Matrix Factorization with Knowledge Graph Propagation for Unsupervised Spoken Language Understanding

Yun-Nung Chen, William Yang Wang, Anatole Gershman, and Alexander I. Rudnicky
School of Computer Science, Carnegie Mellon University
5000 Forbes Aveue, Pittsburgh, PA 15213-3891, USA
{yvchen, yww, anatoleg, air}@cs.cmu.edu

Abstract

Spoken dialogue systems (SDS) typically require a predefined semantic ontology to train a spoken language understanding (SLU) module. In addition to the annotation cost, a key challenge for designing such an ontology is to define a coherent slot set while considering their complex relations. This paper introduces a novel matrix factorization (MF) approach to learn latent feature vectors for utterances and semantic elements without the need of corpus annotations. Specifically, our model learns the semantic slots for a domain-specific SDS in an unsupervised fashion, and carries out semantic parsing using latent MF techniques. To further consider the global semantic structure, such as inter-word and inter-slot relations, we augment the latent MF-based model with a knowledge graph propagation model based on a slot-based semantic graph and a word-based lexical graph. Our experiments show that the proposed MF approaches produce better SLU models that are able to predict semantic slots and word patterns taking into account their relations and domain-specificity in a joint manner.

1 Introduction

A key component of a spoken dialogue system (SDS) is the spoken language understanding (SLU) module—it parses the users' utterances into semantic representations; for example, the utterance "*find a cheap restaurant*" can be parsed into (price=*cheap*, target=*restaurant*) (Pieraccini et al., 1992). To design the SLU module of a SDS, most previous studies relied on predefined slots[1] for training the decoder (Seneff, 1992; Dowding et al., 1993; Gupta et al., 2006; Bohus and Rudnicky, 2009). However, these predefined semantic slots may bias the subsequent data collection process, and the cost of manually labeling utterances for updating the ontology is expensive (Wang et al., 2012).

In recent years, this problem led to the development of unsupervised SLU techniques (Heck and Hakkani-Tür, 2012; Heck et al., 2013; Chen et al., 2013b; Chen et al., 2014b). In particular, Chen et al. (2013b) proposed a frame-semantics based framework for automatically inducing semantic slots given raw audios. However, these approaches generally do not explicitly learn the latent factor representations to model the measurement errors (Skrondal and Rabe-Hesketh, 2004), nor do they jointly consider the complex lexical, syntactic, and semantic relations among words, slots, and utterances.

Another challenge of SLU is the inference of the hidden semantics. Considering the user utterance "*can i have a cheap restaurant*", from its surface patterns, we can see that it includes explicit semantic information about "price (cheap)" and "target (restaurant)"; however, it also includes hidden semantic information, such as "food" and "seeking", since the SDS needs to infer that the user wants to "find" some cheap "food", even though they are not directly observed in the surface patterns. Nonetheless, these implicit semantics are important semantic concepts for domain-specific SDSs. Traditional SLU models use discriminative classifiers (Henderson et al., 2012) to predict whether the predefined slots occur in the utterances or not, ignoring the unobserved concepts and the hidden semantic information.

In this paper, we take a rather radical approach: we propose a novel matrix factorization (MF) model for learning latent features for SLU, taking account of additional information such as the word relations, the induced slots, and the slot relations. To further consider the global coherence of induced slots, we combine the MF model with

[1] A slot is defined as a basic semantic unit in SLU, such as "price" and "target" in the example.

a knowledge graph propagation based model, fusing both a word-based lexical knowledge graph and a slot-based semantic graph. In fact, as it is shown in the Netflix challenge, MF is credited as the most useful technique for recommendation systems (Koren et al., 2009). Also, the MF model considers the unobserved patterns and estimates their probabilities instead of viewing them as negative examples. However, to the best of our knowledge, the MF technique is not yet well understood in the SLU and SDS communities, and it is not very straight-forward to use MF methods to learn latent feature representations for semantic parsing in SLU. To evaluate the performance of our model, we compare it to standard discriminative SLU baselines, and show that our MF-based model is able to produce strong results in semantic decoding, and the knowledge graph propagation model further improves the performance. Our contributions are three-fold:

- We are among the first to study matrix factorization techniques for unsupervised SLU, taking account of additional information;
- We augment the MF model with a knowledge graph propagation model, increasing the global coherence of semantic decoding using induced slots;
- Our experimental results show that the MF-based unsupervised SLU outperforms strong discriminative baselines, obtaining promising results.

In the next section, we outline the related work in unsupervised SLU and latent variable modeling for spoken language processing. Section 3 introduces our framework. The detailed MF approach is explained in Section 4. We then introduce the global knowledge graphs for MF in Section 5. Section 6 shows the experimental results, and Section 7 concludes.

2 Related Work

Unsupervised SLU Tur et al. (2011; 2012) were among the first to consider unsupervised approaches for SLU, where they exploited query logs for slot-filling. In a subsequent study, Heck and Hakkani-Tür (2012) studied the Semantic Web for an unsupervised intent detection problem in SLU, showing that results obtained from the unsupervised training process align well with the performance of traditional supervised learning. Following their success of unsupervised SLU, recent studies have also obtained interesting results on the tasks of relation detection (Hakkani-Tür et al., 2013; Chen et al., 2014a), entity extraction (Wang et al., 2014), and extending domain coverage (El-Kahky et al., 2014; Chen and Rudnicky, 2014). However, most of the studies above do not explicitly learn latent factor representations from the data—while we hypothesize that the better robustness in noisy data can be achieved by explicitly modeling the measurement errors (usually produced by automatic speech recognizers (ASR)) using latent variable models and taking additional local and global semantic constraints into account.

