories and Subject Descriptors

I.2.7 [**Natural Language Processing**]: Language Parsing and Understanding—*speech and behavioral patterns*

Keywords

Spoken Language Understanding; Spoken Dialogue System; Intelligent Assistant; Behavioral Patterns; Matrix Factorization

1. INTRODUCTION

Spoken dialogue systems (SDS) are developed in smart-devices and allow users to launch apps via spontaneous speech. The key component of an SDS is a spoken language understanding (SLU) model, which predicts users' intended apps by understanding input utterances. However, language ambiguity often makes the prediction difficult, for example, two apps "Email" and "Message" are both plausible by hearing an utterance "*Send to Alex*". In this work, we improve the prediction performance base on our observation that the intended apps usually depend on 1) the preference of individual users (some people prefer "Message" to "Email") and 2) behavioral patterns in the app level ("Message" is more likely to follow "Camera" and "Email" is more likely to follow "Excel").

Typical intelligent assistants (IAs) treat each task (e.g. restaurant search, messaging, etc) independent of each other, where only user's current utterances are considered to decide the desired apps for SLU [3]. Some IAs modeled user intents by keeping the contexts from the previous utterances, but they did not consider the behavior patterns of individual users [1]. To improve understanding, some studies utilized the non-verbal contexts like eye gaze and head nod as cues to resolve the referring expression ambiguity and to improve driving performance respectively [7, 9]. Considering that human users often interact with their phones to carry out complicated tasks that span multiple domains and apps, user behavioral patterns as additional non-verbal signals may provide deeper insights into user intent [11, 2]. For example, if an user always texts his friend via "Message" instead of "Email" right after finding a good restaurant via "Yelp", such behavioral pattern helps disambiguate the intended apps of the utterance "*Send to Alex*". Therefore, this paper examines if user behavioral patterns improve understanding along with three-fold contributions:

- We are among the first to model personalized SLU combining with contextual app behaviors.
- We propose a novel matrix factorization (MF) approach to model the implicit semantics based on users' lexical and behavioral patterns.
- Combining lexical and behavior features helps disambiguate users' intended apps for intent prediction.

2. DATA COLLECTION

To captures how users converse in the course of multi-app tasks, we conducted a longitudinal study with 14 participants to investigate how people structure and perform such tasks via speech. This user study investigated users' behavior across multiple domains/apps. Users were provided with an Android app that logs all their app invocation, as well as the time and the phone's location. Logs were uploaded by participants on a daily basis with privacy control. Participants were invited to lab regularly to perform the following annotations as to the nature of their activities [12].

1. **Task Structure** — link applications that served a common goal.
2. **Task Description** — type in a brief description of the goal or intention of the task.

ACM ISBN X-XXXXX-XX-X/XX/XX.
DOI: http://dx.doi.org/10.1145/0000000.0000000

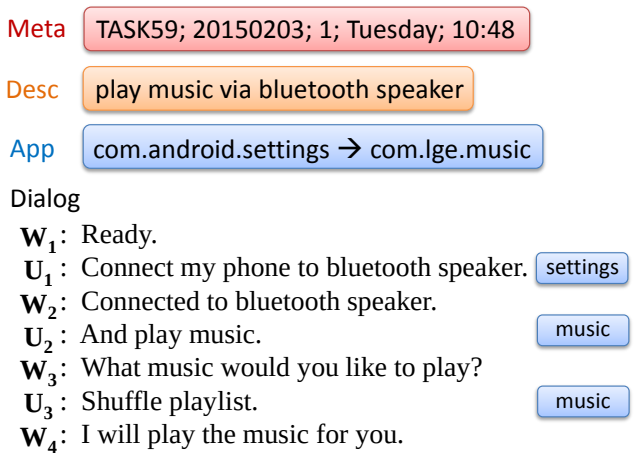

Figure 1: Multi-app task dialogue example

For example, in the upper part of Figure 1, "Settings" and "Music" are linked together since they were used for the goal of "play music via bluetooth speaker".

