Jointly Modeling Inter-Slot Relations by Random Walk on Knowledge Graphs for Unsupervised Spoken Language Understanding

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

Abstract

A key challenge of designing coherent semantic ontology for spoken language understanding is to consider inter-slot relations. In practice, however, it is difficult for domain experts and professional annotators to define a coherent slot set, while considering various lexical, syntactic, and semantic dependencies. In this paper, we exploit the typed syntactic dependency theory for unsupervised induction and filling of semantics slots in spoken dialogue systems. More specifically, we build two knowledge graphs: a slot-based semantic graph, and a word-based lexical graph. To jointly consider word-to-word, word-to-slot, and slot-to-slot relations, we use a random walk inference algorithm to combine the two knowledge graphs, guided by dependency grammars. The experiments show that considering inter-slot relations is crucial for generating a more coherent and compete slot set, resulting in a better spoken language understanding model, while enhancing the interpretability of semantic slots.

1 Introduction

An important requirement for building a successful spoken dialogue system (SDS) is to define a coherent slot set and the corresponding slot-fillers for the spoken language understanding (SLU) component. Unfortunately, since the semantic slots are often mutually-related, it is non-trivial for domain experts and professional annotators to design a such slot set for semantic representation of SLU.

Considering a restaurant domain (Henderson et al., 2012), "*restaurant*" is the target slot, and important adjective modifiers such as "*Asian*" (the restaurant type) and "*cheap*" (the price of the restaurant) should be included in the slot set, so that the semantic representation of SLU can be more coherent and complete. In this case, it is challenging to design such a coherent and complete slot set manually, while considering various lexical, syntactic, and semantic dependencies.

Instead of considering slots independently, this paper takes a data-driven approach to model word-to-word relations via syntactic dependencies and further infer slot-to-slot relations. To do this, we incorporate the typed dependency grammar theory (De Marneffe and Manning, 2008) in a state-of-the-art frame-semantic driven unsupervised slot induction framework (Chen et al., 2013b). In particular, we build two knowledge graphs: a slot-based semantic knowledge graph, and a word-based lexical knowledge graph. Using typed dependency triples, we then study the stochastic relations between slots and words, using a mutually-reinforced random walk inference procedure to combine the two knowledge graphs. In evaluations, we use the jointly learned inter-slot relations to induce a coherent slot set in an unsupervised fashion. Our contributions are three-fold:

- We are among the first to consider unsupervised spoken language understanding combining semantic and lexical knowledge graphs;
- We propose a novel typed syntactic dependency grammar driven random walk model for relation discovery;

Human Language Technologies: The 2015 Annual Conference of the North American Chapter of the ACL, pages 619–629,
Denver, Colorado, May 31 – June 5, 2015.

- Our experimental results suggest that jointly considering inter-slot relations helps obtain a more coherent and complete semantic slot set.

2 Related Work

With the recent success of commercial dialogue systems and personal assistants (e.g., Microsoft's Cortana[1], Google Now[2], Apple's Siri[3], and Amazon's Echo[4]), a key focus on developing spoken understanding techniques is the scalability issue.

From the knowledge management perspective, empowering the system with a large knowledge base is of crucial significance to modern spoken dialogue systems. On this end, our work clearly aligns with recent studies on leveraging semantic knowledge graphs for SLU modeling (Heck et al., 2013; Hakkani-Tür et al., 2013; Hakkani-Tür et al., 2014; El-Kahky et al., 2014; Chen et al., 2014a). While leveraging external knowledge is the trend, efficient inference algorithms, such as random walk, are still less-studied for direct inference on knowledge graphs of the spoken contents.

In the natural language processing literature, Lao et al. (2011) used a random walk algorithm to construct inference rules on large entity-based knowledge bases, and leveraged syntactic information for reading the Web (Lao et al., 2012). Even though this work has important contributions, the proposed algorithm cannot learn mutually-recursive relations, and does not to consider lexical items—in fact, more and more studies show that, in addition to semantic knowledge graphs, lexical knowledge graphs (Inkpen and Hirst, 2006; Song et al., 2011; Li et al., 2013b) that model surface-level natural language realization, multiword expressions, and context (Li et al., 2013a), are also critical for short text understanding (Song et al., 2011; Wang et al., 2014).

