Learning Semantic Hierarchy with Distributed Representations 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

We study the problem of unsupervised ontology learning for semantic understanding in spoken dialogue systems, in particular, learning the hierarchical semantic structure from the data. Given unlabelled conversations, we augment a frame-semantic based unsupervised slot induction approach with hierarchical agglomerative clustering to merge topically-related slots (e.g., both slots "*direction*" and "*locale*" convey location-related information) for building a coherent semantic hierarchy, and then estimate the slot importance at different levels. The high-level semantic estimation involves not only within-slot but also cross-slot relations. The experiments show that high-level semantic information can accurately estimate the prominence of slots, significantly improving the slot induction performance; furthermore, a semantic decoder trained on the data with automatically extracted slots achieves about 68% F-measure, which is close to the one from hand-crafted grammars.

Index Terms: spoken language understanding (SLU), spoken dialogue system (SDS), slot induction, hierarchical agglomerative clustering (HAC), word embeddings.

1. Introduction

Spoken dialogue systems (SDS) rely heavily on the spoken language understanding (SLU) component that structurally understands the language from dialogue participants. More specifically, the SLU component must create a mapping from the natural language inputs to the semantic representations to capture users' intentions. Traditionally, the semantic ontology for SDS is hand-crafted by domain experts or developers: this involves defining the semantic frames, slots, and possible values that situate the domain-specific conversation scenarios before building an SDS. There exists several issues: they might not generalize well to real-world users, the predefined slots can be limited or even bias the subsequent data collection and annotation, the process can be very time-consuming and have high financial costs, and the system maintenance costs are high when new conversational data comes in [1, 2, 3, 4, 5]. To overcome *generalization* and *scalability* issues, recent SLU works have focused on automatic knowledge acquisition and construction of domain-specific ontologies to reduce human effort [6, 7, 8, 9].

Unsupervised slot induction is a recently proposed task that greatly facilitates the process of constructing a semantic ontology [1]. Prior work has also considered leveraging external semantic resources, such as Freebase, and distributional semantics for unsupervised SLU [2, 10, 3]. Following the success of the above approaches, recent studies have also obtained interesting results on the tasks of relation detection [11, 12], and extending domain coverage [13, 14]. However, note that most prior work assumes a relatively flat semantic representation architecture — the cross-slot, cross-frame, and the hierarchical structure of semantic ontology are often ignored for simplicity considerations. This may not be practical enough, because in reality, dialogue systems often include slots shared by many different frames, and moreover, the cross-slot relations and multiple levels of semantic hierarchy contain important signals that are crucial to the entire SLU process [15, 16, 17]. Some recent works improved slot induction and SLU performance by modeling the inter-slot dependency relations to propagate the slot importance [7, 8]. However, the semantic hierarchy is not explicitly involved so that the topically similar slots are still treated individually, and then the learned ontology remains relatively flat.

This paper envisions a more radical *hierarchy learning* approach for *unsupervised ontology learning* to improve slot induction and SLU tasks. To do this, we show how semantic hierarchy can be learned by differentiating the concepts in the hierarchy, and how the slot importance scores estimated at various levels can be used for designing an SLU component. More specifically, we apply a hierarchical agglomerative clustering method on continuous word embeddings trained from very large external corpora to learn semantic hierarchy, and estimate high-level slot importance. To evaluate the performance of our approach, we compare the automatically induced semantic slots with the reference slots created by domain experts. Additionally, we train a semantic decoder on the data labelled with induced slots as the SLU component, and test it on unseen data to comprehensively evaluate the fully unsupervised approach.

To the best of our knowledge, we are among the first to *learn semantic hierarchy* for unsupervised ontology learning in SDSs. For slot induction, the major difference between our work and previous studies is that, instead of estimating the slot importance only based on frames in a flat architecture, we merge topically-similar slots to induce semantic hierarchy, and differentiate the slot importance at different levels. Finally, a multi-level integration strategy significantly improves the performance of slot induction and the SLU component on a real-world SDS dataset.

2. The Proposed Framework

We build our approach on top of the recent success of an unsupervised slot induction with frame-semantic parsing approach [1, 3]. Briefly, the main motivation of prior work is to 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 SDS, reducing development cost. Chen *et al.* formulated the semantic mapping and adaptation as a ranking problem, and proposed the slot importance estimation to differentiate the generic semantic concepts from the target semantic space for task-oriented dialogue systems. To consider the com-

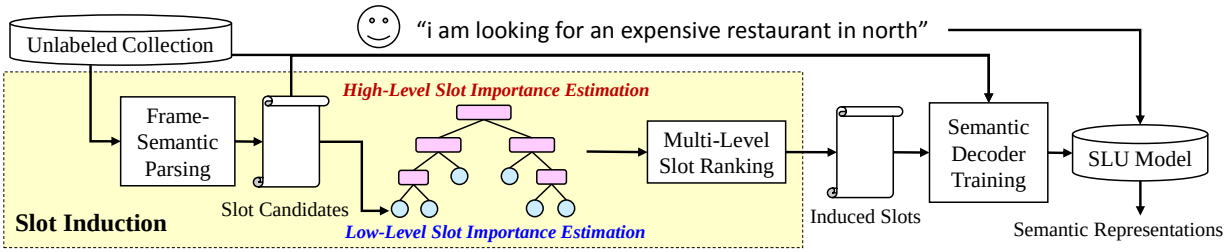


Figure 1: The proposed framework for unsupervised spoken language understanding.

plex relations among different slots that can be highly restricted by the frame-semantic parser's outputs, this paper proposes to improve the semantic adaptation process by learning the underlying semantic hierarchy from data, and integrating multi-level slot importance estimation in the hierarchical structure. The proposed framework is shown in Figure 1.

