UNSUPERVISED USER INTENT MODELING BY FEATURE-ENRICHED MATRIX FACTORIZATION

Yun-Nung Chen   Ming Sun   Alexander I. Rudnicky   Anatole Gershman

Carnegie Mellon University, School of Computer Science
5000 Forbes Ave., Pittsburgh, PA 15213
{yvchen, mings, air, anatoleg}@cs.cmu.edu

ABSTRACT

Spoken language interfaces are being incorporated into various devices such as smart phones and TVs. However, dialogue systems may fail to respond correctly when users' request functionality is not supported by currently installed apps. This paper proposes a feature-enriched matrix factorization (MF) approach to model open domain intents, which allows a system to dynamically add unexplored domains according to users' requests. First we leverage the structured knowledge from Wikipedia and Freebase to automatically acquire domain-related semantics to enrich features of input utterances, and then MF is applied to model automatically acquired knowledge, published app textual descriptions and users' spoken requests in a joint fashion; this generates latent feature vectors for utterances and user intents without need of prior annotations. Experiments show that the proposed MF models incorporated with rich features significantly improve intent prediction, achieving about 34% of mean average precision (MAP) for both ASR and manual transcripts.

Index Terms— Spoken language understanding (SLU), spoken dialog system (SDS), matrix factorization (MF), distributional semantics, enrichment.

1. INTRODUCTION

Spoken dialogue systems (SDS) are recently appearing on smartphones and allow users to launch applications (apps) via spontaneous speech. Typically, an SDS requires a predefined domain ontology to understand corresponding functions, such as alert_clock_setting (CLOCK) and navigation (MAPS) [1]. The key component of an SDS is a spoken language understanding (SLU) module that maps utterances into intents; for example, after hearing "*drive me to CMU*", the system may predict that the user requires navigation and then automatically launches the corresponding app to provide better interactions. To design the SLU module of an SDS, most of previous studies relied on the predefined ontology to train the decoder [2, 3, 4, 5, 6, 7]. However, these predefined knowledge bases may bias the subsequent user data collection process, and incur the cost of manually labeling utterances and updating the ontologies.

In recent years, this issue leads to development of unsupervised SLU techniques [8, 9, 10, 11, 12]. Chen *et al.* proposed a frame-semantics based framework for automatically inducing semantic slots given raw speech audio [10, 12, 13, 14]. 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 [8]. Tur *et al.* also showed that search engine logs and entity types from the knowledge graph can be used to infer implicit semantics and help improve slot-filling performance in a movie domain [15, 16]. Such knowledge can be applied to domain expansion and supports open domain requests in SDSs [1, 17, 18].

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 semantics such as "*message*" and "*email*", because the user likely intends to launch apps like MESSENGER (message) or OUTLOOK (email) even though they are not directly observed in the surface patterns. The prior work only considered explicit information to retrieve apps that are able to support the requests, where the unobserved concepts were not involved [1, 19, 20]. Such hidden semantics was shown to be useful for learning better SLU models and can be captured by matrix factorization (MF) techniques [21].

Therefore, instead of using discriminative classifiers to predict whether predefined slots occur in the utterances, this paper utilizes a similar idea to model implicit relations among various types of features, including word patterns, acquired knowledge, and intentions, in order to accurately infer user intents and provide better interactions with users. Specifically, this paper proposes a feature-enriched MF to learn low-ranked latent features for SLU, taking multimodal features into account [22]. This model incorporates unobserved features and estimates their probabilities instead of viewing them as negative instances, where hidden semantics can be remained for SLU to better predict intents [21, 22].

We evaluate the performance by examining whether predicted apps can satisfy users' requests. The experiments show that our MF-based approach can model user intents and allow an SDS to provide better responses for unsupervised single-turn requests. Our contributions are three-fold:

- This is among the first attempts to apply feature-enriched MF techniques for intent modeling, incorporating different sources of modalities;
- The MF approach jointly models spoken observations and available textual information, and learns implicit semantics based on feature relations;
- Our empirical results indicate that our feature-enriched MF approaches outperform most of strong baselines and achieve better intent prediction performance.