From the engineering perspective, quick and easy development turnaround time for domain-specific dialogue applications is also critical (Chen and Rudnicky, 2014). Prior work shows that it is possible to use the frame-semantics theory to automatically induce and fill semantic slots (Chen et al., 2013b), and that leveraging distributional semantics helps improving the performance (Chen et al., 2014b). However, prior works treat each slot independently and have not considered the inter-slot relations when inducing the semantic slots. To the best of our knowledge, we are the first to use syntactically-informed random walk algorithms to combine the semantic and lexical knowledge graphs, and not individually but globally inducing the semantic slots for building better unsupervised SLU components.

[1] http://www.windowsphone.com/en-us/how-to/wp8/cortana

[2] http://www.google.com/landing/now

[3] http://www.apple.com/ios/siri

[4] http://www.amazon.com/oc/echo

3 The Proposed Framework

We build our approach on top of the recent success of an unsupervised frame-semantic parsing approach (Chen et al., 2013b). The main motivation of prior work is to use a FrameNet-trained statistical probabilistic semantic parser to generate initial frame-semantic parses from automatic speech recognition (ASR) decodings of the raw audio conversation files, and then adapt the FrameNet-style frames to the semantic slots in the target semantic space, so that they can be used practically in the SDSs. Chen et al. formulated the semantic mapping and adaptation problem as a ranking problem to differentiate generic semantic concepts from target semantic space for task-oriented dialogue systems. This paper improves the adaptation process by leveraging distributed word embeddings associated with typed syntactic dependencies between words to infer inter-slot relations (Mikolov et al., 2013b; Mikolov et al., 2013c; Levy and Goldberg, 2014). The proposed framework is shown in Figure 1. In the remainder of the section, we first introduce frame-semantic parsing to obtain slot candidates. With slot candidates, then we train the independent semantic decoders. The adaptation process, which is the main focus of this paper, is performed to decide outputted slots. Finally we can build an SLU model based on the learned semantic decoders and induced slots.

3.1 Probabilistic Semantic Parsing

FrameNet is a linguistically-principled semantic resource that offers annotations of predicate-argument semantics, and associated lexical units for English (Baker et al., 1998). FrameNet is developed based on a semantic theory, Frame Semantics (Fill-

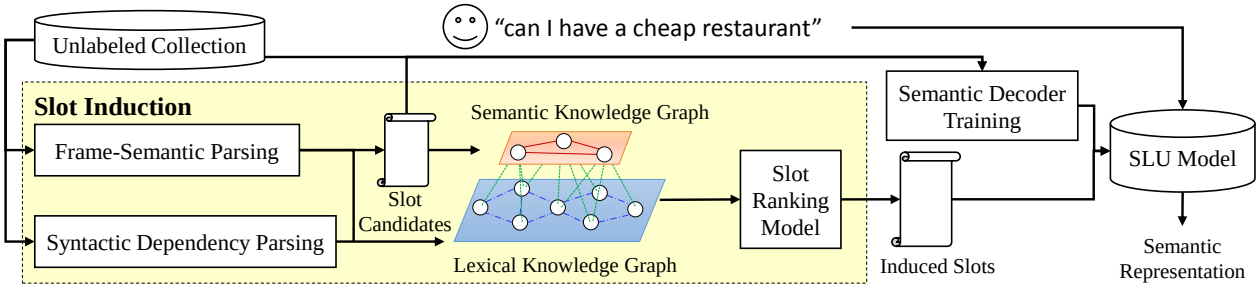

Figure 1: The proposed framework

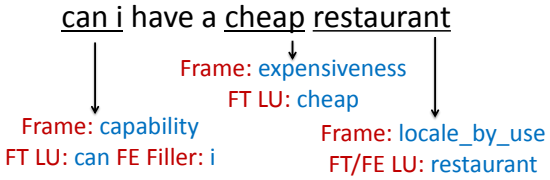

Figure 2: An example of probabilistic frame-semantic parsing on ASR output. FT: frame target. FE: frame element. LU: lexical unit.

more, 1976), which holds that the meaning of most words can be expressed on the basis of semantic frames, which encompass three major components: frame (F), frame elements (FE), and lexical units (LU). For example, the frame "food" contains words referring to items of food. A descriptor frame element within the food frame indicates the characteristic of the food. For example, the phrase "*low fat milk*" should be analyzed with "*milk*" evoking the food frame and "*low fat*" filling the descriptor FE of that frame.

