LEVERAGING FRAME SEMANTICS AND DISTRIBUTIONAL SEMANTICS FOR UNSUPERVISED SEMANTIC SLOT INDUCTION IN SPOKEN DIALOGUE SYSTEMS

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

Distributional semantics and frame semantics are two representative views on language understanding in the statistical world and the linguistic world, respectively. In this paper, we combine the best of two worlds to automatically induce the semantic slots for spoken dialogue systems. Given a collection of unlabeled audio files, we exploit continuous-valued word embeddings to augment a probabilistic frame-semantic parser that identifies key semantic slots in an unsupervised fashion. In experiments, our results on a real-world spoken dialogue dataset show that the distributional word representations significantly improve the adaptation of FrameNet-style parses of ASR decodings to the target semantic space; that comparing to a state-of-the-art baseline, a 13% relative average precision improvement is achieved by leveraging word vectors trained on two 100-billion words datasets; and that the proposed technology can be used to reduce the costs for designing task-oriented spoken dialogue systems.

Index Terms— Unsupervised slot induction, distributional semantics, frame semantics.

1. INTRODUCTION

The distributional view of semantics hypothesizes that words occurring in the same contexts may have similar meanings [1]. As the foundation for modern statistical semantics [2], an early success that implements this distributional theory is Latent Semantic Analysis [3]. Recently, with the advance of deep learning techniques, the continuous representations as word embeddings have further boosted the state-of-the-art results in many applications, such as sentiment analysis [4], language modeling [5], sentence completion [6], and relation detection [7].

Frame semantics, on the other hand, is a linguistic theory that defines meaning as a coherent structure of related concepts [8]. Although there has been some successful applications in natural language processing (NLP) [9, 10, 11], this linguistically principled theory has not been explored in the speech community until recently: Chen *et al.* showed that it was 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 [12], alleviating some of the challenging problems for developing and maintaining spoken language understaning (SLU)-based interactive systems [13]. Comparing to the traditional approach where domain experts and developers manually define the semantic ontology for SDS, the unsupervised semantic induction approach proposed by Chen *et al.* has the advantages to reduce the costs of annotation, avoid human induced bias, and lower the maintenance costs [12].

In this paper, we further improve the state-of-the-art results by leveraging the continuous variant of distributional word embeddings to identify key semantic slots for designing the SLU component in SDS. Given a collection of unlabeled raw audio files, we investigate an unsupervised approach for automatic induction of semantic slots. To do this, we use a state-of-the-art probabilistic frame-semantic parsing approach [14], and perform an unsupervised approach to adapt, rerank, and map the generic FrameNet[1]-style semantic parses to the target semantic space that is suitable for the domain-specific conversation settings [15]. We utilize continuous word embeddings trained on very large external corpora (e.g. Google News and Freebase) to improve 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. Our main contributions of this paper are three-fold:

- We exploit continuous-valued word embeddings for unsupervised spoken language understanding (SLU);
- We propose the first approach of combining distributional and frame semantics for inducing semantic slots from unlabeled speech data;
- We show that this synergized method yields the state-of-the-art performance.

2. RELATED WORK

The idea of leveraging external semantic resources for unsupervised SLU was popularized by the work of Heck and Hakkani-Tür , and Tür *et al.* [16, 17]. The former exploited Semantic Web for the intent detection problem in SLU, and showed that the results obtained from the unsupervised training process align well with the performance of traditional supervised learning [16]. The latter used search queries and obtained promising results on the slot filling task in the movie domain [17]. Following the success of the above applications, recent studies have also obtained interesting results on the tasks of relation detection [18], entity extraction [19], and extending domain coverage [20]. The major difference between our work and previous studies is that, instead of leveraging the discrete representations of Bing search queries or Semantic Web, we build our model on top of the recent success of deep learning—we utilize the continuous-valued word embeddings trained on Google News and Freebase to induce semantic ontology for task-oriented SDS.