Latent Variable Modeling in SLU Early studies on latent variable modeling in speech included the classic hidden Markov model for statistical speech recognition (Jelinek, 1997). Recently, Celikyilmaz et al. (2011) were the first to study the intent detection problem using query logs and a discrete Bayesian latent variable model. In the field of dialogue modeling, the partially observable Markov decision process (POMDP) (Young et al., 2013) model is a popular technique for dialogue management, reducing the cost of handcrafted dialogue managers while producing robustness against speech recognition errors. More recently, Tur et al. (2013) used a semi-supervised LDA model to show improvement on the slot filling task. Also, Zhai and Williams (2014) proposed an unsupervised model for connecting words with latent states in HMMs using topic models, obtaining interesting qualitative and quantitative results. However, for unsupervised learning for SLU, it is not obvious how to incorporate additional information in the HMMs. To the best of our knowledge, this paper is the first to consider MF techniques for learning latent feature representations in unsupervised SLU, taking various local and global lexical, syntactic, and semantic information into account.

3 The Proposed Framework

This paper introduces a matrix factorization technique for unsupervised SLU,. The proposed framework is shown in Figure 1(a). Given the utterances, the task of the SLU model is to decode their surface patterns into semantic forms and differentiate the target semantic concepts from the generic semantic space for task-oriented SDSs simultaneously. Note that our model does not require any human-defined slots and domain-specific semantic representations for utterances.

In the proposed model, we first build a feature matrix to represent the training utterances, where each row represents an utterance, and each column refers to an observed surface pattern or a induced slot candidate. Figure 1(b) illustrates an example

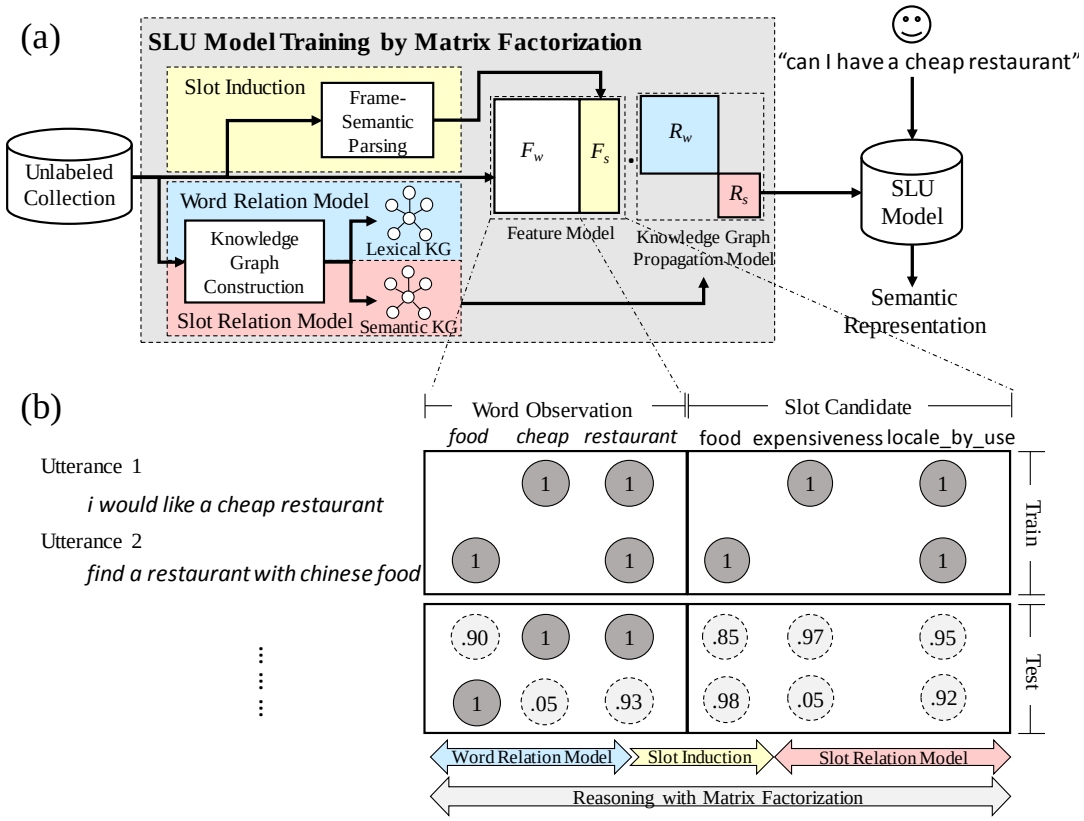

Figure 1: (a): The proposed framework. (b): Our matrix factorization method completes a partially-missing matrix for implicit semantic parsing. Dark circles are observed facts, shaded circles are inferred facts. The slot induction maps (yellow arrow) observed surface patterns to semantic slot candidates. The word relation model (blue arrow) constructs correlations between surface patterns. The slot relation model (pink arrow) learns the slot-level correlations based on propagating the automatically derived semantic knowledge graphs. Reasoning with matrix factorization (gray arrow) incorporates these models jointly, and produces a coherent, domain-specific SLU model.

of the matrix. Given a testing utterance, we convert it into a vector based on the observed surface patterns, and then fill in the missing values of the slots. In the first utterance in the figure, although the semantic slot food is not observed, the utterance implies the meaning facet food. The MF approach is able to learn the latent feature vectors for utterances and semantic elements, inferring implicit semantic concepts to improve the decoding process—namely, by filling the matrix with probabilities (lower part of the matrix).

The feature model is built on the observed word patterns and slot candidates, where the slot candidates are obtained from the slot induction component through frame-semantic parsing (the yellow block in Figure 1(a)) (Chen et al., 2013b). Section 4.1 explains the detail of the feature model.