In our approach, we parse all ASR-decoded utterances in our corpus using SEMAFOR[5], 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, where the LUs that correspond to the frames are extracted for slot filling. For example, Figure 2 shows an example of an ASR-decoded text output parsed by SEMAFOR. SEMAFOR generates three frames (capability, expensiveness, and locale_by_use) for the utterance, which we consider as slot candidates for training the SLU model. Note that for each slot candidate, SEMAFOR also includes the corresponding lexical unit (*can i*, *cheap*, and *restaurant*), which we consider as possible slot-fillers.

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

3.2 Independent Semantic Decoder

With outputted semantic parses, we extract the frames with the top 50 highest frequency as our slot candidates for training SLU. The features for training are generated by word confusion network, where confusion network features are shown to be useful in developing more robust systems for SLU (Hakkani-Tür et al., 2006; Henderson et al., 2012). We build a vector representation of an utterance as $\mathbf{u} = [x_1, ..., x_j, ...]$.

$$x_j = \mathbb{E}[C_u(n\text{-gram}_j)]^{1/|n\text{-gram}_j|}, \quad (1)$$

where $C_u(n\text{-gram}_j)$ counts how many times n-gram$_j$ occurs in the utterance u, $\mathbb{E}(C_u(n\text{-gram}_j))$ is the expected frequency of n-gram$_j$ in u, and $|n\text{-gram}_j|$ is the number of words in n-gram$_j$.

For each slot candidate s_i, we generate a pseudo training data $\mathcal{D}^i$ to train a binary classifier $\mathcal{M}^i$ for predicting the existence of s_i given an utterance, $\mathcal{D}^i = \{(\mathbf{u_k}, l_k^i) \mid \mathbf{u_k} \in \mathbb{R}^+, l_k^i \in \{-1, +1\}\}_{k=1}^K$, where $l_k^i = +1$ when the utterance u_k contains the slot candidate s_i in its semantic parse, $l_k^i = -1$ otherwise, and K is the number of utterances.

3.3 Adaptation Process and SLU 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

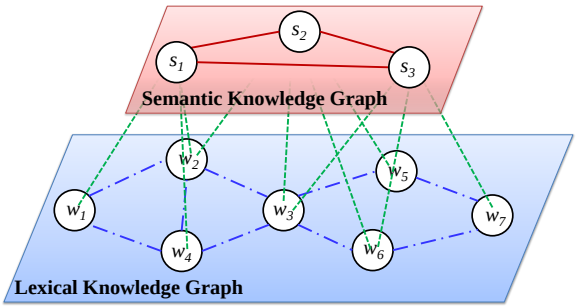

Figure 3: A simplified example of the two knowledge graphs, where a slot candidate s_i is represented as a node in a semantic knowledge graph and a word w_j is represented as a node in a lexical knowledge graph.

the "capability" frame does not convey particular valuable information for SLU. With the trained independent semantic decoders for all slot candidates, adaptation process computes the prominence of slot candidates for ranking and then selects a list of induced slots associated with their corresponding semantic decoders for use in domain-specific dialogue systems, where the detail is described in Section 4.

Then with each induced slot s_i and its corresponding trained semantic decoder $\mathcal{M}^i$, an SLU model can be built to predict whether the semantic slot occurs in the given utterance in a fully unsupervised way. In other words, the SLU model is able to transform the testing utterance into semantic representations without human involvement.

4 Slot Ranking Model

The purpose of the ranking model is to distinguish between generic semantic concepts and domain-specific concepts that are relevant to an SDS. To induce meaningful slots for the purpose of SDS, we compute the prominence of the slot candidates using a slot ranking model described below.

With the semantic parses from SEMAFOR, where each frame is viewed independently, so inter-slot relations are not included, the model ranks the slot candidates by integrating two information: (1) the frequency of each slot candidate in the corpus, since slots with higher frequency may be more important. (2) the relations between slot candidates. Assuming that domain-specific concepts are usually related to each other, globally considering inter-slot relations induces a more coherent slot set. Here for baseline

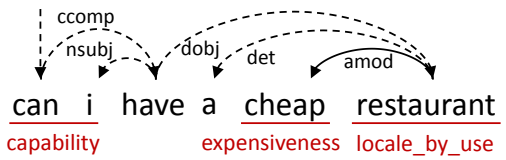

Figure 4: The dependency parsing result on an utterance.

scores, we only use the frequency of each slot candidate as its prominence.

First we construct two knowledge graphs, one is a slot-based semantic knowledge graph and another is a word-based lexical knowledge graph, both of which encode the typed dependency relations in their edge weights. We also connect two graphs to model the relations between slot-filler pairs.