2. USER INTENT MODELING

Under an app-oriented SDS, the main idea is to model user intents [1, 22]. Given a user's spoken utterance, how can an SDS dynamically support functions corresponding to requests beyond predefined domains in an unsupervised manner [1]? Therefore, a such system is

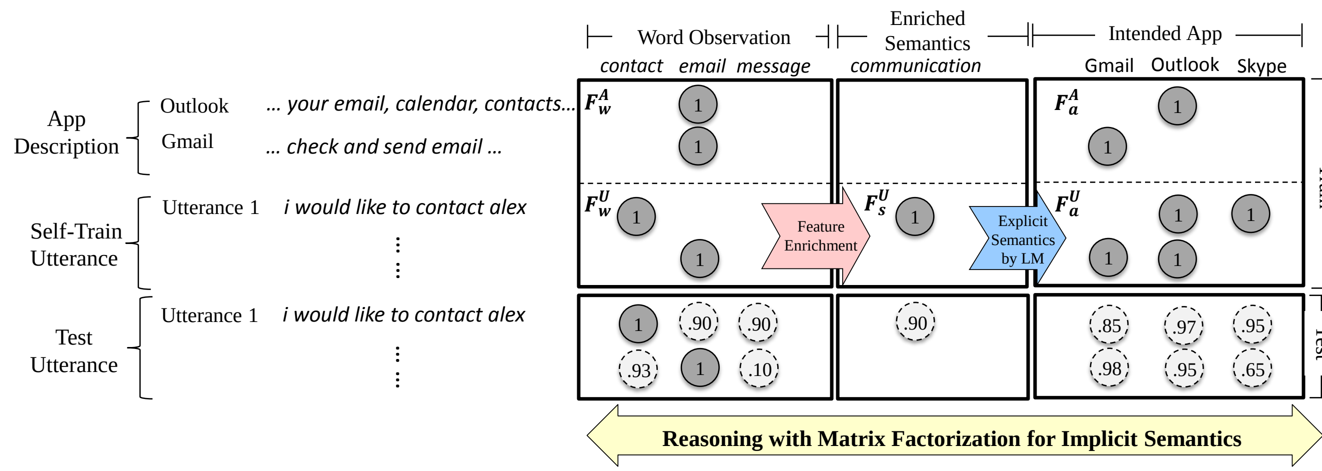

Fig. 1. The feature matrix incorporates app descriptions and utterance contents in a joint fashion. Feature enrichment extracts the domain knowledge given the utterances (middle set of columns), and language modeling (LM) technique is applied to model explicit semantics (right set of columns). Our MF 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 corresponding to the current user utterance.

able to provide more flexible communication after overcoming domain restrictions. The typical retrieval approach can be applied to the problem and then generates a first-pass app ranking list. With the pseudo positive results, a feature-enriched MF model is proposed to estimate implicit intents based on app textual descriptions, observed spoken utterances, automatically acquired knowledge, and user intentions in a jointly fashion. The models focus on estimating the probability of an app a being launched given a currently observed utterance u, $P(a \mid u)$, for intent prediction.

3. FEATURE-ENRICHED MATRIX FACTORIZATION

A matrix factorization (MF) technique has been explored in different domains, which models latent semantics under a low-rank assumption [23, 21]. Considering to 1) model noisy data, 2) model hidden semantics, and 3) model long-range dependencies between observations, this work applies an MF approach 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 $\{\langle x, y \rangle \in \mathcal{O}\}$. 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. We introduce a series of exponential family models that estimate the probability using a natural parameter $\theta_{x,y}$ and the logistic sigmoid function:

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

We construct a matrix M with observed facts, and then factorize it by a matrix completion technique with the low-rank assumption, where the representations of utterances and features can be parameterized.