Our approach is clearly relevant to recent studies on deep learning for SLU. Tür *et al.* have shown that deep convex networks are

[1] http://framenet.icsi.berkeley.edu

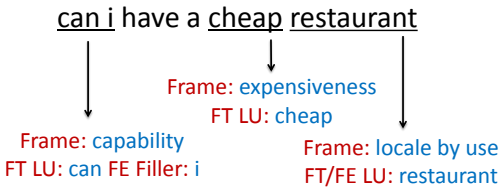

Fig. 1. An example of probabilistic frame-semantic parsing on ASR output. FT: frame target. FE: frame element. LU: lexical unit.

effective for building better semantic utterance classification systems [21]. Following their success, Deng *et al.* have further demonstrated the effectiveness of applying the kernel trick to build better deep convex networks for SLU [22]. To the best of our knowledge, our work is the first study that combines the distributional view of meaning from the deep learning community, and the linguistically principled frame semantic view for improved SLU.

3. THE PROPOSED APPROACH

We build our approach on top of the recent success of an unsupervised frame-semantic parsing approach [12]. The main motivation 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. Then adapt the FrameNet-style frame-semantic parses 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, and proposed the use of unsupervised clustering methods to differentiate the generic semantic concepts from target semantic space for task-oriented dialogue systems [12]. 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 that are trained on very large external datasets [23, 24].

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 [15]. FrameNet is developed based on semantic theory, Frame Semantics [25]. The theory 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[2], a state-of-the-art semantic parser for frame-semantic parsing [14, 26], 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 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. 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.

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 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 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 rank the most important slots, and then generate a list of induced slots for use in domain-specific dialogue systems.

3.2. Continuous Space Word Representations

In NLP, the Brown *et al.* [27] clustering algorithm is an early hierarchical clustering algorithm that extracts word clusters from large corpora, which has been used successfully in many NLP applications [28]. Comparing to traditional bag-of-words (BoWs) and n-gram language models, in recent years, continuous word embeddings (a.k.a. word representations, or neural language models) are shown to be the state-of-the-art in many NLP tasks, due to its rich continuous representations (e.g. vectors, or sometimes matrices, and tensors) that capture the context of the target semantic unit [29, 30].

Considering that this distributional semantic theory may benefit our SLU task, we leverage word representations trained from large external data to differentiate semantic concepts. The rationale behind applying the distributional semantic theory to our task is straight-forward: because spoken language is a very distinct genre comparing to the written language on which FrameNet is constructed, it is necessary to borrow external word representations to help bridge these two data sources for the unsupervised adaptation process.

More specifically, 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 [31]. 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 [23, 24]. 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.

3.3. 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, the model ranks the slot candidates by integrating two scores [12]: (1) the normalized frequency of each slot candidate 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, the coherence of the corresponding slot-fillers can help measure the prominence of the slots because they are simi-

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

lar to each other.

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

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

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

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

where $V(s)$ is the set of slot-fillers corresponding slot s, $|V(s)|$ is the size of the set, and $\mathrm{Sim}(x_a, x_b)$ is the similarity between the slot-filler pair x_a and x_b. The slot s with higher $h(s)$ usually focuses on fewer topics, which is more specific and more likely to be a slot for the dialogue system.

We leverage distributed word representations introduced in Section 3.2 to 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) below.

3.3.1. Representation-Derived Similarity

Given that distributional semantics can be captured by continuous space word representations [23], 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

$$\mathrm{RepSim}(x_a, x_b) = \frac{\mathbf{x_a} \cdot \mathbf{x_b}}{\|\mathbf{x_a}\|\|\mathbf{x_b}\|}. \quad (3)$$

We assume that words occurring in similar domains have similar word representations, and 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.

3.3.2. 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

$$r_x(t) = \begin{cases} \frac{\mathbf{x}\cdot\mathbf{y}_t}{\|\mathbf{x}\|\|\mathbf{y}_t\|} & \text{, if } y_t \text{ is the word whose embedding vector has top } N \text{ greatest similarity to } \mathbf{x}. \\ 0 & \text{, otherwise.} \end{cases} \quad (4)$$

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 remain 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

$$\mathrm{NeiSim}(x_a, x_b) = \frac{\mathbf{r}_{x_a} \cdot \mathbf{r}_{x_b}}{\|\mathbf{r}_{x_a}\|\|\mathbf{r}_{x_b}\|}. \quad (5)$$

The idea of $\mathrm{NeiSim}(x_a, x_b)$ is very similar as $\mathrm{RepSim}(x_a, x_b)$, where we assume that words with similar concepts should have similar representations and share similar neighbors. Hence, the value of $\mathrm{NeiSim}(x_a, x_b)$ is larger when x_a and x_b have more overlapped neighbors in the continuous space. Also, the complexity can be reduced by setting the number of neighbors considered.