4.1 Knowledge Graphs

We construct two undirected graphs, semantic and lexical knowledge graphs. Each node in the semantic knowledge graph is a slot candidate s_i outputted 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 $V_s = \{s_i\}$ 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 $V_w = \{w_i\}$ and $E_{ww} = \{e_{ij} \mid w_i, w_j \in V_w\}$.

With two knowledge graphs, we build the edges between slots and slot-fillers to integrate them as shown in Figure 3. Thus the combined graph can be formulated as $G = \langle V_s, V_w, E_{ss}, E_{ww}, E_{ws} \rangle$, where $E_{ws} = \{e_{ij} \mid w_i \in V_w, s_j \in V_s\}$. E_{ss}, E_{ww}, and E_{ws} correspond to slot-to-slot relations, word-to-word relations, and word-to-slot relations respectively (Chen and Metze, 2012; Chen and Metze, 2013).

4.2 Edge Weight Estimation

Considering the relations in the knowledge graphs, the edge weights for E_{ww} and E_{ss} are measured based on the dependency parsing results. The example utterance "*can i have a cheap restaurant*" and its dependency parsing result are illustrated in Figure 4. The arrows denote the de-

	Typed Dependency Relation	Target Word	Contexts
Word	⟨*restaurant*, AMOD, *cheap*⟩	*restaurant* *cheap*	*cheap*/AMOD *restaurant*/AMOD^{-1}
Slot	⟨locale_by_use, AMOD, expensiveness⟩	locale_by_use expansiveness	expensiveness/AMOD locale_by_use/AMOD^{-1}

Table 1: The contexts extracted for training dependency-based word/slot embeddings from the utterance of Fig. 2.

pendency 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 ⟨*restaurant*, AMOD, *cheap*⟩, ⟨*have*, DOBJ, *restaurant*⟩, 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, ⟨*restaurant*, AMOD, *cheap*⟩ from $\mathcal{T}_w$ is transformed into ⟨locale_by_use, AMOD, expensiveness⟩ for building $\mathcal{T}_s$, because both sides of the non-dotted line are parsed as slot-fillers by SEMAFOR.

For the edges within a single knowledge graph, we assign a weight of the edge connecting nodes x_i and x_j as $\hat{r}(x_i, x_j)$, where x is either s or w. Since the weights are measured based on the relations between nodes regardless of the directions, we combine the scores of two directional dependencies:

$$\hat{r}(x_i, x_j) = r(x_i \rightarrow x_j) + r(x_j \rightarrow x_i), \quad (2)$$

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. In Section 4.2.1 and 4.2.2, we propose two scoring functions for $r(\cdot)$, frequency-based as $r_1(\cdot)$ and embedding-based as $r_2(\cdot)$ respectively.

For the edges in E_{ws}, we estimate the edge weights based on the frequency that the slot candidates and the words are parsed as slot-filler pairs. In other words, the edge weight between the slot-filler w_i and the slot candidate s_j, $\hat{r}(w_i, s_j)$, is equal to how many times the filler w_i corresponds to the slot candidate s_j in the parsing results.

4.2.1 Frequency-Based Measurement

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

$$t^*_{x_i \rightarrow x_j} = \arg\max_t C(x_i \underset{t}{\rightarrow} x_j), \quad (3)$$

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 that estimates the dependency $x_i \rightarrow x_j$ is measured as

$$r_1(x_i \rightarrow x_j) = C(x_i \underset{t^*_{x_i \rightarrow x_j}}{\longrightarrow} x_j), \quad (4)$$

which equals to the highest observed frequency of the dependency $x_i \rightarrow x_j$ among all types from $\mathcal{T}_x$.

4.2.2 Embedding-Based Measurement

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 word w is associated with a word vector $\mathbf{v_w} \in$

$\mathbb{R}^d$ 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 words and contexts such that the dot product $\mathbf{v_w} \cdot \mathbf{v_c}$ associated with "good" word-context pairs belonging to the training data $\mathcal{D}$ is maximized, leading to the objective function:

$$\arg\max_{\mathbf{v_w},\mathbf{v_c}} \sum_{(w,c)\in\mathcal{D}} \log \frac{1}{1+\exp(-\mathbf{v_c}\cdot\mathbf{v_w})}, \quad (5)$$

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.