Then users were shown tasks annotated by themselves earlier along with the meta-data (date, location, and time), the task description they furnished earlier, and the apps that had been grouped (Meta, Desc, App lines in Figure 1). Subsequently they were asked to use a wizard system to perform the annotated task by speech in an office environment. The wizard arrangement was not concealed and the human wizard was in the same space (albeit not directly visible). The wizard was instructed to respond directly to participant's goal-directed requests and to not accept out-of-domain inputs. The participants were informed that they do not need to follow the order of the applications used on their smartphones. Other than for being on-task, we did not constrain what users could say.

Conversations between users (U) and the wizard (W) were recorded, segmented into utterances and transcribed by human and Cloud speech recognizer. One example dialogue is shown in Figure 1. Each user utterance was further associated with the apps that are able to handle it. As in Figure 1, "Settings" would deal with the utterance "*Connect my phone to bluetooth speakers*" (U_1) and "Music" would take care of music related utterances such as "*And play music*" (U_2) and "*Shuffle playlist*" (U_3).[1]

3. SPOKEN LANGUAGE UNDERSTANDING

The goal of our SLU model is to predict the apps that are more likely to be used to handle the user requests given input utterances and behavioral contexts. Considering that the multi-app tasks are usually user-specific, for each user we build a personalized SLU component to model his/her intents by framing the task as a multi-class classification problem, where we estimate the probability of each intent/app given the interaction at each turn t. The extracted features include word patterns in the utterance u_t along with its behavioral history $h_t = \{a_1, ..., a_{t-1}\}$. Note that the behavioral history is the set of apps that were previously launched in the ongoing dialogue. The following approaches focus on estimating $P(a_t \mid u_t, h_t)$ for intent prediction.

We perform 1) a standard multinomial logistic regression (MLR) to model explicit observations, which uses the standard maximum likelihood estimation approach by the gradient ascent to estimate the likelihood $P(a_t \mid u_t, h_t)$, and 2) a matrix factorization (MF) enhanced method to additionally model implicit feedback described below.

[1]Dataset available at http://AppDialogue.com

3.1 Matrix Factorization (MF)

The matrix factorization (MF) technique has been explored in different domains, and is credited as the most useful technique for recommendation systems [8]. An MF model considers the unobserved patterns and estimates their probabilities instead of viewing them as negative, which is able to model the implicit information [4].

We use MF to build personalized SLU models because of the ability to model 1) noisy data, 2) latent information, and 3) long-range dependencies between observations. Figure 2 illustrates the matrix, where an example testing utterance "*Send it to alice*" only contains the observed word "*send*", but the MF model is able to estimate higher probability for the intended app "IM (Instant Message)" than the one of "Email" by learning the latent semantics carried by behavior history ("Camera" in history infers semantics about "*tell*").

In our model, we use U to denote the set of user utterances, which contain word patterns W and the observed behavior history $H = \{h_t\}$, and A as the set of intents that we would like to predict (a predicted intent corresponds to an app). The pair of an utterance $x \in U$ and a word/history/app pattern $\langle x, y \rangle$, $y \in \{W \cup H \cup A\}$, is a *fact*. The input to our model is a set of observed facts $\mathcal{O}$, and the observed facts for a given utterance is denoted by $\{\langle x, y \rangle \in \mathcal{O}\}$. The goal of our model is to estimate, for a given utterance x and a given lexical/behavioral/intent feature y, the probability, $p(M_{x,y} = 1)$, where $M_{x,y}$ is a binary random variable that is true if and only if y is the lexical/behavioral pattern or predicted intent for the utterance x. We introduce a series of exponential family models that estimate the probability using a natural parameter $\theta_{x,y}$ and a logistic sigmoid function:

$$p(M_{x,y} = 1 \mid \theta_{x,y}) = \sigma(\theta_{x,y}) = \frac{1}{1 + \exp(-\theta_{x,y})}. \quad (1)$$

We construct a matrix $M_{|U| \times (|W|+|H|+|A|)}$ with observed facts to integrate lexical and behavioral information together.

3.1.1 Feature Matrix

To model the content of user utterances, we first build a word pattern matrix F_w with binary values based on word observations, where each row represents an interaction and each column refers to an observed word occurring in the utterances. In other words, F_w carries the basic word vectors for all interactions. Similarly, we build a behavioral history matrix F_h for utterances, where each row represents an utterance and each column refers to a set of previously observed apps in the dialogue.

