Leveraging Frame Semantics and Distributional Semantics for Unsupervised Semantic Slot Induction in Spoken Dialogue Systems (Extended Abstract)

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

Abstract

Although the spoken dialogue system community in speech and the semantic parsing community in natural language processing share many similar tasks and approaches, they have progressed independently over years with few interactions. This paper connects two worlds to automatically induce the semantic slots for spoken dialogue systems using frame and distributional semantic theories. Given a collection of unlabeled audio, we exploit continuous-valued word embeddings to augment a probabilistic frame-semantic parser that identifies key semantic slots in an unsupervised fashion. Our experiments on a real-world spoken dialogue dataset show that distributional word representation significantly improves adaptation from FrameNet-style parses of recognized utterances to the target semantic space, that comparing to a state-of-the-art baseline, a 12% relative mean average precision improvement is achieved, and that the proposed technology can be used to reduce the costs for designing task-oriented spoken dialogue systems.

1 Introduction

Frame semantics is a linguistic theory that defines meaning as a coherent structure of related concepts (Fillmore, 1982). Although there has been some successful applications in natural language processing (Hedegaard and Simonsen, 2011; Coyne et al., 2011; Hasan and Ng, 2013), this linguistically principled theory has not been explored in the speech community until recently: Chen et al. (2013b) showed that it is possible to use probabilistic frame-semantic parsing to automatically induce and adapt the semantic ontology for designing spoken dialogue systems (SDS) in an unsupervised fashion. Comparing to the traditional approach where domain experts and developers manually define the semantic ontology for SDS, the unsupervised approach has the advantages to reduce the costs and avoid human induced bias.

On the other hand, the distributional view of semantics hypothesizes that words occurring in the same contexts may have similar meanings (Harris, 1954). With the recent advance of deep learning techniques, the continuous representation of word embeddings has further boosted the state-of-the-art results in many applications, such as frame identification (Hermann et al., 2014), sentiment analysis (Socher et al., 2013), language modeling (Mikolov, 2012), and sentence completion (Mikolov et al., 2013a).

In this paper, given a collection of unlabeled raw audio files, we investigate an unsupervised approach for semantic slot induction. To do this, we use a state-of-the-art probabilistic frame-semantic parsing approach (Das et al., 2010; Das et al., 2014), and perform an adaptation process, mapping the generic FrameNet (Baker et al., 1998) style semantic parses to the target semantic space that is suitable for the domain-specific conversation settings. We utilize continuous word embeddings trained on very large external corpora (e.g. Google News and Freebase) for the adaptation process. To evaluate the performance of our approach, we compare the automatically induced semantic slots with the reference slots created by domain experts. Empirical experiments show that the slot creation results generated by our approach align well with those of domain experts.

2 The Proposed Approach

We build our approach on top of the recent success of an unsupervised frame-semantic parsing approach. Chen et al. (2013b) formulated the se-

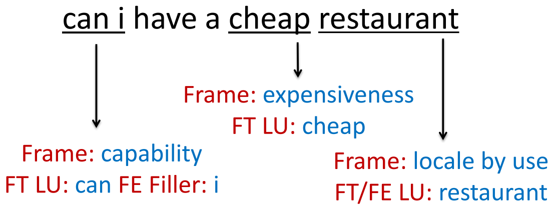

Figure 1: An example of probabilistic frame-semantic parsing on ASR output. FT: frame target. FE: frame element. LU: lexical unit.

mantic mapping and adaptation problem as a ranking problem, and proposed the use of unsupervised clustering methods to differentiate the generic semantic concepts from target semantic space for task-oriented dialogue systems. However, their clustering approach only performs on the small in-domain training data, which may not be robust enough. Therefore, this paper proposes a radical extension of the previous approach: we aim at improving the semantic adaptation process by leveraging distributed word representations.

2.1 Probabilistic Semantic Parsing

FrameNet is a linguistically-principled semantic resource (Baker et al., 1998), developed based on the frame semantics theory (Fillmore, 1976). In our approach, we parse all ASR-decoded utterances in our corpus using SEMAFOR, a state-of-the-art semantic parser for frame-semantic parsing (Das et al., 2010; Das et al., 2014), 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 1, shows an example of SEMAFOR parsing of an ASR-decoded text output.

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 1, we see that the "expensiveness" and "locale by use" frames are essentially the key slots for the purpose of understanding in the restaurant query domain, whereas the "capability" frame does not convey particular valuable information for SLU. In order to fix this issue, we compute the prominence of these slot candidates, use a slot ranking model to rerank the most frequent slots, and then generate a list of induced slots for use in domain-specific dialogue systems.