3.4. Late Fusion

With multiple ranked lists of induced slots, where $h(s)$ in (1) can be derived from representation-derived similarity or neighbor-derived similarity, and the word embeddings can be trained on different resources, we can fuse multiple results by a simple voting method to combine multiple lists. Here we define the ranking score of slot s in the ranked list l as

$$R_l(s) = k, \quad (6)$$

where k is the position where the slot s is ranked. Then given multiple ranked lists $L = \{l\}$, the fused ranking score of slot s is computed as its average position from all lists.

$$R_L(s) = \frac{1}{|L|} \sum_{l \in L} R_l(s) \quad (7)$$

For example, a slot is ranked at t-th, u-th, and v-th positions by three approaches, and its average position from these three lists is $\frac{1}{3}(t+u+v)$. Finally we can obtain fused ranked list by ranking the slots via $R_L(s)$. Note that the slots with smaller scores are ranked higher, because the smaller scores imply that the slots are ranked higher by multiple methods.

4. EXPERIMENTS

To evaluate the effectiveness of our approach, we performed two evaluations. First, 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 [32]. Secondly, using the reranked list of induced slots and their associated slot-fillers, we compare against the human annotation. For the experiments, we evaluate both on ASR transcripts of the raw audio, and on the manual transcripts.

4.1. Experimental Setup

In this experiment, we used the Cambridge University SLU corpus, previously used on several other SLU tasks [33, 34]. 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 11,288 utterances with semantic slots. 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. 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[3].

- Word and Phrase Vectors from Google News
  The word vectors are trained on 10^9 words from Google News. Training was performed 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.
- Entity Vectors with Freebase Naming
  The entity vectors are trained on 10^9 words from Google News with naming from Freebase[4]. 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.

4.2. Evaluation Metrics

4.2.1. Slot Induction

To evaluate the accuracy of the induced slots, we measure their quality as the proximity between induced slots and reference slots. Figure 2 shows the mappings that indicate semantically related induced slots and reference slots [12]. 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. Note that two slots, name and signature, do not have proper mappings, because they are too specific on restaurant-related domain, where name records the name of restaurant and signature refers to signature dishes. This means that the 80% recall is achieved by our approach because we consider all outputted frames as slot candidates.

Since we define the adaptation task as a ranking problem, with a ranked list of induced slots, we can use the standard average precision (AP) as our metric, where the induced slot is counted as correct when it has a mapping to a reference slot. For a ranked list of induced slots $l = s^1, ..., s^k, ...$, where the s^k is the induced slot ranked at k-th position, the average precision is

$$\text{AP}(l) = \frac{\sum_{k=1}^{n} P(k) \times \mathbb{1}[s^k \text{ has a mapping to a reference slot}]}{\text{number of induced slots with mapping}}, \tag{8}$$

where $P(k)$ is the precision at cut-off k in the list and $\mathbb{1}$ is an indicator function equaling 1 if ranked k-th induced slot s^k has a mapping to a reference slot, 0 otherwise. Since the slots generated by our method cover only 80% of the referenced slots, the oracle recall is 80%. Therefore, average precision is a proper way to measure the slot ranking problem, which is also an approximation of the area under the precision-recall curve (AUC-PR) [35].

4.2.2. Slot Induction and Filling

While semantic slot induction is essential for providing semantic categories and imposing semantic constraints, we are also interested in understanding the performance of our induced slot-fillers. For each matched mapping between the induced slot and the reference slot, we can 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 the average precision with corresponding F-measure as AP-F to evaluate the performance of slot induction and slot filling tasks together [12].

$$\text{AP-F}(l) = \frac{\sum_{k=1}^{n} P(k) \times F(k)}{\text{number of induced slots with mapping}}, \tag{9}$$

where we replace $rel(k)$ in (8) with $F(k)$, which is the F-measure of slot-fillers corresponding s^k if it has mapping to reference slot, 0 otherwise. The metric scores the ranking result higher if an induced slot with more accurate slot-fillers.

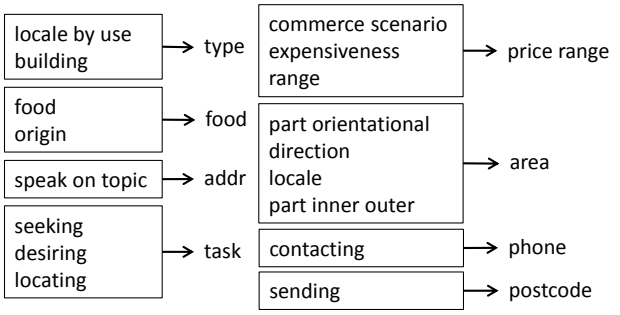

Fig. 2. The mappings from induced slots (within blocks) to reference slots (right sides of arrows).