In order to consider the additional inter-word and inter-slot relations, we propose a knowledge graph propagation model based on two knowledge graphs, which includes a word relation model (blue block) and a slot relation model (pink block), described in Section 4.2. The method of automatic knowledge graph construction is introduced in Section 5, where we leverage distributed word embeddings associated with typed syntactic dependencies to model the relations (Mikolov et al., 2013b; Mikolov et al., 2013c; Levy and Goldberg, 2014; Chen et al., 2015).

Finally, we train the SLU model by learning latent feature vectors for utterances and slot candidates through MF techniques. Combining with a knowledge graph propagation model based on word/slot relations, the trained SLU model estimates the probability that each semantic slot occurs in the testing utterance, and how likely each slot is domain-specific simultaneously. In other words, the SLU model is able to transform the testing utterances into domain-specific semantic representations without human involvement.

4 The Matrix Factorization Approach

Considering the benefits brought by MF techniques, including 1) modeling the noisy data, 2) modeling hidden semantics, and 3) modeling the

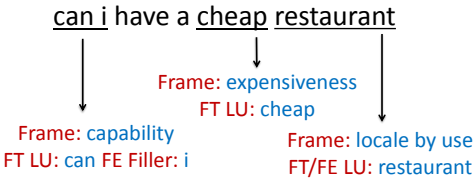

Figure 2: An example of probabilistic frame-semantic parsing on ASR output. FT: frame target. FE: frame element. LU: lexical unit.

long-range dependencies between observations, in this work we apply an MF approach to SLU modeling for SDSs. In our model, we use U to denote the set of input utterances, W as the set of word patterns, and S as the set of semantic slots that we would like to predict. The pair of an utterance $u \in U$ and a word pattern/semantic slot $x \in \{W \cup S\}$, $\langle u, x \rangle$, 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 $\{\langle u, x \rangle \in \mathcal{O}\}$. The goal of our model is to estimate, for a given utterance u and a given word pattern/semantic slot x, the probability, $p(M_{u,x} = 1)$, where $M_{u,x}$ is a binary random variable that is true if and only if x is the word pattern/domain-specific semantic slot in the utterance u. We introduce a series of exponential family models that estimate the probability using a natural parameter $\theta_{u,x}$ and the logistic sigmoid function:

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

We construct a matrix $M_{|U| \times (|W|+|S|)}$ as observed facts for MF by integrating a feature model and a knowledge graph propagation model below.

4.1 Feature Model

First, we build a word pattern matrix F_w with binary values based on observations, where each row represents an utterance and each column refers to an observed unigram. In other words, F_w carries the basic word vectors for the utterances, which is illustrated as the left part of the matrix in Figure 1(b).

To induce the semantic elements, we parse all ASR-decoded utterances in our corpus using SEMAFOR[2], a state-of-the-art semantic parser for frame-semantic parsing (Das et al., 2010; Das et al., 2013), and extract all frames from semantic parsing results as slot candidates (Chen et al., 2013b; Dinarelli et al., 2009). Figure 2 shows an example of an ASR-decoded output parsed by SEMAFOR. Three FrameNet-defined frames (capability, expensiveness, and locale_by_use) are generated for the utterance, which we consider as slot candidates for a domain-specific dialogue system (Baker et al., 1998). Then we build a slot matrix F_s with binary values based on the induced slots, which also denotes the slot features for the utterances (right part of the matrix in Figure 1(b)).

To build the feature model M_F, we concatenate two matrices:

$$M_F = [\ F_w \quad F_s\ ], \quad (2)$$

which is the upper part of the matrix in Figure 1(b) for training utterances. Note that we do not use any annotations, so all slot candidates are included.

4.2 Knowledge Graph Propagation Model

Since SEMAFOR was trained on FrameNet annotation, which has a more generic frame-semantic context, not all the frames from the parsing results can be used as the actual slots in the domain-specific dialogue systems. For instance, in Figure 2, we see that the frames "expensiveness" and "locale_by_use" are essentially the key slots for the purpose of understanding in the restaurant query domain, whereas the "capability" frame does not convey particularly valuable information for SLU.

Assuming that domain-specific concepts are usually related to each other, considering global relations between semantic slots induces a more coherent slot set. It is shown that the relations on knowledge graphs help make decisions on domain-specific slots (Chen et al., 2015). Considering two directed graphs, semantic and lexical knowledge graphs, each node in the semantic knowledge graph is a slot candidate s_i generated by the frame-semantic parser, and each node in the lexical knowledge graph is a word w_j.

- **Slot-based semantic knowledge graph** is built as $G_s = \langle V_s, E_{ss} \rangle$, where $V_s = \{s_i \in S\}$ and $E_{ss} = \{e_{ij} \mid s_i, s_j \in V_s\}$.
- **Word-based lexical knowledge graph** is built as $G_w = \langle V_w, E_{ww} \rangle$, where $V_w = \{w_i \in W\}$ and $E_{ww} = \{e_{ij} \mid w_i, w_j \in V_w\}$.

The edges connect two nodes in the graphs if there is a typed dependency between them. Figure 3 is a simplified example of a slot-based semantic knowledge graph. The structured graph helps define a coherent slot set. To model the relations between words/slots based on the knowledge graphs, we define two relation models below.

[2] http://www.ark.cs.cmu.edu/SEMAFOR/

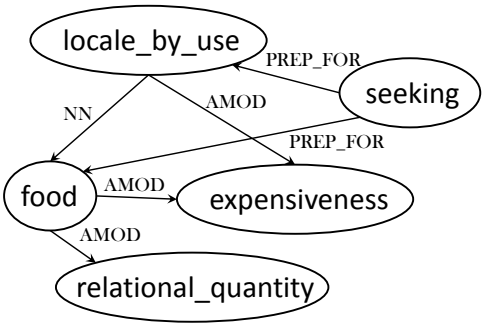

Figure 3: A simplified example of the automatically derived knowledge graph.