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

$$P(x_i \underset{t}{\rightarrow} x_j) = \frac{Sim(x_i, x_j/t) + Sim(x_j, x_i/t^{-1})}{2}, \quad (6)$$

where $Sim(x_i, x_j/t)$ is the cosine similarity between the slot/word embeddings $\mathbf{v_{x_i}}$ and the context embeddings $\mathbf{v_{x_j/t}}$ after normalizing to $[0, 1]$. Then we can measure the scoring function $r_2(\cdot)$ as

$$r_2(x_i \rightarrow x_j) = C(x_i \xrightarrow[t^*_{x_i\rightarrow x_j}]{} x_j) \cdot P(x_i \xrightarrow[t^*_{x_i\rightarrow x_j}]{} x_j), \quad (7)$$

which is similar to (4) but additionally weighted by the estimated probability. The estimated probability smooths the observed frequency to avoid overfitting due to a smaller dataset.

4.3 Random Walk Algorithm

We first compute $L_{ww} = [\hat{r}(w_i, w_j)]_{|V_w|\times|V_w|}$ and $L_{ss} = [\hat{r}(s_i, s_j)]_{|V_s|\times|V_s|}$, where $\hat{r}(w_i, w_j)$ and $\hat{r}(s_i, s_j)$ are either from frequency-based ($r_1(\cdot)$) or embedding-based measurements ($r_2(\cdot)$). Similarly, $L_{ws} = [\hat{r}(w_i, s_j)]_{|V_w|\times|V_s|}$ and $L_{sw} = [\hat{r}(w_i, s_j)]^T_{|V_w|\times|V_s|}$, where $\hat{r}(w_i, s_j)$ is the frequency that s_j and w_i are a slot-filler pair computed in Section 4.2. Then we only keep the top N highest weights for each row in L_{ww} and L_{ss} ($N = 10$), which means that we filter out the edges with smaller weights within the single knowledge graph. Column-normalization are performed for L_{ww}, L_{ss}, L_{ws}, L_{sw} (Shi and Malik, 2000). They can be viewed as word-to-word, slot-to-slot, and word-to-slot relation matrices.

4.3.1 Single-Graph Random Walk

Here we run random walk only on the semantic knowledge graph to propagate the scores based on inter-slot relations through the edges E_{ss}.

$$R_s^{(t+1)} = (1-\alpha)R_s^{(0)} + \alpha L_{ss} R_s^{(t)}, \quad (8)$$

where $R_s^{(t)}$ denotes the importance scores of the slot candidates V_s in t-th iteration. In the algorithm, the score is the interpolation of two scores, the normalized baseline importance of slot candidates ($R_s^{(0)}$), and the scores propagated from the neighboring nodes in the semantic knowledge graph based on slot-to-slot relations via L_{ss}. The algorithm will converge when $R_s^{(t+1)} = R_s^{(t)} = R_s^*$ and (9) can be satisfied.

$$R_s^* = \left((1-\alpha)R_s^{(0)}e^T + \alpha L_{ss}\right) R_s^* = M_1 R_s^*, \quad (9)$$

where $e = [1, 1, ..., 1]^T$. It has been shown that the closed-form solution R_s^* of (9) is the dominant eigenvector of M_1 (Langville and Meyer, 2005), the eigenvector corresponding to the largest absolute eigenvalue of M_1. The solution of R_s^* denotes the updated importance scores for all utterances. Similar to the PageRank algorithm (Brin and Page, 1998), the solution can also be obtained by iteratively updating $R_s^{(t)}$.

4.3.2 Double-Graph Random Walk

Here we borrow the idea from two-layer mutually reinforced random walk to propagate the scores based on not only internal importance propagation within the same graph but external mutual reinforcement between different knowledge graphs (Chen and Metze, 2012; Chen and Metze, 2013).

$$\begin{cases} R_s^{(t+1)} = (1-\alpha)R_s^{(0)} + \alpha L_{ss} L_{sw} R_w^{(t)} \\ R_w^{(t+1)} = (1-\alpha)R_w^{(0)} + \alpha L_{ww} L_{ws} R_s^{(t)} \end{cases} \quad (10)$$

In the algorithm, they are the interpolations of two scores, the normalized baseline importance ($R_s^{(0)}$ and $R_w^{(0)}$) and the scores propagated from another graph. For the semantic knowledge graph, $L_{sw}R_w^{(t)}$ is the score from the word set weighted by slot-to-word relations, and then the scores are propagated based on slot-to-slot relations via L_{ss}. Similarly, nodes of the lexical knowledge graph also include

the scores propagated from the semantic knowledge graph. Then $R_s^{(t+1)}$ and $R_w^{(t+1)}$ can be mutually updated by the latter parts in (10) iteratively. When the algorithm converges, we have R_s^* as follows.