3.1. Feature Model Construction

The constructed feature matrix is illustrated in Fig. 1, which is enriched with various modalitieis: word observations, enriched semantics, and pseudo relevant apps for intent modeling.

Algorithm 1 Semantics Enrichment Procedure

Require: a word observation set W in the utterance; a vocabulary V; a word relatedness function $f_s(\cdot)$
Ensure: a set of enriched semantics S

```
Initializing S* = {};
for all w ∈ W do
    Extracting the words with similarity higher than a threshold
    from the vocabulary, V* = {v | f_s(w, v) ≥ δ, v ∈ V};
    Enrich the semantic set S* ← S* ∪ V*
end for
return S*;
```

3.1.1. Word Observation Matrix

A word observation matrix features with binary values based on n-gram word patterns. 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 basic word vectors for all apps and all utterances respectively.

3.1.2. Enriched Semantics Matrix

In order to incorporate open domain knowledge based on the user's utterance, we utilize distributed word representations to capture syntactic and semantic relationships for knowledge acquisition [1, 12].

- Embedding-based semantics: We enrich original utterances with semantically similar words, where the similarity is measured by word embeddings trained on app descriptions [24, 1]. Algorithm 1 shows the procedure of acquiring domain knowledge for semantics enrichment.
- Type-embedding-based semantics: In addition to semantically similar words, types of concepts are included to further expand the semantic information. For example, "*play lady gaga's bad romance*" may contain the types "singer" and "song" (domain-related cues about music playing), so that we can improve semantic inference by detecting all entity mention candidates in

the given utterances and using entity linking with Freebase and Wikipedia to mine entity types [1].

- Wikipedia page linking: For each entity mention from the given utterance, we output a set of linked Wikipedia pages, where an Integer Linear Programming (ILP) formulation generates the mapping from mentions to Wikipedia pages [25, 26]. For each entity, we extract its definition sentence from the linked page, and then all words parsed into adjectives or nouns in the noun phrase just following the part-of-speech pattern (VBZ) (DT) such as "*is a/an/the*" are extracted as semantic concepts. For example, the sentence about the entity "*lady gaga*" is "*Stefani Joanne Angelina Germanotta, better known by her stage name Lady Gaga, is an American singer and songwriter.*", and the entity types, "*American singer*" and "*songwriter*", are extracted.
- Freebase list linking: Each mention can be linked to a ranked list of Freebase nodes by Freebase API[1], and we extract top K notable types for each entity as the acquired knowledge.

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. The illustration is shown in Fig. 1.

3.1.3. Intent Matrix

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

To induce user intents, we use a retrieval model for returning top K relevant apps for each utterance u, and treat them as pseudo relevant app behaviors [1], which is detailed in Section 4. Fig. 1 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 returned apps for denoting intent features of utterances. Note that we do not use any annotations, the app-related intents are returned by a retrieval model and may contain some noises.

3.1.4. Integrated Model

As shown in Fig. 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 can be formulate as

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

Hence, the relations among word patterns, domain knowledge, and intents can be automatically learned from the integrated model. The goal of the MF model is, for a given user utterance, to predict the probability that the user intends to launch each app.

3.2. Optimization Procedure

With the built matrix, 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 from M [27].

$$\begin{aligned} \theta^* &= \arg\max_\theta \prod_{x \in U} P(\theta \mid M_x) \quad (3) \\ &= \arg\max_\theta \prod_{x \in U} P(M_x \mid \theta) \cdot P(\theta) \\ &= \arg\max_\theta \sum_{x \in U} \ln P(M_x \mid \theta) - \lambda_\theta, \end{aligned}$$

where M_x is a row vector corresponding to the utterance x in M, because we assume that each utterance is independent of others.

