Matrix Factorization with Domain Knowledge and Behavioral Patterns for Intent Modeling

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

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

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

Abstract

Spoken language interfaces are being incorporated into various devices such as smart-phones and TVs. However, dialog systems will fail to respond correctly when users request functionality not supported by currently installed apps. We propose a feature-enriched matrix factorization (MF) approach to model open domain intents that allow a system to dynamically add app-relevant domains according to users' requests. We use MF to jointly model published app descriptions and users' spoken requests; this generates latent feature vectors for utterances and user intents without need for prior annotation. The matrix can further incorporate user behavioral patterns found in their activity logs to learn user-specific intent prediction models. We show that the MF models enriched with multimodality significantly improve the intent prediction, achieving 34% and 55% of mean average precision (MAP) for unsupervised single-turn requests and for supervised multi-turn interactions on ASR transcripts respectively.

1 Introduction

Spoken dialogue systems (SDS) are appearing on smart-phones and allow users to launch applications (apps) via spontaneous speech. Typically, an SDS requires predefined domain knowledge to understand corresponding functions, where the key component of an SDS is a spoken language understanding (SLU) model that maps utterances into actions; for example, after listening to "*drive me to cmu*", the system may predict that the user requires navigation and automatically launches the corresponding app "MAPS" to provide better interactions. Such apps support a *single-turn request task*.

To design the SLU module of an SDS, most previous studies relied on predefined ontology to train the decoder [1, 2, 3, 4]. However, these predefined ontologies may bias the subsequent user data collection process, and incur the cost of manually labeling utterances and updating the ontology. This problem recently leads to the development of unsupervised SLU techniques [5, 6, 7, 8, 9]. Chen *et al.* proposed a frame-semantics based framework for automatically inducing semantic slots given raw speech audio [7]. A knowledge graph resource was used to train models for intent detection in SLU, and results obtained from an unsupervised training process aligned well with the performance of traditional supervised learning [5]. Search engine logs and entity types from the knowledge graph were utilized to infer semantics and help improve the slot-filling performance in a movie domain [10, 11]. Such knowledge can be applied to domain expansion and supports open domain requests in SDSs [12, 13]. However, these approaches generally do not explicitly learn the latent factor representations that models the inference of hidden semantics. Considering that a user utterance "*i would like to contact alex*" includes explicit semantic information about "*contact*" in its surface patterns, 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. To provide better interactions with users, modeling the hidden intent helps predict the users' desired apps. Traditional SLU models use discriminative classifiers to predict whether the predefined slots occur in the utterances or not and ignore hidden semantic information.

In addition to the difficulty caused by language ambiguity, behavioral patterns also influence the user intents. Typical intelligent assistants (IAs) treat each task (e.g. restaurant search, messaging, etc) independent of each other, where only users' current utterances are considered to decide the desired apps for SLU [13]. Some IAs model user intents by using the contextual utterances, but they do not take into account the behavioral patterns of individual users [14]. This work improves the intent prediction based on our observation that the intended apps usually depend on 1) individual preference (some people prefer MESSAGE to EMAIL) and 2) behavioral patterns at the app level (MESSAGE is more likely to follow CAMERA, and EMAIL is more likely to follow EXCEL). We refer to it as *multi-turn interaction* task.

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 [15, 16]. 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 intents [17, 18]. For example, if a 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*".