To link the feature patterns with the corresponding apps, an intended app matrix F_a is constructed, where each column is an intent for launching a specific app. Hence, the entry is 1 for the intended app, and 0 otherwise. The feature model M is built from these three matrices: $M = [\ F_w \quad F_h \quad F_a\ ]$, which is illustrated in Figure 2.

3.1.2 Parameter Estimation

The proposed model is parametrized through weights and

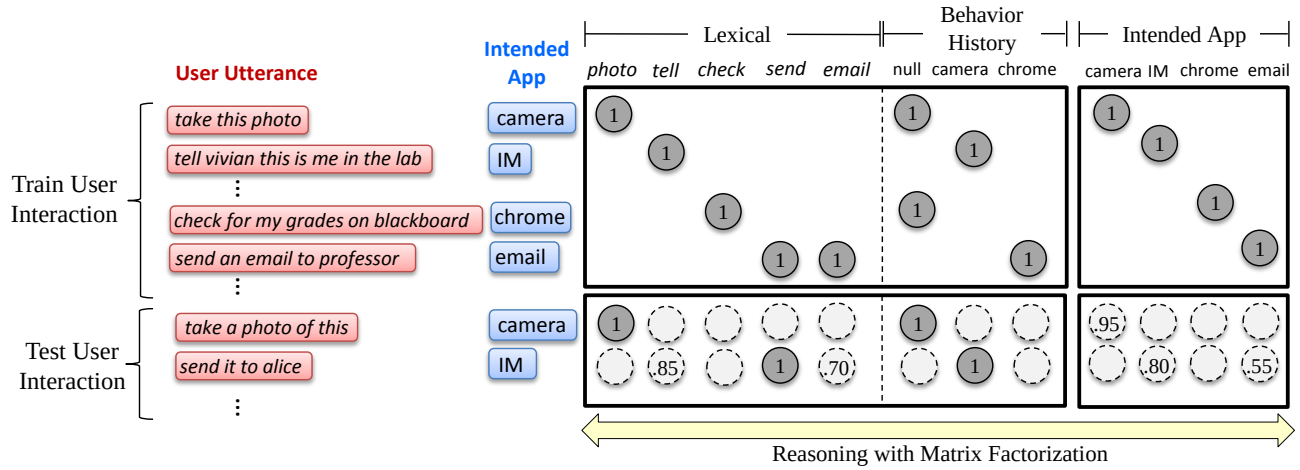

Figure 2: Our MF method completes a partially-missing matrix for implicit information modeling. Dark circles are observed facts, and shaded circles are latent and inferred facts. Reasoning with MF considers latent semantics to predict intents based on user utterance contents and app behavior history.

latent component vectors, where the parameters are estimated by maximizing the log likelihood of observed data [5].

$$\theta^* = \arg\max_\theta \prod_{x\in U} p(\theta \mid M_x) = \arg\max_\theta \prod_{x\in U} p(M_x \mid \theta)p(\theta)$$
$$= \arg\max_\theta \sum_{x\in U} \ln p(M_x \mid \theta) - \lambda_\theta, \quad (2)$$

where M_x is the vector corresponding to the utterance x from $M_{x,y}$ in (1), because we assume that each utterance is independent of others.

To avoid treating unobserved facts as designed negative facts, we consider our positive-only data as *implicit feedback*. Bayesian Personalized Ranking (BPR) is a optimization criterion that learns from implicit feedback for MF, which uses a variant of the ranking: giving observed true facts higher scores than unobserved (true or false) facts [10].