$$\begin{aligned} R_s^* &= (1-\alpha)R_s^{(0)} \\ &+ \alpha L_{ss}L_{sw}\left((1-\alpha)R_w^{(0)} + \alpha L_{ww}L_{ws}R_s^*\right) \\ &= \Big((1-\alpha)R_s^{(0)}e^T + \alpha(1-\alpha)L_{ss}L_{sw}R_w^{(0)}e^T \\ &+ \alpha^2 L_{ss}L_{sw}L_{ww}L_{ws}\Big)R_s^* = M_2 R_s^*. \end{aligned} \tag{11}$$

The closed-form solution R_s^* of (11) is the dominant eigenvector of M_2.

5 Experiments

We evaluate our approach in two ways. First, we examine the slot induction accuracy by comparing the ranked list of induced slots with the reference slots created by system developers (Young, 2007). Secondly, with the ranked list of induced slots and their associated semantic decoders, we can evaluate the SLU performance. For the experiments, we evaluate both on ASR transcripts of the raw audio, and on the manual transcripts.

5.1 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 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 damping factor for random walk α is empirically set as 0.9 for all experiments. For training the semantic decoders, we use SVM with a linear kernel to predict each semantic slot. We use Stanford Parser to obtain the collapsed typed syntactic dependencies (Socher et al., 2013) and set the dimensionality of embeddings $d = 300$ in all experiments.

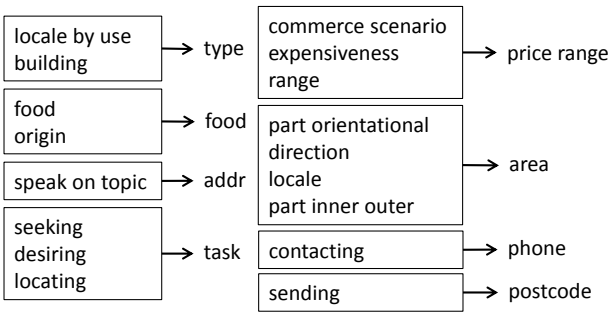

Figure 5: The mappings from induced slots (within blocks) to reference slots (right sides of arrows).

5.2 Evaluation Metrics

To eliminate the influence of threshold selection when choosing induced 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.

5.2.1 Slot Induction

To evaluate the accuracy of the induced slots, we measure their quality as the proximity between induced slots and reference slots. Figure 5 shows the mappings that indicate semantically related induced slots and reference slots (Chen et al., 2013b). For example, "expensiveness → price", "food → food", and "direction → area" show that these induced slots can be mapped into the reference slots defined by experts and carry important semantics in the target domain for developing the task-oriented SDS. Since we define the adaptation task as a ranking problem, with a ranked list of induced slots and associated scores, we can use the standard average precision (AP) and the area under the precision-recall curve (PR-AUC) as our metrics, where the induced slot is counted as correct when it has a mapping to a reference slot.

5.2.2 SLU Model

While semantic slot induction is essential for providing semantic categories and imposing semantic constraints, we are also interested in understanding the performance of our unsupervised SLU models.

Approach			ASR				Manual			
			Slot Induction		SLU Model		Slot Induction		SLU Model	
			AP	PR-AUC	WAP	AF	AP	PR-AUC	WAP	AF
(a)	Baseline (Frequency)		56.69	54.67	35.82	43.28	53.01	50.80	36.78	44.20
(b)	Single	Frequency	63.88	62.05	41.67	47.38	63.02	61.10	43.76	48.53
(c)		Embedding	69.04	68.25	**46.29**	**48.89**	75.15	74.50	**54.50**	**50.86**
(d)	Double	Frequency	56.83	55.31	32.64	44.91	52.12	50.54	34.01	45.05
(e)		Embedding	**71.48**	**70.84**	44.06	47.91	**76.42**	**75.94**	52.89	50.40

Table 2: The performance of induced slots and corresponding SLU models (%)

For each induced slot with the mapping to a reference slot, we can compute an F-measure of the corresponding semantic decoder, and weight the average precision with corresponding F-measure as weighted average precision (WAP) to evaluate the performance of slot induction and SLU tasks together. The metric scores the ranking result higher if the induced slots corresponding to better semantic decoders are ranked higher. Another metric is the average F-measure (AF), which is the average micro F-measure of SLU models at all cut-off positions in the ranked list. Compared to WAP, AF additionally considers the slot popularity in the dataset.