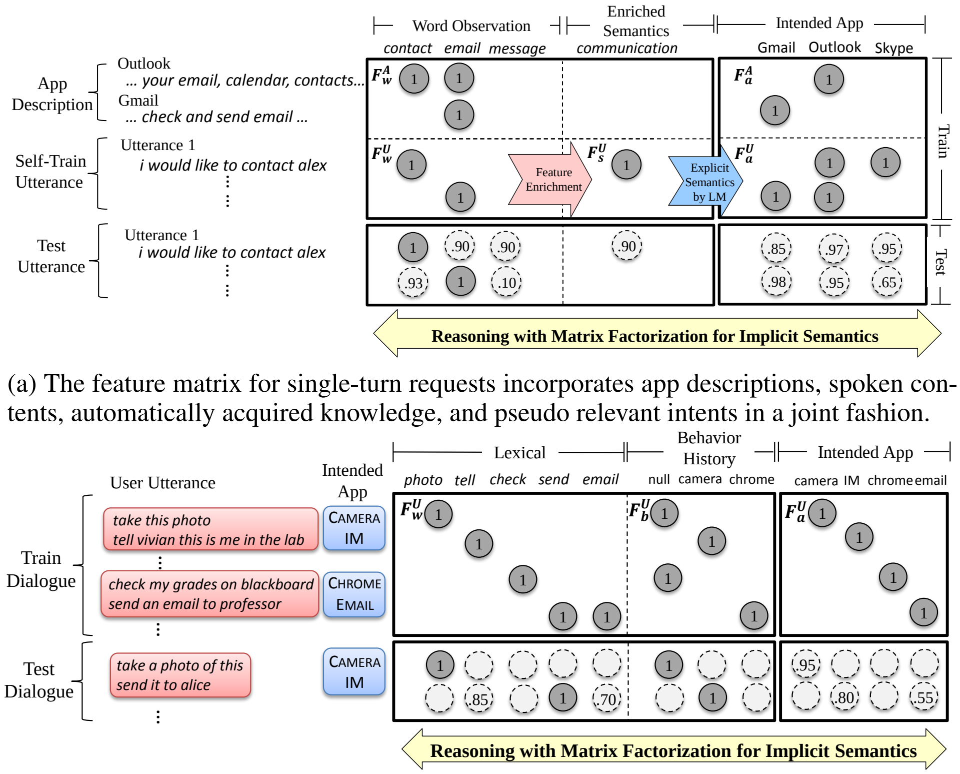
Therefore, this paper proposes a feature-enriched matrix factorization (MF) model to learn low-ranked latent features for SLU, where the unobserved patterns can be considered [19]. For the single-turn request task, the model takes account of app descriptions, observed spoken utterances, and automatically acquired domain knowledge to predict intents in a joint fashion. For the multi-turn interaction task, the model incorporates contextual behavior history along with lexical observations to improve intent prediction. We evaluate the performance by examining whether predicted apps can satisfy users' requests. The experiments show that our feature-enriched MF approach can model user intents and allow an SDS to provide better responses for both unsupervised single-turn requests and supervised multi-turn interactions. Our contributions are four-fold:

- This is among the first attempts to apply feature-enriched MF techniques for intent modeling, incorporating different sources of rich information (app description, semantic knowledge, behavioral patterns);
- The feature-enriched MF approach jointly models spoken observations, available text information, and structured knowledge to infer user intents for single-turn requests, taking hidden semantics into account;
- The behavioral patterns can be incorporated into the feature-enriched MF approach to model user preference for personalized understanding in multi-turn interactions;
- Our experimental results indicate that feature-enriched MF approaches outperform most strong baselines and achieve better intent prediction performance of both single-turn requests and multi-turn interactions.

2 User Intent Prediction by Matrix Factorization

Under the app-oriented SDS, the main idea is to predict user intents along with corresponding apps. For single-turn requests, given a user's spoken utterance, how can an SDS dynamically support functions corresponding to requests beyond predefined domains in an unsupervised manner [13]? For multi-turn interactions, the goal is to predict the apps that are more likely to be used to handle the user requests given input utterances and behavioral contexts, considering not only the desired functionality but also user preference. We build an SLU component to model user intents: we frame the task as a multi-class classification problem, where we estimate the probability of each intent/app a given an utterance u, $P(a \mid u)$, using a proposed feature-enriched MF approach.

An MF model considers the unobserved patterns and estimates their probabilities instead of viewing them as negative, allowing it to model the implicit information [19]. Given the benefits brought by MF techniques, including 1) modeling the noisy data, 2) modeling hidden semantics, and 3) modeling the long-range dependencies between observations, in this work we apply an MF approach



(a) The feature matrix for single-turn requests incorporates app descriptions, spoken contents, automatically acquired knowledge, and pseudo relevant intents in a joint fashion.

(b) The feature matrix for multi-turn interactions incorporates lexical and behavioral patterns to build the personalized model.