4.3. 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 [12]. 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 combine the results of two data considering coverage of Google data and precision of Freebase data by the method described in Section 3.4. We find almost all results are improved by including distributed word information.

4.3.1. Comparison between Google News and Freebase Data

For ASR results, the performance from Google News and Freebase is similar. However, for manual transcripts, Google News performs better than Freebase (rows (d)-(e) v.s. rows (f)-(g)), probably because manual transcripts include more correct words, which can benefit from useful word representations trained on the larger vocabulary in Google News. Rows (h) and (i) fuse the two systems. For ASR, AP scores are slightly improved by integrating coverage of Google News and accuracy from Freebase (from about 72% to 74%), but AP-F scores do not increase. This may be because some correct slots are ranked higher after combining the two sources of evidence,

[3] `https://code.google.com/p/word2vec/`
[4] `http://www.freebase.com/`

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

Approach				ASR Transcripts			Manual Transcripts		
				AP	AP-F (Hard)	AP-F (Soft)	AP	AP-F (Hard)	AP-F (Soft)
Frame Semantics	(a)	Baseline: Frequency		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 Semantics + Dist. Semantics	(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
Maximum Relative Improvement (%)				+13.2	+16.6	+16.6	+16.1	+38.7	+34.4

but their slot-fillers do not perform well enough to increase AP-F scores. For manual transcripts, all results of combining two data do not show significant improvement. For manual transcripts, Google News may provide good enough word representations for computing coherence in our ranking model; combining with Freebase will not help in this case.

4.3.2. Comparing Different Similarity Measures

We evaluate two approaches of computing distributional semantic similarity: representation-derived (RepSim) and neighbor-derived similarity (NeiSim). 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 errors. In terms of Freebase data, rows (f) and (g) do not have significant difference, probably because entities in Freebase are more precise and their word representations have higher accuracy. Hence, considering additional neighbors in the continuous vector space does not obtain improvement, and fusion of results from two sources (rows (h) and (i)) does not show the difference between two similarity measures. However, note that neighbor-derived similarity requires less space for the computational procedure and sometimes produces results the same or better as the representation-derived similarity.

4.3.3. Overall Results

For slot induction task, combining all results of two corpora and differently derived similarities achieves the best average precision of 76.22% on ASR output (row (j)), while the best average precision on manual transcripts is 68.95% performed by combing the results of neihbor-derived similarity on two data (row (i)). The reason about better AP scores of ASR may be that users tend to speak keywords clearer than generic words, higher word error rate of generic words makes these slot candidates ranked lower due to lower frequency and coherence. For slot filling task, the best results performance on both ASR and manual transcripts is from neighbor-derived similarity using word vectors trained on Google News, because considering more semantically-related words helps recovering slot-fillers. Note that for evaluation of slot filling, some slot-fillers cannot match to reference slot-fillers due to recognition errors, which accounts for the reason that the results of manual transcripts is better than ASR performance.

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% of AP 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 natural language understanding and slot filling tasks, reducing labor cost.

5. CONCLUSION

We propose the first unsupervised approach unifying distributional and frame 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, yielding the state-of-the-art performance. Also, we demonstrate the feasibility of automatically induced slots from this approach for benefiting SLU tasks.

6. REFERENCES

[1] Zellig S Harris, "Distributional structure," *Word*, 1954.

[2] GW Furnas, TK Landauer, LM Gomez, and ST Dumais, "Statistical semantics: Analysis of the potential performance of keyword information systems," in *Proceedings of Human Factors in Computer Systems*, 1984, pp. 187–242.

[3] Scott C. Deerwester, Susan T Dumais, Thomas K. Landauer, George W. Furnas, and Richard A. Harshman, "Indexing by latent semantic analysis," *JASIS*, vol. 41, no. 6, pp. 391–407, 1990.

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

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

[6] Tomáš Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean, "Efficient estimation of word representations in vector space," in *Proceedings of ICLR*, 2013.