2.2 Continuous Space Word Representations

To better adapt the FrameNet-style parses to the target task-oriented SDS domain, we make use of continuous word vectors derived from a recurrent neural network architecture (Mikolov et al., 2010). The recurrent neural network language models use the context history to include long-distance information. Interestingly, the vector-space word representations learned from the language models were shown to capture syntactic and semantic regularities (Mikolov et al., 2013c; Mikolov et al., 2013b). The word relationships are characterized by vector offsets, where in the embedded space, all pairs of words sharing a particular relation are related by the same constant offset. Considering that this distributional semantic theory may benefit our SLU task, we leverage word representations trained from large external data to differentiate semantic concepts.

2.3 Slot Ranking Model

Our model ranks the slot candidates by integrating two scores (Chen et al., 2013b): (1) the relative frequency of each candidate slot in the corpus, since slots with higher frequency may be more important. (2) the coherence of slot-fillers corresponding to the slot. Assuming that domain-specific concepts focus on fewer topics and are similar to each other, the coherence of the corresponding values can help measure the prominence of the slots.

$$w(s_i) = (1 - \alpha) \cdot \log f(s_i) + \alpha \cdot \log h(s_i), \quad (1)$$

where $w(s_i)$ is the ranking weight for the slot candidate s_i, $f(s_i)$ is the frequency of s_i from semantic parsing, $h(s_i)$ is the coherence measure of s_i, and α is the weighting parameter within the interval $[0, 1]$.

For each slot s_i, we have the set of corresponding slot-fillers, $V(s_i)$, constructed from the utterance including the slot s_i in the parsing results. The coherence measure $h(s_i)$ is computed as average pair-wise similarity of slot-fillers to evaluate if slot s_i corresponds to centralized or scattered topics.

$$h(s_i) = \frac{\sum_{x_a, x_b \in V(s_i), x_a \neq x_b} \mathrm{Sim}(x_a, x_b)}{|V(s_i)|^2}, \quad (2)$$

where $V(s_i)$ is the set of slot-fillers corresponding slot s_i, $|V(s_i)|$ is the size of the set, and

$\mathrm{Sim}(x_a, x_b)$ is the similarity between the pair of fillers x_a and x_b. The slot s_i with higher $h(s_i)$ usually focuses on fewer topics, which is more specific and more likely for slots occurring in dialogue systems.

We involve distributional semantics of slot-fillers x_a and x_b for deriving $\mathrm{Sim}(x_a, x_b)$. Here, we propose two similarity measures: the representation-derived similarity and the neighbor-derived similarity as $\mathrm{Sim}(x_a, x_b)$ in (2).

2.3.1 Representation-Derived Similarity

Given that distributional semantics can be captured by continuous space word representations (Mikolov et al., 2013c), we transform each token x into its embedding vector $\mathbf{x}$ by pre-trained distributed word representations, and then the similarity between a pair of slot-fillers x_a and x_b can be computed as their cosine similarity, called $\mathrm{RepSim}(x_a, x_b)$.

We assume that words occurring in similar domains have similar word representations thus $\mathrm{RepSim}(x_a, x_b)$ will be larger when x_a and x_b are semantically related. The representation-derived similarity relies on the performance of pre-trained word representations, and higher dimensionality of embedding words results in more accurate performance but greater complexity.

2.3.2 Neighbor-Derived Similarity

With embedding vector $\mathbf{x}$ corresponding token x in the continuous space, we build a vector $\mathbf{r}_x = [r_x(1), ..., r_x(t), ..., r_x(T)]$ for each x, where T is the vocabulary size, and the t-th element of $\mathbf{r}_x$ is defined as

$$r_x(t) = \begin{cases} \frac{\mathbf{x} \cdot \mathbf{y}_t}{\|\mathbf{x}\| \|\mathbf{y}_t\|} & \text{, if } y_t \text{ is the word whose embedding} \\ & \text{vector has top } N \text{ greatest similarity} \\ & \text{to } \mathbf{x}. \\ 0 & \text{, otherwise.} \end{cases} \tag{3}$$

The t-th element of vector $\mathbf{r}_x$ is the cosine similarity between the embedding vector of slot-filler x and the t-th embedding vector y_t of pre-trained word representations (t-th token in the vocabulary of external larger dataset), and we only include the elements with top N greatest values to form a sparse vector for space reduction (from T to N). $\mathbf{r}_x$ can be viewed as a vector indicating the N nearest neighbors of token x obtained from continuous word representations. Then the similarity between a pair of slot-fillers x_a and x_b, $\mathrm{Sim}(x_a, x_b)$ in (2), can be computed as the cosine similarity between $\mathbf{r}_{x_a}$ and $\mathbf{r}_{x_b}$, called $\mathrm{NeiSim}(x_a, x_b)$.