Figure 1: Our MF method completes a partially-missing matrix to factorize the low-rank 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 rich features of utterances.

to intent modeling for SDSs. First we define $\langle x, y \rangle$ as a *fact*, which refers to an entry in a matrix. The input of our model is a set of observed facts $\mathcal{O}$, and the observed facts for a given utterance is denoted by $\{\langle x, y \rangle \in \mathcal{O}\}$, where y can be a n-gram observation, a enriched semantic concept, or an intended app. The goal of our model is to estimate, for a given utterance x and an app-related intent 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 app for supporting the utterance x. Similarly, a series of exponential family models are introduced to estimate the probability using a natural parameter $\theta_{x,y}$ and the logistic sigmoid function:

$$P(M_{x,y} = 1 \mid \theta_{x,y}) = \sigma(\theta_{x,y}) = \frac{1}{1 + \exp(-\theta_{x,y})} \tag{1}$$

For single-turn requests and multi-turn interactions, we construct a matrix M as observed facts using different types of enriched features, and the matrix can then be factorized by a matrix completion technique with the assumption that the matrix is low-rank.

2.1 Feature-Enriched Matrix Construction

For each of single-turn request and multi-turn interaction tasks, we construct a feature-enriched matrix below. The illustration of two matrices is shown in Figure 1. They are enriched with various modalities. For unsupervised single-turn requests, the matrix in Figure 1 (a) incorporates word observations, enriched semantics, and pseudo relevant apps for intent modeling. For supervised multi-turn interactions, the matrix in Figure 1 (b) models word observations and contextual behavior for intent prediction. Below we use a principle model, which contains three sets of information, low-level spoken features (word observation matrix), high-level semantic features (enriched semantics matrix), and intent results (intent matrix), to model user intents.

2.1.1 Word Observation Matrix

A word observation matrix features with binary values based on n-gram word patterns. For single-turn requests, two word observation matrices are built, where F_w^A is for textual app descriptions and F_w^U is for spoken utterances. Each row in the matrix represents an app/utterance and each column refers to an observed word pattern. In other words, F_w^A and F_w^U carry the basic word vectors for all apps and all utterances respectively. Similarly, for multi-turn interactions, a word observation matrix, F_w^U, is constructed for spoken utterances. The left-most column set in Figure 1 illustrates the lexical features for the given utterances.

2.1.2 Enriched Semantics Matrix

For single-turn requests, considering to include open domain knowledge based on the user's utterance, we utilize distributed word representations to capture syntactic and semantic relationship for acquiring domain knowledge [13, 9].

- Embedding-based semantics: We enrich the original utterances with semantically similar words, where the similarity is measured by CBOW word embeddings trained on the app descriptions [20, 13].
- Type-embedding-based semantics: The concept types are additionally included to further expand the semantic information. For example, "*play lady gaga's bad romance*" may contain the types "singer" and "song" to improve semantic inference (domain-related cues about music playing), where we detect all entity mention candidates in the given utterances and use entity linking with Freebase and Wikipedia to mine entity types [13]. Then an enriched semantics matrix can be built as F_s^U, where each row is a utterance and each column corresponds a semantic element shown in Fig. 1.

For multi-turn interactions, we enrich the utterance with contextual behaviors to incorporate behavioral information into personalized and context-aware intent modeling. Figure 1 (b) illustrates the enriched behavioral features as F_b^U, where the second utterance "*tell vivian this is me in the lab*" involves "CAMERA" acquired from the previous turn "*take this photo*". The behavioral history at turn t, h_t, can be formulated as $\{a_1, ..., a_{t-1}\}$, which is the set of apps that were previously launched in the ongoing dialogue. Note that multi-turn interaction uses supervised labels, where intended apps are given during training.

2.1.3 Intent Matrix

To link the word patterns with the corresponding intent, an intended app matrix F_a^A is constructed, where each column corresponds to launching a specific app. Hence, the entry is 1 when the app and the intent correspond to each other, and 0 otherwise,

For unsupervised single-turn requests, to induce the user intent, we use a basic retrieval model for returning the top K relevant apps for each utterance u, and treat them as pseudo intended apps [13]. Figure 1 (a) includes an example of utterance "*i would like to contact alex*", where the utterance is treated as a request to search for relevant apps such as "OUTLOOK" and "SKYPE". Then we build an app matrix F_a^U with binary values based on the top relevant apps, which also denotes intent features for utterances. Note that we do not use any annotations, the app-related intents are returned by a retrieval model and may contain some noise.

For personalized intent prediction on multi-turn interactions, the intent matrix can be directly acquired from users' app usage logs. F_a^U can be built and illustrated in the right part of matrix from Figure 1 (b).