To complete the missing entries of the matrix, our model can be factorized by a matrix completion technique with a low-rank assumption, which uses a variant of the ranking: giving observed true facts higher scores than unobserved (true or false) facts to factorize the given matrix [21, 28, 29, 30]. To estimate the parameters in (3), we create a dataset of *ranked pairs* from M: for each app/utterance x and each observed fact $f^+ = \langle x, y^+ \rangle$, we choose each intent y^- referring to the app that does not correspond to x, or the app that is not returned as by the retrieval model according to the utterance x. Then for each pair of facts f^+ and f^-, 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). Our objective maximizes the summation of each ranked pair:

$$\sum_{x \in U} \ln P(M_x \mid \theta) = \sum_{f^+ \in \mathcal{O}} \sum_{f^- \notin \mathcal{O}} \ln \sigma(\theta_{f^+} - \theta_{f^-}). \quad (4)$$

The objective is an approximation to the per utterance AUC (area under the ROC curve), which correlates with well-ranked apps per utterance. 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 (f^+, f^-). For each pair, we perform an stochastic gradient descent (SGD) update using the gradient of the corresponding objective function for MF [31].

Finally we can obtain the estimated probabilities of various features given the current utterance, which includes probabilities of intended apps given an utterance, $P(a \mid u)$. For our task, Fig. 1 shows that the hidden semantics, "*message*", "*email*", and "*communication*", are inferred from "*i would like to contact alex*" because semantic relations between various features are captured by the model.

4. MOBILE APP PREDICTION

For each test utterance u, with the trained MF model, we can predict the probability of each app a based on the observed features corresponding to the current utterance by taking into account two models, 1) a baseline model for explicit semantics and 2) an MF-based model for implicit semantics:

$$\begin{aligned} P(a \mid u) &= P_{\text{exp}}(a \mid u) \times P_{\text{imp}}(a \mid u), \quad (5) \\ &= P_{\text{exp}}(a \mid u) \times P(M_{u,a} = 1 \mid \theta), \end{aligned}$$

where $P(a \mid u)$ is an integrated probability for ranking apps, $P_{\text{exp}}(a \mid u)$ is the probability outputted by the baseline model that considers explicit semantics, and $P_{\text{imp}}(a \mid u)$ is the probability estimated by the proposed feature-enriched MF model. The fused probabilities are able to consider hidden intents by learning latent semantics from enriched features.

[1]https://developers.google.com/freebase/

Table 1. User intent prediction on mean average precision (MAP) and precision at 10 (P@10) (%). LM is a baseline language modeling approach which models explicit semantics. The relative improvement is shown in parentheses.

Feature		ASR Transcripts				Manual Transcripts			
		MAP		P@10		MAP		P@10	
		LM	w/ MF	LM	w/ MF	LM	w/ MF	LM	w/ MF
(a)	Baseline: Word Observation	25.1	29.2 (+16.2%)	28.6	29.5 (+3.4%)	26.1	30.4 (+16.4%)	29.2	30.1 (+2.8%)
(b)	(a) + Embedding-Enrichment	**32.0**	**34.2** (+6.8%)	31.2	**32.5** (+4.3%)	**33.3**	33.3 (-0.2%)	32.0	33.0 (+3.4%)
(c)	(a) + Type-Embedding-Enrichment	31.5	32.2 (+2.1%)	**31.3**	30.6 (-2.3%)	32.9	**34.0** (+3.4%)	**32.5**	**34.7** (+6.8%)

For an unsupervised task of ranking apps based on user spoken requests, a language modeling retrieval technique is used for query likelihood estimation [32, 1], and app-related intents are ranked by

$$\begin{aligned} P_{\text{exp}}(a \mid u) &= \frac{P(u \mid a)P(a)}{P(u)} \\ &\propto P(u \mid a) = \frac{1}{|u|}\sum_{w \in u} \log P(w \mid a), \end{aligned} \tag{6}$$

where u is the user's query, a is an intended app, w represents the token in the utterance, and $P(u \mid a)$ represents the probability that user speaks the utterance u to make the request for launching the app a[2]. For example, in order to use the app GMAIL, a user is more likely to say "*compose an email to alex*", while the same utterance should correspond to a lower probability when launching the app MAPS. To estimate the likelihood by the language modeling approach, we use the description content of an app with an assumption that it carries semantically related information.