5.3 Evaluation Results

Table 5.1 shows the results on both ASR and manual transcripts. Rows (a) is the baseline only considering the frequency of each slot candidate for ranking (Chen et al., 2013b). Rows (b) and (c) show performance after leveraging a semantic knowledge graph through random walk. Rows (d) and (e) are the results after combining two knowledge graphs. We find almost all results are improved by additionally considering inter-slot relations in terms of single- and double-graph random walk for both ASR and manual transcripts.

5.3.1 Slot Induction

For both ASR and manual transcripts, almost all results outperform the baseline, showing that inter-slot relations significantly influence the performance of slot induction. The best performance is from the results of double-graph random walk with the embedding-based measurement, which integrate a semantic knowledge graph and a lexical knowledge graph together and jointly consider slot-to-slot, word-to-word, and word-to-slot relations when scoring the prominence of slot candidates to generate a coherent slot set.

5.3.2 SLU Model

For both ASR and manual transcripts, almost all results outperform the baseline, which shows the practical usage for training dialogue systems. The best performance is from the results of single-graph random walk with the embedding-based measurement, which only use the semantic knowledge graph to involve the inter-slot relations. The semantic knowledge graph is not as precise as the lexical one and may be influenced by the performance of the semantic parser more. Although the row (e) does not show better performance than the row (c), double-graph random walk may be more robust because it additionally includes the word relations to avoid only relying on the relations tied with the slot candidates.

5.4 Discussion and Analysis

5.4.1 Comparing Frequency- and Embedding-Based Measurements

Table 5.1 shows that all results with the embedding-based measurement perform better than those with the frequency-based measurement. The frequency-based measurement also brings large improvement for single-graph approaches, but does not for double-graph ones. The reason is probably that using observed frequencies in the lexical knowledge graph may result in overfitting issues due to the smaller dataset. Additionally including embedding information can smooth the edge weights and deal with data sparsity to improve the performance, especially for the lexical knowledge graph.

5.4.2 Comparing Single- and Double-Graph Approaches

Considering that the embedding-based measurement performs better, we only compare the results of single- and double-graph random walk using this measurement (rows (c) and (e)). It can be seen that the difference between them is not consistent in terms of slot induction and SLU model.

For evaluating slot induction (AP and PR-AUC), the double-graph random walk (row (e)) performs better on both ASR and manual results, which implies that additionally integrating the lexical knowledge graph helps decide a more coherent and complete slot set since we can model the score propagation more precisely (not only slot-level but word-level information). However, for SLU evaluation (WAP and AF), the single-graph random walk (row (c)) performs better, which may imply that the slots carrying the coherent relations from the row (e) may not have good semantic decoder performance so that the performance is decreased a little. For example, double-graph random walk scores the slots local_by_use and expensiveness higher than the slot contacting, while the single-graph method ranks the latter higher. local_by_use and expensiveness are more important on this domain but contacting has very good performance of its semantic decoder, so the double-graph approach does not show the improvement when evaluating SLU models. This allows us to try an improved method of jointly optimizing the slot coherence and SLU performance in the future.

5.4.3 Relation Discovery Analysis

To interpret the inter-slot relations, we output the slot-to-slot relations with highest scores from the best results (row (e) in Table 5.1) in Table 3, and the automatically constructed ontology is shown in Figure 6. It can be shown that the outputted inter-slot relations are reasonable and usually connect two important semantic slots in this restaurant domain. This proves that inter-slot relations help decide a coherent and complete slot set and enhance the interpretability of semantic slots. Therefore, from a practical perspective, developers are able to design the framework of dialogue systems more easily, and the development of SDS can be speeded up with less human effort.

Rank	Relation
1	⟨locale_by_use, NN, food⟩
2	⟨food, AMOD, expensiveness⟩
3	⟨locale_by_use, AMOD, expensiveness⟩
4	⟨seeking, PREP_FOR, food⟩
5	⟨food, AMOD, relational_quantity⟩
6	⟨desiring, DOBJ, food⟩
7	⟨seeking, PREP_FOR, locale_by_use⟩
8	⟨food, DET, quantity⟩

Table 3: The top inter-slot relations learned from the training set of ASR outputs.