2.1.4 Integrated Model

As shown in Figure 1, we integrate word matrices, an enriched semantics matrix, and intent matrices from both apps and utterances together for training the MF model. The integrated model for single-turn requests can be formulated as

$$M = [\begin{array}{ccc} F_w^A & 0 & F_a^A \\ F_w^U & F_s^U & F_a^U \end{array}]. \tag{2}$$

Table 1: User intent prediction for single-turn requests on mean average precision (MAP) using different training features (%). LM is a baseline language modeling approach which models explicit semantics.

Feature Matrix		ASR		Manual	
		LM	w/ MF	LM	w/ MF
(a)	Word Observation	25.1	29.2 (+16.2%)	26.1	30.4 (+16.4%)
(b)	Word + Embedding-Based Semantics	32.0	**34.2** (+6.8%)	33.3	33.3 (-0.2%)
(c)	Word + Type-Embedding-Based Semantics	31.5	32.2 (+2.1%)	32.9	**34.0** (+3.4%)

Similarly, the integrated matrix for multi-turn interactions can be built as $M = [F_w^U \; F_s^U \; F_a^U]$. Hence, the relations among word patterns, domain knowledge, and behaviors can be automatically learned from the integrated model in the joint fashion. The goal of the MF model is, for a given user utterance, to predict the probability that the user intents to launch each app.

2.2 Optimization Procedure

With the built matrix, M, we can learn a model θ^* that can best estimate the observed patterns by parametrizing the matrix through weights and latent component vectors, where the parameters are estimated by maximizing the log likelihood of observed data [21].

$$\begin{aligned} \theta^* &= \arg\max_{\theta} \prod_{x \in X} P(\theta \mid M_x) = \arg\max_{\theta} \prod_{x \in X} P(M_x \mid \theta) \cdot P(\theta) \\ &= \arg\max_{\theta} \sum_{x \in X} \ln P(M_x \mid \theta) - \lambda_\theta = \sum_{f^+ \in \mathcal{O}} \sum_{f^- \notin \mathcal{O}} \ln \sigma(\theta_{f^+} - \theta_{f^-}) - \lambda_\theta, \end{aligned} \tag{3}$$

where X is a set indicating row information. For single-turn requests, M_x is a row vector corresponding either an app or an utterance; for multi-turn interactions, M_x corresponds to an utterance. Here the assumption is that each row (app/utterance) is independent of others. To avoid treating unobserved facts as designed negative facts and to complete the missing entries of the matrix, our model can be factorized by a matrix completion technique with a low-rank assumption [22, 23], where we use a variant of the ranking: giving observed true facts $\langle x, y^+ \rangle$ higher scores than unobserved (true or false) facts $\langle x, y^- \rangle$ from observations $\mathcal{O}$ constructed from M to parameterize the given integrated model by performing an SGD update [23].

3 Experiments

For single-turn requests, total 195 utterances were collected for 13 domains, which are representatives of most frequently used goals, including “navigation”, “email writing”, “music playing”, etc [13]. Using Google Speech API, the word error rate (WER) is 19.8% . The average word count of an utterance is 6.8 for ASR outputs and 7.2 for manual transcripts, which suggests the challenge of retrieving relevant apps given limited information in a request. The apps for returning were collected from Google Play in November 2012. Each Android app in Google Play has its own description page and metadata (name, number of downloads, content description, etc.) Total 1,881 apps with more than million downloads were considered. For evaluation, judges manually identified apps from Google Play that could support the corresponding tasks. We used the judge-labeled apps as ground truth for evaluating predicted apps and reported standard information retrieval metrics, mean average precision (MAP).

For multi-turn interactions, we collected 533 multi-app spoken dialogs with 1607 utterances (about 3 user utterances per dialog). Among these dialogs, we have 455 multi-turn dialogs (82.3%), providing behavioral information. Using Google Speech API, the WER is 22.7%. For each subject the (chronologically ordered) data were split 70% for training and 30% for testing in the experiments. For each user, we built a personalized SLU model with his/her own training data. We also compute MAP to evaluate the ranked app lists.

Table 1 and Table 2 present the results using different features before and after feature enrichment and integration of the MF model on ASR and manual transcripts for different tasks. In single-turn requests, LM is a baseline language modeling retrieval approach, where $P(a \mid u)$ is estimated

Table 2: User intent prediction for multi-turn interactions on mean average precision (MAP) using different training features (%). MLR is a multi-class baseline for modeling explicit semantics.