- Semantic Relation
  For modeling word semantic relations, we compute a matrix $R_w^S = [Sim(w_i, w_j)]_{|W|\times|W|}$, where $Sim(w_i, w_j)$ is the cosine similarity between the dependency embeddings of the word patterns w_i and w_j after normalization. For slot semantic relations, we compute $R_s^S = [Sim(s_i, s_j)]_{|S|\times|S|}$ similarly[3]. The matrices R_w^S and R_s^S model not only the semantic but functional similarity since we use dependency-based embeddings (Levy and Goldberg, 2014).

- Dependency Relation
  Assuming that important semantic slots are usually mutually related to each other, that is, connected by syntactic dependencies, our automatically derived knowledge graphs are able to help model the dependency relations. For word dependency relations, we compute a matrix $R_w^D = [\hat{r}(w_i, w_j)]_{|W|\times|W|}$, where $\hat{r}(w_i, w_j)$ measures the dependency between two word patterns w_i and w_j based on the word-based lexical knowledge graph, and the detail is described in Section 5. For slot dependency relations, we similarly compute $R_s^D = [\hat{r}(s_i, s_j)]_{|S|\times|S|}$ based on the slot-based semantic knowledge graph.

With the built word relation models (R_w^S and R_w^D) and slot relation models (R_s^S and R_s^D), we combine them as a knowledge graph propagation matrix M_R[4].

$$M_R = \begin{bmatrix} R_w^{SD} & 0 \\ 0 & R_s^{SD} \end{bmatrix}, \quad (3)$$

where $R_w^{SD} = R_w^S + R_w^D$ and $R_s^{SD} = R_s^S + R_s^D$ to integrate semantic and dependency relations. The goal of this matrix is to propagate scores between nodes according to different types of relations in the knowledge graphs (Chen and Metze, 2012).

4.3 Integrated Model

With a feature model M_F and a knowledge graph propagation model M_R, we integrate them into a single matrix.

$$\begin{aligned} M &= M_F \cdot (\alpha I + \beta M_R) \quad (4) \\ &= \begin{bmatrix} \alpha F_w + \beta F_w R_w & 0 \\ 0 & \alpha F_s + \beta F_s R_s \end{bmatrix}, \end{aligned}$$

where M is the final matrix and I is the identity matrix. α and β are the weights for balancing original values and propagated values, where $\alpha + \beta = 1$. The matrix M is similar to M_F, but some weights are enhanced through the knowledge graph propagation model, M_R. The word relations are built by $F_w R_w$, which is the matrix with internal weight propagation on the lexical knowledge graph (the blue arrow in Figure 1(b)). Similarly, $F_s R_s$ models the slot correlations, and can be treated as the matrix with internal weight propagation on the semantic knowledge graph (the pink arrow in Figure 1(b)). The propagation models can be treated as running a random walk algorithm on the graphs.

F_s contains all slot candidates generated by SEMAFOR, which may include some generic slots (such as capability), so the original feature model cannot differentiate the domain-specific and generic concepts. By integrating with R_s, the semantic and dependency relations can be propagated via the knowledge graph, and the domain-specific concepts may have higher weights based on the assumption that the slots for dialogue systems are often mutually related (Chen et al., 2015). Hence, the structure information can be automatically involved in the matrix. Also, the word relation model brings the same function, but now on the word level. In conclusion, for each utterance, the integrated model not only predicts the probability that semantic slots occur but also considers whether the slots are domain-specific. The following sections describe the learning process.

4.4 Parameter Estimation

The proposed model is parameterized through weights and latent component vectors, where the parameters are estimated by maximizing the log

[3]For each column in R_w^S and R_s^S, we only keep top 10 highest values, which correspond the top 10 semantically similar nodes.

[4]The values in the diagonal of M_R are 0 to model the propagation from other entries.

likelihood of observed data (Collins et al., 2001).

$$\begin{aligned}\theta^* &= \arg\max_{\theta} \prod_{u \in U} p(\theta \mid M_u) \qquad (5)\\ &= \arg\max_{\theta} \prod_{u \in U} p(M_u \mid \theta) p(\theta)\\ &= \arg\max_{\theta} \sum_{u \in U} \ln p(M_u \mid \theta) - \lambda_\theta,\end{aligned}$$

where M_u is the vector corresponding to the utterance u from $M_{u,x}$ 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 an 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 (Rendle et al., 2009). Riedel et al. (2013) also showed that BPR learns the implicit relations for improving the relation extraction task.

4.4.1 Objective Function

To estimate the parameters in (5), we create a dataset of *ranked pairs* from M in (4): for each utterance u and each observed fact $f^+ = \langle u, x^+ \rangle$, where $M_{u,x} \geq \delta$, we choose each word pattern/slot x^- such that $f^- = \langle u, x^- \rangle$, where $M_{u,x} < \delta$, which refers to the word pattern/slot we have not observed to be in utterance u. That is, we construct the observed data $\mathcal{O}$ from M. Then for each pair of facts f^+ and f^-, we want to model $p(f^+) > p(f^-)$ and hence $\theta_{f^+} > \theta_{f^-}$ according to (1). BPR maximizes the summation of each ranked pair, where the objective is

$$\sum_{u \in U} \ln p(M_u \mid \theta) = \sum_{f^+ \in \mathcal{O}} \sum_{f^- \notin \mathcal{O}} \ln \sigma(\theta_{f^+} - \theta_{f^-}). \quad (6)$$

The BPR objective is an approximation to the per utterance AUC (area under the ROC curve), which directly correlates to what we want to achieve – well-ranked semantic slots per utterance.

4.4.2 Optimization

To maximize the objective in (6), we employ a stochastic gradient descent (SGD) algorithm (Rendle et al., 2009). For each randomly sampled observed fact $\langle u, x^+ \rangle$, we sample an unobserved fact $\langle u, x^- \rangle$, 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 matrix factorization (Gantner et al., 2011).