5. EXPERIMENTS

5.1. Corpus and Setup

With a view of expanding a set of domains for SDS interfaces, we identified the most popular apps available from a mobile app store, representative of important domains that users tend to access frequently; the defined domains were used to design our 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 fulfilling the task implied by the images. The corpus contains 195 utterances, and the word error rate is reported as 19.8% using Google Speech API. 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 an utterance.

The data to populate the database was collected from Google Play in November 2012. Total 140,854 apps were available; only apps with more than one 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) and precision at 10 (P@10).

5.2. Evaluation Results

Table 1 presents the results using different features before and after integrating with the MF model for ASR and manual transcripts. For ASR results, row (a) only takes word patterns as observations, and first-pass LM performs 25% on MAP and 28% on P@10. It can be found that combining with the standard MF model significantly improves the performance.

For the baseline LM technique, semantics enrichment acquires various domain knowledge and improves the performance (rows (b) and (c)), but this can be further improved by integrating with the feature-enriched MF model, achieving 34.2% on MAP and 32.5% on P@10 for ASR transcripts. However, the type-embedding-enrichment approach (row (c)) appears to introduce noises due to imperfect types that are automatically acquired, the performance does not show consistent improvement before and after combining with MF.

For manually transcribed speech, MF models improve the performance of original word patterns (row (a)). Different from ASR results, feature-enriched MF models are able to predict user intents better for both embedding-enriched and type-embedding-enriched approaches, especially for P@10. 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 MF to model user intents more accurately.

In sum, the experiments show that almost all results are improved after combining with MF-based models, indicating that hidden intents modeled by MF techniques help better estimate intent probabilities. Also, the results show that the rich features acquired by a semantics enrichment procedure can improve intent prediction for most cases, showing the effectiveness of proposed feature-enriched MF models. Comparing between best results on MAP for ASR and manual transcripts (34.2% and 34.0% respectively), it can be seen that MF can model not only hidden semantics but noisy data, and achieve good performance even on noisy ASR results.

6. CONCLUSION

This paper proposes a matrix factorization approach to learn user intents based on rich feature patterns from multiple modalities, which takes into account app descriptions, automatically acquired knowledge and user utterances. In a smart-phone intelligent assistant setting (e.g. requesting an app), the proposed model considers implicit semantics to enhance intent inference given noisy ASR inputs. We believe that this approach will lead to systems that are able to handle users' open domain intents by retrieving relevant apps that provide desired functionality either locally available or by suggesting installation of suitable apps and doing so in an unsupervised way. In sum, the effectiveness of the proposed feature-enriched model can be shown in different domains, indicating good generality and providing a reasonable direction for future work.

[2]Here we assume that the priors for apps/utterances, $P(a)$, are the same.

7. REFERENCES

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

[2] Stephanie Seneff, "TINA: A natural language system for spoken language applications," *Computational linguistics*, vol. 18, no. 1, pp. 61–86, 1992.

[3] 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 *Proc. of ACL*, 1993, pp. 54–61.

[4] Narendra Gupta, Gökhan Tür, Dilek Hakkani-Tür, Srinivas Bangalore, Giuseppe Riccardi, and Mazin Gilbert, "The AT&T spoken language understanding system," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 1, pp. 213–222, 2006.

[5] Dan Bohus and Alexander I Rudnicky, "The RavenClaw dialog management framework: Architecture and systems," *Computer Speech & Language*, vol. 23, no. 3, pp. 332–361, 2009.

[6] Sangkeun Jung, Cheongjae Lee, Kyungduk Kim, and Gary Geunbae Lee, "Hybrid approach to user intention modeling for dialog simulation," in *Proc. of ACL-IJCNLP*. Association for Computational Linguistics, 2009, pp. 17–20.

[7] David Griol and Zoraida Callejas, "A neural network approach to intention modeling for user-adapted conversational agents," *Computational Intelligence and Neuroscience*, vol. 2016, 2015.