Feature Matrix		ASR		Manual	
		MLR	w/ MF	MLR	w/ MF
(d)	Word Observation	52.1	52.7 (+1.2%)	55.5	55.4 (-0.2%)
(e)	Behavioral Patterns	25.2	26.7 (+6.0%)	25.2	26.7 (+6.0%)
(f)	Word Observation + Behavioral Patterns	53.9	**55.7** (+3.3%)	56.6	**57.7** (+1.9%)

based on the probability that user speaks the utterance u to make the request for launching the app a [13]. In multi-turn interactions, MLR is a standard multinomial logistic regression model, where $P(a \mid u)$ is estimated according to the observed training data. It can be found that almost all results are improved after combining with the MF model, where the scores from the baseline and the MF model are averaged, indicating that the hidden semantics modeled by MF techniques helps estimate the intent probability.

In single-turn requests, for ASR results, enriching semantics using embedding-based (row (b)) and type-embedding-based semantics (row (c)) significantly improve the baseline performance (row (a)) using the basic retrieval model, where the MAP performance is from 25% to 31%. Then the performance can be further improved by integrating MF to additionally model hidden semantics, where row (b) achieves 34.2% on MAP. The reason why the type-embedding-based semantics (row (c)) does not perform better compared with embedding-based semantics (row (b)) is that the automatically acquired type information appears to introduce noise, and row (c) is slightly worse than row (b) for ASR results. For manually transcribed speech, the semantic enrichment procedure and MF models also improve the performance. Different from ASR results, the best result for user intent prediction is based on the features enriched with type-embedding-based semantics (row (c)), achieving 34.0% on MAP. The reason may be that manual transcripts are more likely to capture the correct semantic information by word embeddings and have more consistent type information, allowing the MF technique to model user intents better.

In multi-turn interactions, comparing between lexical features (row (d)) and behavioral features (row (e)), lexical features capture more informative cues for intent prediction. For both ASR and manual transcripts, enriching word features with behavioral patterns (row (f)) significantly outperform the original lexical features alone. Additionally integrating with MF models further improves the performance, achieving 55.7% and 57.7% on MAP for ASR and manual results respectively.

In sum, the results show that the rich features carried by app descriptions and utterance-related contents can help intent prediction in single-turn requests using proposed model for most cases. Also, the features involving behavioral patterns improve intent prediction in multi-turn interaction through the proposed approach. The evaluation results also prove the effectiveness of our feature-enriched MF models, which incorporate the enriched semantics and model the implicit semantics along with explicit semantics in a joint fashion to significantly improve the performance of intent prediction.

4 Conclusions

This paper proposes a feature-enriched matrix factorization approach to learn user intents based on the automatically acquired rich features, in one case taking account into domain knowledge and in another case behavioral patterns along with users' utterances. In a smart-phone intelligent assistant setting (e.g. requesting an app), the proposed model considers implicit semantics to enhance intent inference given the noisy ASR inputs for single-turn request dialogues. The model is also able to incorporate users' behavioral patterns and their app preferences to better predict user intent in multi-turn interactions. We believe that this approach allows systems to handle users' open domain intents when retrieving relevant apps that provide desired functionality either locally available or by suggesting installation of suitable apps and doing so in an unsupervised way. The framework can extend to incorporate personal behavior history and use it to improve a system's ability to assist users pursuing multi-app activities. In sum, the effectiveness of the feature-enriched MF model can be shown in different domains, indicating good generality and providing a reasonable direction for the future work.

References

[1] Stephanie Seneff. TINA: A natural language system for spoken language applications. *Computational Linguistics*, 18(1):61–86, 1992.

[2] John Dowding, Jean Mark Gawron, Doug Appelt, John Bear, Lynn Cherny, Robert Moore, and Douglas Moran. Gemini: A natural language system for spoken-language understanding. In *Proceedings of ACL*, 1993.

[3] Narendra Gupta, Gokhan Tur, Dilek Hakkani-Tur, Srinivas Bangalore, Giuseppe Riccardi, and Mazin Gilbert. The AT&T spoken language understanding system. *IEEE Transactions on Audio, Speech, and Language Processing*, 14(1):213–222, 2006.