The idea of using $\mathrm{NeiSim}(x_a, x_b)$ is very similar as using $\mathrm{RepSim}(x_a, x_b)$, where we assume that words with similar concepts should have similar representations and share similar neighbors. Hence, $\mathrm{NeiSim}(x_a, x_b)$ is larger when x_a and x_b have more overlapped neighbors in continuous space.

3 Experiments

We examine the slot induction accuracy by comparing the reranked list of frame-semantic parsing induced slots with the reference slots created by system developers. Furthermore, using the reranked list of induced slots and their associated slot fillers (value), we compare against the human annotation. For the slot-filling task, we evaluate both on ASR transcripts of the raw audio, and on the manual transcripts.

3.1 Experimental Setup

In this experiment, we used the Cambridge University spoken language understanding 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 spoken dialogue systems, in an in-car setting. The corpus contains a total number of 2,166 dialogues, and 15,453 utterances. The ASR system that was used to transcribe the speech has a word error rate of 37%. There are 10 slots created by domain experts: addr, area, food, name, phone, postcode, price range, signature, task, and type. The parameter α in (1) can be empirically set; we use $\alpha = 0.2, N = 100$ for all experiments.

To include distributional semantics information, we use two lists of pre-trained distributed vectors described as below[1].

- **Word/Phrase Vectors from Google News**: word vectors are trained on 10^9 words from Google News, using the continuous bag of words architecture, which predicts the current word based on the context. The resulting vectors have dimensionality 300, vocabulary size is 3×10^6; the entities contain both words and automatically derived phrases.

[1] `https://code.google.com/p/word2vec/`

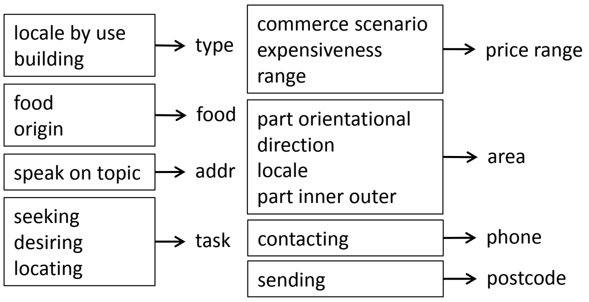

Figure 2: The mappings from induced slots (within blocks) to reference slots (right sides of arrows).

- **Entity Vectors with Freebase Naming**: the entity vectors are trained on 10^9 words from Google News with naming from Freebase[2]. The training was performed using the continuous skip gram architecture, which predicts surrounding words given the current word. The resulting vectors have dimensionality 1000, vocabulary size is 1.4×10^6, and the entities contain the deprecated `/en/` naming from Freebase.

The first dataset provides a larger vocabulary and better coverage; the second has more precise vectors, using knowledge from Freebase.

3.2 Evaluation Metrics

To evaluate the induced slots, we measure their quality as the proximity between induced slots and reference slots. Figure 2 shows many-to-many mappings that indicate semantically related induced slots and reference slots (Chen et al., 2013b). Since we define the adaptation task as a ranking problem, with a ranked list of induced slots, we can use the standard mean average precision (MAP) as our metric, where the induced slot is counted as correct when it has a mapping to a reference slot.

To evaluate slot fillers, for each matched mapping between the induced slot and the reference slot, we compute an F-measure by comparing the lists of extracted slot fillers corresponding to the induced slots, and the slot fillers in the reference list. Since slot fillers may contain multiple words, we use hard and soft matching to define whether two slot fillers match each other, where "hard" requires that the two slot fillers should be exactly the same; "soft" means that two slot fillers match if they share at least one overlapping word. We weight MAP scores with corresponding F-measure as MAP-F-H (hard) and MAP-F-S (soft) to evaluate the performance of slot induction and slot-filling tasks together (Chen et al., 2013b).

3.3 Evaluation Results

Table 1 shows the results. Rows (a)-(c) are the baselines without leveraging distributional word representations trained on external data, where row (a) is the baseline only using frequency for ranking, and rows (b) and (c) are the results of clustering-based ranking models in the prior work (Chen et al., 2013b). Rows (d)-(j) show performance after leveraging distributional semantics. Rows (d) and (e) are the results using representation- and neighbor-derived similarity from Google News data respectively, while row (f) and row (g) are the results from Freebase naming data. Rows (h)-(j) are performance of late fusion, where we use voting to combine the results of two data considering coverage of Google data and precision of Freebase data. We find almost all results are improved by including distributed word information.

For ASR results, the performance from Google News and Freebase is similar. Rows (h) and (i) fuse the two systems. For ASR, MAP scores are slightly improved by integrating coverage of Google News and accuracy from Freebase (from about 72% to 74%), but MAP-F scores do not increase. This may be because some correct slots are ranked higher after combining the two sources of evidence, but their slot-fillers do not perform well enough to increase MAP-F scores.