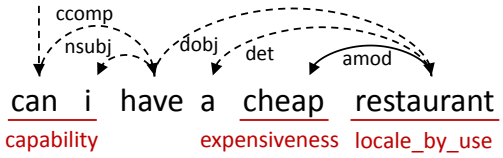

Figure 4: The dependency parsing result.

5 Knowledge Graph Construction

This section introduces the procedure of constructing knowledge graphs in order to estimate $\hat{r}(w_i, w_j)$ for building R_w^D and $\hat{r}(s_i, s_j)$ for R_s^D in Section 4.2. Considering the relations in the knowledge graphs, the edge weights for E_{ww} and E_{ss} are measured as $\hat{r}(w_i, w_j)$ and $\hat{r}(s_i, s_j)$ based on the dependency parsing results respectively.

The example utterance "*can i have a cheap restaurant*" and its dependency parsing result are illustrated in Figure 4. The arrows denote the dependency relations from headwords to their dependents, and words on arcs denote types of the dependencies. All typed dependencies between two words are encoded in triples and form a word-based dependency set $\mathcal{T}_w = \{\langle w_i, t, w_j \rangle\}$, where t is the typed dependency between the headword w_i and the dependent w_j. For example, Figure 4 generates $\langle$*restaurant*, AMOD, *cheap*$\rangle$, $\langle$*restaurant*, DOBJ, *have*$\rangle$, etc. for $\mathcal{T}_w$. Similarly, we build a slot-based dependency set $\mathcal{T}_s = \{\langle s_i, t, s_j \rangle\}$ by transforming dependencies between slot-fillers into ones between slots. For example, $\langle$*restaurant*, AMOD, *cheap*$\rangle$ from $\mathcal{T}_w$ is transformed into $\langle$locale_by_use, AMOD, expensiveness$\rangle$ for building $\mathcal{T}_s$, because both sides of the non-dotted line are parsed as slot-fillers by SEMAFOR.

5.1 Relation Weight Estimation

For the edges in the knowledge graphs, we model the relations between two connected nodes x_i and x_j as $\hat{r}(x_i, x_j)$, where x is either a slot s or a word pattern w. Since the weights are measured based on the relations between nodes regardless of the directions, we combine the scores of two directional dependencies:

$$\hat{r}(x_i, x_j) = r(x_i \rightarrow x_j) + r(x_j \rightarrow x_i), \qquad (7)$$

where $r(x_i \rightarrow x_j)$ is the score estimating the dependency including x_i as a head and x_j as a dependent. We propose a scoring function for $r(\cdot)$ using dependency-based embeddings.

Table 1: The example contexts extracted for training dependency-based word/slot embeddings.

	Typed Dependency Relation	Target Word	Contexts
Word	$\langle$*restaurant*, AMOD, *cheap*$\rangle$	*restaurant* *cheap*	*cheap*/AMOD *restaurant*/AMOD^{-1}
Slot	$\langle$locale_by_use, AMOD, expensiveness$\rangle$	locale_by_use expansiveness	expensiveness/AMOD locale_by_use/AMOD^{-1}

5.1.1 Dependency-Based Embeddings

Most neural embeddings use linear bag-of-words contexts, where a window size is defined to produce contexts of the target words (Mikolov et al., 2013c; Mikolov et al., 2013b; Mikolov et al., 2013a). However, some important contexts may be missing due to smaller windows, while larger windows capture broad topical content. A dependency-based embedding approach was proposed to derive contexts based on the syntactic relations the word participates in for training embeddings, where the embeddings are less topical but offer more functional similarity compared to original embeddings (Levy and Goldberg, 2014).

Table 1 shows the extracted dependency-based contexts for each target word from the example in Figure 4, where headwords and their dependents can form the contexts by following the arc on a word in the dependency tree, and -1 denotes the directionality of the dependency. After replacing original bag-of-words contexts with dependency-based contexts, we can train dependency-based embeddings for all target words (Yih et al., 2014; Bordes et al., 2011; Bordes et al., 2013).

For training dependency-based word embeddings, each target x is associated with a vector $\mathbf{v_x} \in \mathbb{R}^d$ and each context c is represented as a context vector $\mathbf{v_c} \in \mathbb{R}^d$, where d is the embedding dimensionality. We learn vector representations for both targets and contexts such that the dot product $\mathbf{v_x} \cdot \mathbf{v_c}$ associated with "good" target-context pairs belonging to the training data $\mathcal{D}$ is maximized, leading to the objective function:

$$\arg\max_{\mathbf{v_x},\mathbf{v_c}} \sum_{(w,c)\in\mathcal{D}} \log \frac{1}{1+\exp(-\mathbf{v_c}\cdot\mathbf{v_x})}, \quad (8)$$

which can be trained using stochastic-gradient updates (Levy and Goldberg, 2014). Then we can obtain the dependency-based slot and word embeddings using $\mathcal{T}_s$ and $\mathcal{T}_w$ respectively.

5.1.2 Embedding-Based Scoring Function

With trained dependency-based embeddings, we estimate the probability that x_i is the headword and x_j is its dependent via the typed dependency t as

$$P(x_i \underset{t}{\rightarrow} x_j) = \frac{Sim(x_i, x_j/t) + Sim(x_j, x_i/t^{-1})}{2}, \quad (9)$$

where $Sim(x_i, x_j/t)$ is the cosine similarity between word/slot embeddings $\mathbf{v_{x_i}}$ and context embeddings $\mathbf{v_{x_j/t}}$ after normalizing to $[0, 1]$.