To estimate the parameters in (2), we create a dataset of *ranked pairs* from M: for each utterance x (e.g. "*take this photo*" in Figure 2) we created pairs of observed and unobserved facts: $f^+ = \langle x, y^+\rangle$ and $f^- = \langle x, y^-\rangle$, where y^+ corresponds to an observed lexical/behavioral/app feature (e.g., "*photo*" for lexical, "*null*" for behavior, "*camera*" for intended app); y^- corresponds to an unobserved feature (e.g. "*tell*", "*camera*", "*email*"). Then for each pair $\langle f^+, f^-\rangle$, we want our model to maximize the margin between $p(f^+)$ and $p(f^-)$ i.e., the difference between θ_{f+} and θ_{f-} according to (1). BPR maximizes the summation of each ranked pair with the objective:

$$\sum_{x\in U} \ln p(M_x \mid \theta) = \sum_{f^+\in\mathcal{O}} \sum_{f^-\notin\mathcal{O}} \ln \sigma(\theta_{f^+} - \theta_{f^-}), \quad (3)$$

which is an approximation to the per interaction AUC (area under the ROC curve), which correlates to well-ranked intents per utterance.

To maximize the objective in (3), we employ a stochastic gradient descent (SGD) algorithm [10]. For each randomly sampled observed fact $\langle x, y^+\rangle$, we sample an unobserved fact $\langle x, y^-\rangle$, which results in $|\mathcal{O}|$ fact pairs $\langle f^+, f^-\rangle$. For each pair, we perform an SGD update using the gradient of the corresponding objective function for MF [6].

Finally we can obtain the estimated probabilities of all lexical, behavioral, and intent-related features given the current dialogue turn. In this paper, we are interested in probabilities of intended apps $P(a_t \mid u_t, h_t)$, but as shown in Figure 2, hidden semantics (e.g. "*tell*", "*email*") can also be inferred from "*send it to alice*" in this model.

4. EXPERIMENTS

4.1 Experimental Setup

In our experiments, we collected smart-phone app usage data from 14 participants with Android OS version 4. The total number of multi-app spoken dialogues is 533 with 1607 utterances (on average 3 user utterances per dialogue). Among these dialogues, we have 455 multi-turn dialogs (82.3%), presenting the richness of behavioral features. The word error rate from Google automatic speech recognition (ASR) is reported as 25% without further text normalization.

We group 70% of each user's multi-app dialogues into the training set, and use the rest 30% as the testing set. For each user, we build a personalized SLU model with his/her own training data to estimate the probability distribution of all intended apps for each dialogue turn. To evaluate the prediction performance, we compute turn accuracy (ACC) as the percentage of our top 1 predictions match the correct apps. For a soft prediction, we also evaluate the performance by mean average precision (MAP), which considers the whole ranking list of all apps corresponding to each utterance. Table 1 shows the experiments performed on both Cloud ASR and manual transcripts.

4.2 Evaluation Results

The baselines here apply maximum likelihood estimation (MLE) to predict intents according to the observed distribution in the training data, where the prediction accuracy is around 20%. The reasons of the poor performance includes: 1) This intent prediction task here is different from traditional one, because a correct prediction is a single app that the user plan to use; hence this task is more difficult. For example, in traditional intent prediction task, "Messager" and "Gmail" belong to the same intent "communication", so deciding a single app is more challenging. 2) The total number of intents/apps in our dataset is relatively large, because we do not limit the apps users can use, where the average

Table 1: User intent prediction on turn accuracy (ACC) and mean average precision (MAP) (%). † means that all features perform significantly better than lexical/behavioral features alone; § means that integrating with MF significantly improves the MLR model (t-test with $p < 0.05$).

Approach			ACC					MAP				
			Lexical		Behavioral	All		Lexical		Behavioral	All	
			ASR	Trans		ASR	Trans	ASR	Trans		ASR	Trans
(a)	MLE	User-Indep	13.5					19.6				
(b)		User-Dep	20.2					27.9				
(c)	MLR	User-Indep	42.8	47.7	14.9	46.2†	48.8	46.4	51.3	18.7	50.1†	53.1
(d)		User-Dep	48.2	51.6	19.3	50.1†	52.8	52.1	55.5	25.2	53.9†	56.6
(e)	(c) + Personalized MF		47.6	49.1	16.4	50.3†§	53.5†§	51.1	53.3	20.3	54.2†§	57.6†§
(f)	(d) + Personalized MF		48.3	51.3	20.6	**51.9**†§	**54.0**†	52.7	55.4	26.7	**55.7**†§	**57.7**†§

number of apps per user is 15.8 and total number is 133.