To compare the representation-derived (RepSim) and neighbor-derived (NeiSim) similarities, for both ASR and manual transcripts, neighbor-derived similarity performs better on Google News data (row (d) v.s. row (e)). The reason may be that neighbor-derived similarity considers more semantically-related words to measure similarity (instead of only two tokens), while representation-derived similarity is directly based on trained word vectors, which may degrade with recognition error. In terms of Freebase data, rows (f) and (g) do not show significant improvement, probably because entities in Freebase are more precise and their word representations have higher accuracy. Hence, considering additional neighbors in continuous vector space does not ob-

[2] `http://www.freebase.com/`

Table 1: The performance of induced slots and corresponding slot-fillers (%)

Approach				ASR			Manual		
				MAP	MAP-F-H	MAP-F-S	MAP	MAP-F-H	MAP-F-S
Frame Sem	(a)	Frequency (baseline)		67.31	26.96	27.29	59.41	27.29	28.68
	(b)	K-Means		67.38	27.38	27.99	59.48	27.67	28.83
	(c)	Spectral Clustering		68.06	30.52	28.40	59.77	30.85	29.22
Frame Sem + Dist Sem	(d)	Google News	RepSim	72.71	31.14	31.44	66.42	32.10	33.06
	(e)		NeiSim	73.35	**31.44**	**31.81**	68.87	**37.85**	**38.54**
	(f)	Freebase	RepSim	71.48	29.81	30.37	65.35	34.00	35.04
	(g)		NeiSim	73.02	30.89	30.72	64.87	31.05	31.86
	(h)	(d) + (f)		74.60	29.82	30.31	66.91	34.84	35.90
	(i)	(e) + (g)		74.34	31.01	31.28	**68.95**	33.73	34.28
	(j)	(d) + (e) + (f) + (g)		**76.22**	30.17	30.53	66.78	32.85	33.44

tain improvement, and fusion of results from two sources (rows (h) and (i)) cannot perform better. However, note that neighbor-derived similarity requires less space for computational procedure and sometimes produces results the same or better as the representation-derived similarity.

Overall, we see that all combinations that leverage distributional semantics outperform only using frame semantics; this demonstrates the effectiveness of applying distributional information to slot induction. The 76% MAP performance indicates that our proposed approach can generate good coverage for domain-specific slots in a real-world SDS. While we present results in the SLU domain, it should be possible to apply our approach to text-based NLU and slot filling tasks.

4 Conclusion

We propose the first unsupervised approach unifying frame and distributional semantics for the automatic induction and filling of slots. Our work makes use of a state-of-the-art semantic parser, and adapts the generic linguistically-principled FrameNet representation to a semantic space characteristic of a domain-specific SDS. With the incorporation of distributional word representations, we show that our automatically induced semantic slots align well with reference slots. We show feasibility of this approach, including slot induction and slot-filling for dialogue tasks.

References

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

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 ASRU*, pages 120–125.

Bob Coyne, Daniel Bauer, and Owen Rambow. 2011. Vignet: Grounding language in graphics using frame semantics. In *Proceedings of the ACL 2011 Workshop on Relational Models of Semantics*, pages 28–36.

Dipanjan Das, Nathan Schneider, Desai Chen, and Noah A Smith. 2010. Probabilistic frame-semantic parsing. In *Proceedings of NAACL-HLT*, pages 948–956.

Dipanjan Das, Desai Chen, André F. T. Martins, Nathan Schneider, and Noah A. Smith. 2014. Frame-semantic parsing. *Computational Linguistics*, 40(1):9–56.

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

Charles Fillmore. 1982. Frame semantics. *Linguistics in the morning calm*, pages 111–137.

Zellig S Harris. 1954. Distributional structure. *Word*.

Kazi Saidul Hasan and Vincent Ng. 2013. Frame semantics for stance classification. *CoNLL-2013*, page 124.

Steffen Hedegaard and Jakob Grue Simonsen. 2011. Lost in translation: authorship attribution using frame semantics. In *Proceedings of ACL-HLT*, pages 65–70.

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.

Karl Moritz Hermann, Dipanjan Das, Jason Weston, and Kuzman Ganchev. 2014. Semantic frame identification with distributed word representations. In *Proceedings of ACL*.

Tomas Mikolov, Martin Karafiát, Lukas Burget, Jan Cernockỳ, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In *Proceedings of INTERSPEECH*, pages 1045–1048.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. In *Proceedings of 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 NIPS*, pages 3111–3119.

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

Tomáš Mikolov. 2012. *Statistical language models based on neural networks*. Ph.D. thesis, Brno University of Technology.

Richard Socher, Alex Perelygin, Jean Y Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of EMNLP*, pages 1631–1642.