[4] Dan Bohus and Alexander I Rudnicky. The RavenClaw dialog management framework: Architecture and systems. *Computer Speech & Language*, 23(3):332–361, 2009.

[5] Larry Heck and Dilek Hakkani-Tür. Exploiting the semantic web for unsupervised spoken language understanding. In *Proceedings of SLT*, 2012.

[6] Larry P Heck, Dilek Hakkani-Tür, and Gokhan Tur. Leveraging knowledge graphs for web-scale unsupervised semantic parsing. In *Proceedings of INTERSPEECH*, 2013.

[7] Yun-Nung Chen, William Yang Wang, and Alexander I Rudnicky. Unsupervised induction and filling of semantic slots for spoken dialogue systems using frame-semantic parsing. In *Proceedings of ASRU*, 2013.

[8] Yun-Nung Chen, Dilek Hakkani-Tür, and Gokhan Tur. Deriving local relational surface forms from dependency-based entity embeddings for unsupervised spoken language understanding. In *Proceedings of SLT*, 2014.

[9] Yun-Nung Chen, William Yang Wang, and Alexander I Rudnicky. Leveraging frame semantics and distributional semantics for unsupervised semantic slot induction in spoken dialogue systems. In *Proceedings of SLT*, 2014.

[10] Gokhan Tur, Minwoo Jeong, Ye-Yi Wang, Dilek Hakkani-Tür, and Larry P Heck. Exploiting the semantic web for unsupervised natural language semantic parsing. In *Proceedings of INTERSPEECH*, 2012.

[11] Dilek Hakkani-Tür, Asli Celikyilmaz, Larry Heck, Gokhan Tur, and Geoff Zweig. Probabilistic enrichment of knowledge graph entities for relation detection in conversational understanding. In *Preceedings of INTERSPEECH*, 2014.

[12] Ali El-Kahky, Derek Liu, Ruhi Sarikaya, Gokhan Tur, Dilek Hakkani-Tür, and Larry Heck. Extending domain coverage of language understanding systems via intent transfer between domains using knowledge graphs and search query click logs. In *Proceedings of ICASSP*, 2014.

[13] Yun-Nung Chen and Alexander I. Rudnicky. Dynamically supporting unexplored domains in conversational interactions by enriching semantics with neural word embeddings. In *Proceedings of SLT*, 2014.

[14] Anshuman Bhargava, Asli Celikyilmaz, Dilek Hakkani-Tur, and Ruhi Sarikaya. Easy contextual intent prediction and slot detection. In *Proceedings of ICASSP*, 2013.

[15] Dilek Hakkani-Tür, Malcolm Slaney, Asli Celikyilmaz, and Larry Heck. Eye gaze for spoken language understanding in multi-modal conversational interactions. In *Proceedings of ICMI*, 2014.

[16] Spyros Kousidis, Casey Kennington, Timo Baumann, Hendrik Buschmeier, Stefan Kopp, and David Schlangen. A multimodal in-car dialogue system that tracks the driver's attention. In *Proceedings of ICMI*, 2014.

[17] Choonsung Shin, Jin-Hyuk Hong, and Anind K Dey. Understanding and prediction of mobile application usage for smart phones. In *Proceedings of UbiComp*, 2012.

[18] Asli Celikyilmaz, Zhaleh Feizollahi, Dilek Hakkani-Tür, and Ruhi Sarikaya. Resolving referring expressions in conversational dialogs for natural user interfaces. In *Proceedings of EMNLP*, 2014.

[19] Yun-Nung Chen, William Yang Wang, Anatole Gershman, and Alexander I. Rudnicky. Matrix factorization with knowledge graph propagation for unsupervised spoken language understanding. In *Proceedings of ACL-IJCNLP*, 2015.

[20] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Proceedings of NIPS*, 2013.

[21] Michael Collins, Sanjoy Dasgupta, and Robert E Schapire. A generalization of principal components analysis to the exponential family. In *Proceedings of NIPS*, 2001.

[22] Raghunandan H Keshavan, Andrea Montanari, and Sewoong Oh. Matrix completion from a few entries. *Information Theory, IEEE Transactions on*, 56(6):2980–2998, 2010.

[23] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. BPR: Bayesian personalized ranking from implicit feedback. In *Proceedings of UAI*, 2009.