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

[8] Charles Fillmore, "Frame semantics," *Linguistics in the morning calm*, pp. 111–137, 1982.

[9] Steffen Hedegaard and Jakob Grue Simonsen, "Lost in translation: authorship attribution using frame semantics," in *Proceedings of ACL-HLT*, 2011, pp. 65–70.

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

[11] Kazi Saidul Hasan and Vincent Ng, "Frame semantics for stance classification," *CoNLL-2013*, p. 124, 2013.

[12] Yun-Nung Chen, William Yang Wang, and Alexander I Rudnicky, "Unsupervised induction and filling of semantic slots for spoken dialogue systems using frame-semantic parsing," in *Proceedings of ASRU*, 2013, pp. 120–125.

[13] William Yang Wang, Dan Bohus, Ece Kamar, and Eric Horvitz, "Crowdsourcing the acquisition of natural language corpora: Methods and observations," in *Proceedings of SLT*, 2012, pp. 73–78.

[14] Dipanjan Das, Nathan Schneider, Desai Chen, and Noah A Smith, "Probabilistic frame-semantic parsing," in *Proceedings of NAACL-HLT*, 2010, pp. 948–956.

[15] Collin F Baker, Charles J Fillmore, and John B Lowe, "The Berkeley FrameNet project," in *Proceedings of COLING*, 1998, pp. 86–90.

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

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

[18] Dilek Hakkani-Tür, Larry Heck, and Gokhan Tur, "Using a knowledge graph and query click logs for unsupervised learning of relation detection," in *Proceedings of ICASSP*, 2013, pp. 8327–8331.

[19] Lu Wang, Dilek Hakkani-Tür, and Larry Heck, "Leveraging semantic web search and browse sessions for multi-turn spoken dialog systems," in *Proceedings of ICASSP*, 2014.

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

[21] Gokhan Tur, Li Deng, Dilek Hakkani-Tür, and Xiaodong He, "Towards deeper understanding: deep convex networks for semantic utterance classification," in *Proceedings of ICASSP*, 2012, pp. 5045–5048.

[22] Li Deng, Gokhan Tur, Xiaodong He, and Dilek Hakkani-Tür, "Use of kernel deep convex networks and end-to-end learning for spoken language understanding," in *Proceedings of SLT*, 2012, pp. 210–215.

[23] Tomáš Mikolov, Wen-tau Yih, and Geoffrey Zweig, "Linguistic regularities in continuous space word representations," in *Proceedings of NAACL-HLT*, 2013, pp. 746–751.

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

[25] Charles J Fillmore, "Frame semantics and the nature of language," *Annals of the NYAS*, vol. 280, no. 1, pp. 20–32, 1976.

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

[27] Peter F Brown, Peter V Desouza, Robert L Mercer, Vincent J Della Pietra, and Jenifer C Lai, "Class-based n-gram models of natural language," *Computational linguistics*, vol. 18, no. 4, pp. 467–479, 1992.

[28] Percy Liang, *Semi-supervised learning for natural language*, Ph.D. thesis, Massachusetts Institute of Technology, 2005.

[29] Joseph Turian, Lev Ratinov, and Yoshua Bengio, "Word representations: a simple and general method for semi-supervised learning," in *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 2010, pp. 384–394.

[30] Marco Baroni, Georgiana Dinu, and Germán Kruszewski, "Dont count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors," in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, 2014, vol. 1.

[31] Tomáš Mikolov, Martin Karafiát, Lukas Burget, Jan Cernockỳ, and Sanjeev Khudanpur, "Recurrent neural network based language model," in *Proceedings of INTERSPEECH*, 2010, pp. 1045–1048.

[32] Steve Young, "CUED standard dialogue acts," Tech. Rep., Cambridge University Engineering Department, 2007.

[33] Matthew Henderson, Milica Gasic, Blaise Thomson, Pirros Tsiakoulis, Kai Yu, and Steve Young, "Discriminative spoken language understanding using word confusion networks," in *Proceedings of SLT*, 2012, pp. 176–181.

[34] Yun-Nung Chen, William Yang Wang, and Alexander I. Rudnicky, "An empirical investigation of sparse log-linear models for improved dialogue act classification," in *Proceedings of ICASSP*, 2013, pp. 8317–8321.

[35] Kendrick Boyd, Vitor Santos Costa, Jesse Davis, and C David Page, "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*. NIH Public Access, 2012, vol. 2012, p. 349.