Based on the dependency set $\mathcal{T}_x$, we use $t^*_{x_i \to x_j}$ to denote the most possible typed dependency with x_i as a head and x_j as a dependent.

$$t^*_{x_i\to x_j} = \arg\max_t C(x_i \underset{t}{\rightarrow} x_j), \quad (10)$$

where $C(x_i \underset{t}{\rightarrow} x_j)$ counts how many times the dependency $\langle x_i, t, x_j\rangle$ occurs in the dependency set $\mathcal{T}_x$. Then the scoring function $r(\cdot)$ in (7) that estimates the dependency $x_i \to x_j$ is measured as

$$r(x_i \to x_j) = C(x_i \xrightarrow[t^*_{x_i\to x_j}]{} x_j) \cdot P(x_i \xrightarrow[t^*_{x_i\to x_j}]{} x_j), \quad (11)$$

which is equal to the highest observed frequency of the dependency $x_i \to x_j$ among all types from $\mathcal{T}_x$ and additionally weighted by the estimated probability. The estimated probability smoothes the observed frequency to avoid overfitting due to the smaller dataset. Figure 3 is a simplified example of an automatically derived semantic knowledge graph with the most possible typed dependencies as edges based on the estimated weights. Then the relation weights $\hat{r}(x_i, x_j)$ can be obtained by (7) in order to build R^D_w and R^D_s matrices.

6 Experiments

6.1 Experimental Setup

In this experiment, we used the Cambridge University SLU corpus, previously used on several other SLU tasks (Henderson et al., 2012; Chen et al., 2013a). The domain of the corpus is about restaurant recommendation in Cambridge; subjects were asked to interact with multiple SDSs in an in-car setting. The corpus contains a total number of 2,166 dialogues, including 15,453 utterances (10,571 for self-training and 4,882 for

Table 2: The MAP of predicted slots (%); † indicates that the result is significantly better than the Logistic Regression (row (b)) with $p < 0.05$ in t-test.

Approach				ASR		Manual	
				w/o	w/ Explicit	w/o	w/ Explicit
Explicit	SVM		(a)	32.48		36.62	
	MLR		(b)	33.96		38.78	
Implicit	Baseline	Random	(c)	3.43	22.45	2.63	25.09
		Majority	(d)	15.37	32.88	16.43	38.41
	MF	Feature	(e)	24.24	37.61†	22.55	45.34†
		Feature + KGP	(f)	**40.46**†	**43.51**†	**52.14**†	**53.40**†

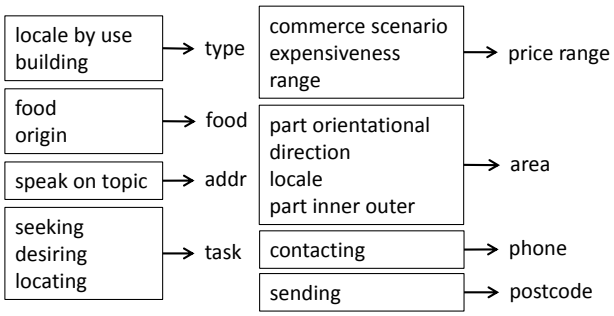

Figure 5: The mappings from induced slots (within blocks) to reference slots (right sides of arrows).

testing). The data is gender-balanced, with slightly more native than non-native speakers. The vocabulary size is 1868. An ASR system was used to transcribe the speech; the word error rate was reported as 37%. There are 10 slots created by domain experts: addr, area, food, name, phone, postcode, price range, signature, task, and type.

For parameter setting, the weights for balancing feature models and propagation models, α and β, are set as 0.5 to give the same influence, and the threshold for defining the unobserved facts δ is set as 0.5 for all experiments. We use the Stanford Parser[5] to obtain the collapsed typed syntactic dependencies (Socher et al., 2013) and set the dimensionality of embeddings $d = 300$ in all experiments.

6.2 Evaluation Metrics

To evaluate the accuracy of the automatically decoded slots, we measure their quality as the proximity between predicted slots and reference slots. Figure 5 shows the mappings that indicate semantically related induced slots and reference slots (Chen et al., 2013b).

To eliminate the influence of threshold selection when predicting semantic slots, in the following metrics, we take the whole ranking list into account and evaluate the performance by the metrics that are independent of the selected threshold. For each utterance, with the predicted probabilities of all slot candidates, we can compute an average precision (AP) to evaluate the performance of SLU by treating the slots with mappings as positive. AP scores the ranking result higher if the correct slots are ranked higher, which also approximates to the area under the precision-recall curve (Boyd et al., 2012). Mean average precision (MAP) is the metric for evaluating all utterances. For all experiments, we perform a paired t-test on the AP scores of the results to test the significance.

6.3 Evaluation Results

Table 2 shows the MAP performance of predicted slots for all experiments on ASR and manual transcripts. For the first baseline using explicit semantics, we use the observed data to self-train models for predicting the probability of each semantic slot by support vector machine (SVM) with a linear kernel and multinomial logistic regression (MLR) (row (a)-(b)) (Pedregosa et al., 2011; Henderson et al., 2012). It is shown that SVM and MLR perform similarly, and MLR is slightly better than SVM because it has better capability of estimating probabilities. For modeling implicit semantics, two baselines are performed as references, Random (row (c)) and Majority (row (d)), where the former assigns random probabilities for all slots, and the later assigns probabilities for the slots based on their frequency distribution. To improve probability estimation, we further integrate the results from implicit semantics with the better result from explicit approaches, MLR (row (b)), by averaging the probability distribution from two results.

Two baselines, Random and Majority, cannot model the implicit semantics, producing poor results. The results of Random integrated with MLR significantly degrades the performance of

[5] http://nlp.stanford.edu/software/lex-parser.shtml

Table 3: The MAP of predicted slots using different types of relation models in M_R (%); † indicates that the result is significantly better than the feature model (column (a)) with $p < 0.05$ in t-test.