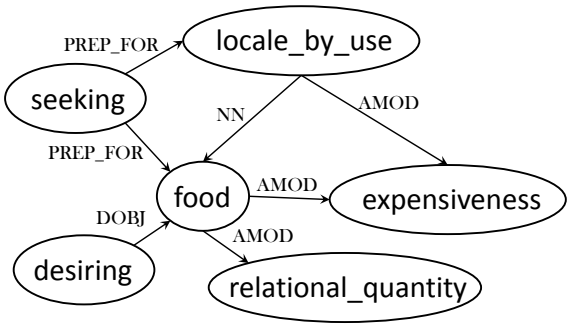

Figure 6: The automatically constructed domain-specific ontology based on Table 3.

6 Conclusion

The paper proposes an approach of jointly considering inter-slot relations for slot induction to output a more coherent slot set, where two knowledge graphs, a slot-based semantic knowledge graph and a word-based lexical knowledge graph, are built and combined by a random walk algorithm. The automatically induced slots carry coherent and interpretable relations and can be used for training better SLU models of SDSs in an unsupervised fashion.

Acknowledgments

We thank Anatole Gershman for helpful discussions and anonymous reviewers for their useful comments. 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.

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 *Advances in Neural Information Processing Systems*, pages 2787–2795.

Sergey Brin and Lawrence Page. 1998. The anatomy of a large-scale hypertextual web search engine. *Computer networks and ISDN systems*, 30(1):107–117.

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 Florian Metze. 2013. Multi-layer mutually reinforced random walk with hidden parameters for improved multi-party meeting summarization. In *INTERSPEECH*, pages 485–489.

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),*.

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 Gokhan 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),*.

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),*.

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*.

Marie-Catherine De Marneffe and Christopher D Manning. 2008. The Stanford typed dependencies representation. In *Coling 2008: Proceedings of the workshop on Cross-Framework and Cross-Domain Parser Evaluation*, pages 1–8. Association for Computational Linguistics.

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*.

Charles J Fillmore. 1976. Frame semantics and the nature of language. *Annals of the NYAS*, 280(1):20–32.

Dilek Hakkani-Tür, Frédéric Béchet, Giuseppe Riccardi, and Gokhan Tur. 2006. Beyond ASR 1-best: Using word confusion networks in spoken language understanding. *Computer Speech & Language*, 20(4):495–514.

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.

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

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

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.

Diana Inkpen and Graeme Hirst. 2006. Building and using a lexical knowledge base of near-synonym differences. *Computational Linguistics*, 32(2):223–262.

Amy N Langville and Carl D Meyer. 2005. A survey of eigenvector methods for web information retrieval. *SIAM review*, 47(1):135–161.

Ni Lao, Tom Mitchell, and William W Cohen. 2011. Random walk inference and learning in a large scale

knowledge base. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 529–539. Association for Computational Linguistics.

Ni Lao, Amarnag Subramanya, Fernando Pereira, and William W Cohen. 2012. Reading the web with learned syntactic-semantic inference rules. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1017–1026. Association for Computational Linguistics.

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

Peipei Li, Haixun Wang, Hongsong Li, and Xindong Wu. 2013a. Assessing sparse information extraction using semantic contexts. In *Proceedings of the 22nd ACM international conference on Conference on information & knowledge management*, pages 1709–1714. ACM.

Peipei Li, Haixun Wang, Kenny Q Zhu, Zhongyuan Wang, and Xindong Wu. 2013b. Computing term similarity by large probabilistic isa knowledge. In *Proceedings of the 22nd ACM international conference on Conference on information & knowledge management*, pages 1401–1410. ACM.

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.

Jianbo Shi and Jitendra Malik. 2000. Normalized cuts and image segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(8):888–905.

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

Yangqiu Song, Haixun Wang, Zhongyuan Wang, Hongsong Li, and Weizhu Chen. 2011. Short text conceptualization using a probabilistic knowledgebase. In *Proceedings of the Twenty-Second international joint conference on Artificial Intelligence-Volume Volume Three*, pages 2330–2336. AAAI Press.

Fang Wang, Zhongyuan Wang, Zhoujun Li, and Ji-Rong Wen. 2014. Concept-based short text classification and ranking. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, pages 1069–1078. ACM.

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

Steve Young. 2007. CUED standard dialogue acts. Technical report, Cambridge University Engineering Department.