[8] Larry Heck and Dilek Hakkani-Tür, "Exploiting the semantic web for unsupervised spoken language understanding," in *Proc. of SLT*, 2012, pp. 228–233.

[9] Larry P Heck, Dilek Hakkani-Tür, and Gokhan Tur, "Leveraging knowledge graphs for web-scale unsupervised semantic parsing.," in *Proc. of INTERSPEECH*, 2013, pp. 1594–1598.

[10] 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 *Proc. of ASRU*, 2013, pp. 120–125.

[11] 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 *Proc. of SLT*, 2014.

[12] 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 *Proc. of SLT*, 2014, pp. 584–589.

[13] Yun-Nung Chen, William Yang Wang, and Alexander I. Rudnicky, "Learning semantic hierarchy with distributional representations for unsupervised spoken language understanding," in *Proc. of INTERSPEECH*, 2015, pp. 1869–1873.

[14] Yun-Nung Chen, William Yang Wang, and Alexander I. Rudnicky, "Jointly modeling inter-slot relations by random walk on knowledge graphs for unsupervised spoken language understanding," in *Proc. of NAACL-HLT*. ACL, 2015.

[15] 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 *Proc. of INTERSPEECH*, 2012.

[16] 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.

[17] Ali El-Kahky, Derek Liu, Ruhi Sarikaya, Gökhan Tür, 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 *Proc. of ICASSP*, 2014.

[18] Yun-Nung Chen, Dilek Hakkani-Tür, and Xiaodong He, "Zero-shot learning of intent embeddings for expansion by convolutional deep structured semantic models," in *Proc. of ICASSP*, 2016.

[19] Anshuman Bhargava, Asli Celikyilmaz, Dilek Hakkani-Tur, and Ruhi Sarikaya, "Easy contextual intent prediction and slot detection," in *Proc. of ICASSP*, 2013.

[20] Anders Skrondal and Sophia Rabe-Hesketh, *Generalized latent variable modeling: Multilevel, longitudinal, and structural equation models*, Crc Press, 2004.

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

[22] Yun-Nung Chen, Ming Sun, Alexander I. Rudnicky, and Anatole Gershman, "Leveraging behavioral patterns of mobile applications for personalized spoken language understanding," in *Proc. of ICMI*, 2015.

[23] Yehuda Koren, Robert Bell, and Chris Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, , no. 8, pp. 30–37, 2009.

[24] Tomáš Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean, "Distributed representations of words and phrases and their compositionality," in *Proc. of NIPS*, 2013, pp. 3111–3119.

[25] L. Ratinov, D. Roth, D. Downey, and M. Anderson, "Local and global algorithms for disambiguation to wikipedia," in *Proc. of ACL*, 2011.

[26] X. Cheng and D. Roth, "Relational inference for wikification," in *Proc. of EMNLP*, 2013.

[27] Michael Collins, Sanjoy Dasgupta, and Robert E Schapire, "A generalization of principal components analysis to the exponential family," in *Proc. of NIPS*, 2001, pp. 617–624.

[28] Raghunandan H Keshavan, Andrea Montanari, and Sewoong Oh, "Matrix completion from a few entries," *IEEE Transactions on Information Theory*, vol. 56, no. 6, pp. 2980–2998, 2010.

[29] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme, "BPR: Bayesian personalized ranking from implicit feedback," in *Proc. of UAI*, 2009.

[30] Sebastian Riedel, Limin Yao, Andrew McCallum, and Benjamin M Marlin, "Relation extraction with matrix factorization and universal schemas," in *Proc. of NAACL-HLT*, 2013.

[31] Zeno Gantner, Steffen Rendle, Christoph Freudenthaler, and Lars Schmidt-Thieme, "MyMedialite: A free recommender system library," in *Proc. of RecSys*, 2011, pp. 305–308.

[32] Jay M Ponte and W Bruce Croft, "A language modeling approach to information retrieval," in *Proc. of SIGIR*, 1998.