Model	Feature	Knowledge Graph Propagation Model				
Rel.	(a) None	(b) Semantic	(c) Dependency	(d) Word	(e) Slot	(f) All
M_R	-	$\begin{bmatrix} R_w^S & 0 \\ 0 & R_s^S \end{bmatrix}$	$\begin{bmatrix} R_w^D & 0 \\ 0 & R_s^D \end{bmatrix}$	$\begin{bmatrix} R_w^{SD} & 0 \\ 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 \\ 0 & R_s^{SD} \end{bmatrix}$	$\begin{bmatrix} R_w^{SD} & 0 \\ 0 & R_s^{SD} \end{bmatrix}$
ASR	37.61	41.39†	41.63†	39.19†	42.10†	**43.51**†
Manual	45.34	51.55†	49.04†	45.18	49.91†	**53.40**†

MLR for both ASR and manual transcripts. Also, the results of Majority integrated with MLR does not produce any difference compared to the MLR baseline. Among the proposed MF approaches, only using feature model for building the matrix (row (e)) achieves 24.2% and 22.6% of MAP for ASR and manual results respectively, which are worse than two baselines using explicit semantics. However, with the combination of explicit semantics, using only the feature model significantly outperforms the baselines, where the performance comes from about 34.0% to 37.6% and from 38.8% to 45.3% for ASR and manual results respectively. Additionally integrating a knowledge graph propagation (KGP) model (row (f)) outperforms the baselines for both ASR and manual transcripts, and the performance is further improved by combining with explicit semantics (achieving MAP of 43.5% and 53.4%). The experiments show that the proposed MF models successfully learn the implicit semantics and consider the relations and domain-specificity simultaneously.

6.4 Discussion and Analysis

With promising results obtained by the proposed models, we analyze the detailed difference between different relation models in Table 3.

6.4.1 Effectiveness of Semantic and Dependency Relation Models

To evaluate the effectiveness of semantic and dependency relations, we consider each of them individually in M_R of (3) (columns (b) and (c) in Table 3). Comparing to the original model (column (a)), both modeling semantic relations and modeling dependency relations significantly improve the performance for ASR and manual results. It is shown that semantic relations help the SLU model infer the implicit meaning, and then the prediction becomes more accurate. Also, dependency relations successfully differentiate the generic concepts from the domain-specific concepts, so that the SLU model is able to predict more coherent set of semantic slots (Chen et al., 2015). Integrating two types of relations (column (f)) further improves the performance.

6.4.2 Comparing Word/ Slot Relation Models

To analyze the performance results from inter-word and inter-slot relations, the columns (d) and (e) show the results considering only word relations and only slot relations respectively. It can be seen that the inter-slot relation model significantly improves the performance for both ASR and manual results. However, the inter-word relation model only performs slightly better results for ASR output (from 37.6% to 39.2%), and there is no difference after applying the inter-word relation model on manual transcripts. The reason may be that inter-slot relations carry high-level semantics that align well with the structure of SDSs, but inter-word relations do not. Nevertheless, combining two relations (column (f)) outperforms both results for ASR and manual transcripts, showing that different types of relations can compensate each other and then benefit the SLU performance.

7 Conclusions

This paper presents an MF approach to self-train the SLU model for semantic decoding in an unsupervised way. The purpose of the proposed model is not only to predict the probability of each semantic slot but also to distinguish between generic semantic concepts and domain-specific concepts that are related to an SDS. The experiments show that the MF-based model obtains promising results, outperforming strong discriminative baselines.

Acknowledgments

We thank anonymous reviewers for their useful comments and Prof. Manfred Stede for his mentoring. We are also grateful to MetLife's support. Any opinions, findings, and conclusions expressed in this publication are those of the authors and do not necessarily reflect the views of funding agencies.

References

Collin F Baker, Charles J Fillmore, and John B Lowe. 1998. The Berkeley FrameNet project. In *Proceedings of COLING*, pages 86–90.

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

Antoine Bordes, Jason Weston, Ronan Collobert, Yoshua Bengio, et al. 2011. Learning structured embeddings of knowledge bases. In *Proceedings of AAAI*.

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *Proceedings of Advances in Neural Information Processing Systems*, pages 2787–2795.

Kendrick Boyd, Vitor Santos Costa, Jesse Davis, and C David Page. 2012. Unachievable region in precision-recall space and its effect on empirical evaluation. In *Machine learning: proceedings of the International Conference. International Conference on Machine Learning*, volume 2012, page 349. NIH Public Access.

Asli Celikyilmaz, Dilek Hakkani-Tür, and Gokhan Tür. 2011. Leveraging web query logs to learn user intent via bayesian discrete latent variable model. In *Proceedings of ICML*.

Yun-Nung Chen and Florian Metze. 2012. Two-layer mutually reinforced random walk for improved multi-party meeting summarization. In *Proceedings of The 4th IEEE Workshop on Spoken Language Tachnology*, pages 461–466.

Yun-Nung Chen and Alexander I. Rudnicky. 2014. Dynamically supporting unexplored domains in conversational interactions by enriching semantics with neural word embeddings. In *Proceedings of 2014 IEEE Spoken Language Technology Workshop (SLT)*, pages 590–595. IEEE.

Yun-Nung Chen, William Yang Wang, and Alexander I. Rudnicky. 2013a. An empirical investigation of sparse log-linear models for improved dialogue act classification. In *Proceedings of ICASSP*, pages 8317–8321.

Yun-Nung Chen, William Yang Wang, and Alexander I Rudnicky. 2013b. Unsupervised induction and filling of semantic slots for spoken dialogue systems using frame-semantic parsing. In *Proceedings of 2013 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pages 120–125. IEEE.

Yun-Nung Chen, Dilek Hakkani-Tür, and Gokan Tur. 2014a. Deriving local relational surface forms from dependency-based entity embeddings for unsupervised spoken language understanding. In *Proceedings of 2014 IEEE Spoken Language Technology Workshop (SLT)*, pages 242–247. IEEE.