To compare the performance between features, we show the results using lexical and behavioral features individually for intent prediction. We can see that lexical features achieve better performance than behavioral features alone, indicating that the majority of utterances contains explicit expressions that are predictable. In addition, combining behavior history patterns with lexical features performs best in terms of two measures. For ASR results, adding behavioral features significantly improves ACC from 48% to 50% and MAP from 52% to 54% (row (d)).

To evaluate the effectiveness of modeling latent semantics via MF, Table 1 also shows the performance of the MLR model integrated with personalized MF estimation in the rows (e)-(f)[2]. For both user-independent and user-dependent MLR results, additional integrating the MF model outperforms almost all MLR models, especially for all features (significant improvement with $p < 0.05$ in t-test), because MF is able to model the latent semantics under lexical and behavioral patterns. However, for manual results, integrating with the MF model does not perform better when using only lexical features (from 51.6% to 51.3% on ACC and from 55.5% to 55.4% on MAP), probably because manually transcribed utterances contain more clear and explicit semantics, so that learning latent semantics does not improve the performance.

Comparing the results for ASR and manual transcripts, we observed that the performance is worse when the user utterances contain recognition errors, because the explicit expression contains more noises and they may result in worse prediction. The benefit of MF techniques on ASR results is more significant (3.6% and 3.3% relative improvement of ACC and MAP respectively) than under manual transcripts, showing the effective capability of MF about modeling latent semantics in noisy data. Additionally, user-dependent results are better than user-independent results, showing that different users use mobile phones in a different ways, so modeling personalized SLU is able to effectively improve intent prediction. Finally, our experiments show the feasibility of disambiguating spoken language inputs for better intent prediction via behavioral patterns. The best prediction on ASR reaches 51.9% of ACC and 55.7% of MAP.

5. CONCLUSION

This paper presents an MF model that exploits both lexical and behavioral features for SLU in dialogue systems. The proposed model considers implicit semantics to enhance intent inference given the noisy ASR inputs. Also we are able to model users' behavioral patterns and their app preference, and further to better predict user intents in a smart-phone intelligent assistant setting (e.g. requesting an app). The resulting multi-model personalized system effectively improves intent prediction performance, achieving about 52% on turn accuracy and 56% on mean average precision for ASR transcripts with 25% word error rate.

[2]Using an MF model alone does not perform better compared to MLR (performing around 20-30% on ACC and MAP), because MF takes latent information into account and has weaker capability of modeling explicit observations.

6. REFERENCES

[1] A. Bhargava, A. Celikyilmaz, D. Hakkani-Tur, and R. Sarikaya. Easy contextual intent prediction and slot detection. In *ICASSP*, 2013.

[2] A. Celikyilmaz, Z. Feizollahi, D. Hakkani-Tür, and R. Sarikaya. Resolving referring expressions in conversational dialogs for natural user interfaces. In *EMNLP*, 2014.

[3] Y.-N. Chen and A. I. Rudnicky. Dynamically supporting unexplored domains in conversational interactions by enriching semantics with neural word embeddings. In *SLT*, 2014.

[4] Y.-N. Chen, W. Y. Wang, A. Gershman, and A. I. Rudnicky. Matrix factorization with knowledge graph propagation for unsupervised spoken language understanding. In *ACL-IJCNLP*, 2015.

[5] M. Collins, S. Dasgupta, and R. E. Schapire. A generalization of principal components analysis to the exponential family. In *NIPS*, 2001.

[6] Z. Gantner, S. Rendle, C. Freudenthaler, and L. Schmidt-Thieme. MyMedialite: A free recommender system library. In *RecSys*, 2011.

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

[8] Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *Computer*, (8):30–37, 2009.

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

[10] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme. BPR: Bayesian personalized ranking from implicit feedback. In *UAI*, 2009.

[11] C. Shin, J.-H. Hong, and A. K. Dey. Understanding and prediction of mobile application usage for smart phones. In *UbiComp*, 2012.

[12] P. Stenetorp, S. Pyysalo, G. Topić, T. Ohta, S. Ananiadou, and J. Tsujii. BRAT: a web-based tool for NLP-assisted text annotation. In *EACL*, 2012.