Yun-Nung Chen, William Yang Wang, and Alexander I. Rudnicky. 2014b. Leveraging frame semantics and distributional semantics for unsupervised semantic slot induction in spoken dialogue systems. In *Proceedings of 2014 IEEE Spoken Language Technology Workshop (SLT)*, pages 584–589. IEEE.

Yun-Nung Chen, William Yang Wang, and Alexander I. Rudnicky. 2015. Jointly modeling inter-slot relations by random walk on knowledge graphs for unsupervised spoken language understanding. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics - Human Language Technologies*. ACL.

Michael Collins, Sanjoy Dasgupta, and Robert E Schapire. 2001. A generalization of principal components analysis to the exponential family. In *Proceedings of Advances in Neural Information Processing Systems*, pages 617–624.

Dipanjan Das, Nathan Schneider, Desai Chen, and Noah A Smith. 2010. Probabilistic frame-semantic parsing. In *Proceedings of The Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 948–956.

Dipanjan Das, Desai Chen, André F. T. Martins, Nathan Schneider, and Noah A. Smith. 2013. Frame-semantic parsing. *Computational Linguistics*.

Marco Dinarelli, Silvia Quarteroni, Sara Tonelli, Alessandro Moschitti, and Giuseppe Riccardi. 2009. Annotating spoken dialogs: from speech segments to dialog acts and frame semantics. In *Proceedings of the 2nd Workshop on Semantic Representation of Spoken Language*, pages 34–41. ACL.

John Dowding, Jean Mark Gawron, Doug Appelt, John Bear, Lynn Cherny, Robert Moore, and Douglas Moran. 1993. Gemini: A natural language system for spoken-language understanding. In *Proceedings of ACL*, pages 54–61.

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

Zeno Gantner, Steffen Rendle, Christoph Freudenthaler, and Lars Schmidt-Thieme. 2011. Mymedialite: A free recommender system library. In *Proceedings of the fifth ACM conference on Recommender systems*, pages 305–308. ACM.

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

Dilek Hakkani-Tür, Larry Heck, and Gokhan Tur. 2013. Using a knowledge graph and query click logs for unsupervised learning of relation detection. In *Proceedings of ICASSP*, pages 8327–8331.

Larry Heck and Dilek Hakkani-Tür. 2012. Exploiting the semantic web for unsupervised spoken language understanding. In *Proceedings of SLT*, pages 228–233.

Larry P Heck, Dilek Hakkani-Tür, and Gokhan Tur. 2013. Leveraging knowledge graphs for web-scale unsupervised semantic parsing. In *Proceedings of INTERSPEECH*, pages 1594–1598.

Matthew Henderson, Milica Gasic, Blaise Thomson, Pirros Tsiakoulis, Kai Yu, and Steve Young. 2012. Discriminative spoken language understanding using word confusion networks. In *Proceedings of SLT*, pages 176–181.

Frederick Jelinek. 1997. *Statistical methods for speech recognition*. MIT press.

Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. *Computer*, (8):30–37.

Omer Levy and Yoav Goldberg. 2014. Dependency-based word embeddings. In *Proceedings of ACL*.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. In *Proceedings of Workshop at ICLR*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *Proceedings of Advances in Neural Information Processing Systems*, pages 3111–3119.

Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013c. Linguistic regularities in continuous space word representations. In *HLT-NAACL*, pages 746–751. Citeseer.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine learning in python. *The Journal of Machine Learning Research*, 12:2825–2830.

Roberto Pieraccini, Evelyne Tzoukermann, Zakhar Gorelov, J Gauvain, Esther Levin, Chin-Hui Lee, and Jay G Wilpon. 1992. A speech understanding system based on statistical representation of semantics. In *Proceedings of 1992 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 1, pages 193–196. IEEE.

Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian personalized ranking from implicit feedback. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, pages 452–461. AUAI Press.

Sebastian Riedel, Limin Yao, Andrew McCallum, and Benjamin M Marlin. 2013. Relation extraction with matrix factorization and universal schemas. In *Proceedings of NAACL-HLT*, pages 74–84.

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

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

Richard Socher, John Bauer, Christopher D Manning, and Andrew Y Ng. 2013. Parsing with compositional vector grammars. In *Proceedings of the ACL conference*. Citeseer.

Gokhan Tur, Dilek Z Hakkani-Tür, Dustin Hillard, and Asli Celikyilmaz. 2011. Towards unsupervised spoken language understanding: Exploiting query click logs for slot filling. In *Proceedings of INTERSPEECH*, pages 1293–1296.

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

Gokhan Tur, Asli Celikyilmaz, and Dilek Hakkani-Tur. 2013. Latent semantic modeling for slot filling in conversational understanding. In *Proceedings of 2013 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8307–8311. IEEE.

William Yang Wang, Dan Bohus, Ece Kamar, and Eric Horvitz. 2012. Crowdsourcing the acquisition of natural language corpora: Methods and observations. In *Proceedings of SLT*, pages 73–78.

Lu Wang, Dilek Hakkani-Tür, and Larry Heck. 2014. Leveraging semantic web search and browse sessions for multi-turn spoken dialog systems. In *Proceedings of 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4082–4086. IEEE.

Wen-tau Yih, Xiaodong He, and Christopher Meek. 2014. Semantic parsing for single-relation question answering. In *Proceedings of ACL*.

Steve Young, Milica Gasic, Blaise Thomson, and Jason D Williams. 2013. POMDP-based statistical spoken dialog systems: A review. *Proceedings of the IEEE*, 101(5):1160–1179.

Ke Zhai and Jason D Williams. 2014. Discovering latent structure in task-oriented dialogues. In *Proceedings of the Association for Computational Linguistics*.