The first component performs frame-semantic parsing on ASR-decoded utterances in our unlabelled corpus, and extracts all frames from the parses as slot candidates, where the words corresponding to the frames are extracted as slot-fillers [18, 19]. FrameNet is a linguistically semantic resource that offers annotations of predicate-argument semantics, and associated lexical units [20], developed based on Frame Semantics [21]. The theory holds that the meaning of most words can be expressed on the basis of semantic frames. For example, a parsing result of an ASR-decoded utterance "*can i have a cheap restaurant*" includes three frames (capability, expensiveness, and locale_by_use) and corresponding lexical units ("*can i*", "*cheap*", and "*restaurant*").

With the list of slot candidates, slot importance estimators are performed to estimate the prominence of these slot candidates at various levels. The key slots for understanding the specific domain should be scored higher to be used in domain-specific dialogue systems. Instead of estimating the prominence of each slot based on a flat architecture, this paper focuses on a semantic hierarchy learner, namely, the low-level and high-level estimators, to share the importance across slots. Then the estimates from multiple levels can be integrated. In the end, the induced slots are used to automatically label the corpus for training an SLU component in an unsupervised manner, and then the trained model is able to decode the new utterances into the semantic representations.

3. Low-Level Slot Importance Estimation

With slot candidates from semantic parses, the model estimates their importance by integrating two scores [3]: 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 slot-fillers can help measure the prominence of the slots.

$$w^0(s) = (1-\alpha)\cdot \log f(s) + \alpha \cdot \log h(s), \qquad (1)$$

where $w^0(s)$ is the low-level importance score for the slot candidate s, $f(s)$ is the frequency of s from all parses, $h(s)$ is the coherence measure of s, and α is the weighting parameter.

For each slot s, we have a set of corresponding slot-fillers, $V(s)$, constructed from the utterances including the slot s in the parsing results. Note that $V(s)$ may include multiple same fillers. The coherence measure $h(s)$ is computed as the average pair-wise semantic similarity of slot-fillers to evaluate if slot s corresponds to centralized or scattered topics, where the semantic similarity is measured as the cosine similarity between slot-fillers' word embeddings trained on the large external data [3, 22]. The slot s with higher $h(s)$ usually focuses on fewer topics, which is more specific and more likely to be a slot for dialogue systems.

4. High-Level Slot Importance Estimation

Since the low-level slot importance estimator only considers the slot-fillers within a single slot (frame) outputted by the frame-semantic parser, the cross-slot relations are not included during the estimation. Thus, this work uses hierarchical agglomerative clustering to learn a semantic hierarchy, and utilizes high-level semantic representations to improve SLU.

4.1. Hierarchical Agglomerative Clustering

Hierarchical clustering builds nested clusters by merging or splitting them successively [23, 24, 25]. The agglomerative clustering algorithm performs hierarchical clustering using a bottom-up approach: each observation starts in its own cluster, and clusters are successively merged together based on the merge strategy. The linkage criterion determines the distance between sets of observations as a function of the pairwise distances between observations. A commonly used linkage criterion between two sets of observations A and B is average linkage:

$$\frac{1}{|A||B|}\sum_{a\in A}\sum_{b\in B} d(a,b), \qquad (2)$$

where $d(a,b)$ is the distance between observations a and b, so it allows the algorithm to minimize the average of the distances between all observations of pairs of clusters [26]. We perform the algorithm at different levels to build a hierarchical structure.

4.1.1. Word-Level Clustering

Before running the algorithm, we use all slot-fillers as seeds to expand the words based on the word representations trained on the external data. The rationale behind the word expansion includes: 1) the data collection may be too narrow in the specific domain to differentiate between domain-specific and general concepts, so word expansion based on the external data may help measure the difference; 2) the words in the evoked frames may be limited by FrameNet, so it is necessary to borrow external words to help bridge the slot-fillers for constructing a hierarchical structure. The observations for word-level clustering are word embeddings, and $d(a,b)$ in (2) is measured as the cosine distance between word embeddings of a and b.

Given the word set W after word expansion and specified number of clusters M, the clustering algorithm outputs $c(w)$ for each word $w \in W$, which is defined as the cluster label including the word w. The clustering results should group the words with similar topics together, because word embeddings based on distributional semantics contain topical information.

4.1.2. Slot-Level Clustering

The observations for slot-level clustering are slot vectors built on word-level clustering results from Section 4.1.1. Here for the slot s, we build a slot vector $\mathbf{r_s} = [r_s(1), ..., r_s(m), ..., r_s(M)]$, where M is the number of clusters, and the m-th element of $\mathbf{r}_s$ is defined as the number of words clustered into group m with normalization:

$$r_s(m) = \frac{1}{|V(s)|} \sum_{w \in V(s)} I[c(w) = m], \quad (3)$$

where $V(s)$ is the slot-filler set defined in Section 3, and $I[c(w) = m]$ is an indicator function equal to 1 if the word w is grouped into the m-th cluster, 0 otherwise. Here the constructed slot vectors embed the semantic information provided by word-level clustering results, which can be generalized to higher levels. The distance between two slots for slot-level clustering can be measured as their slot vectors' cosine similarity.

Given the slot vectors, the algorithm outputs $\hat{c}^h(s)$ as the cluster label for the slot s at the h-th level. The clustering procedure tends to merge topically-related slots, where the smaller distance between two slots means that they contain some words belonging to the same cluster but unnecessarily include the same slot-filler. For example, the slot candidate expensiveness contains the fillers "*cheap*", "*cost*", "*expensive*", and so on. Another example commerce_scenario includes fillers such as "*price*" and "*prices*". They would be merged together by the algorithm because most words from them are labelled the same by word-level clustering, even though they do not contain any fillers in common.

4.2. Bottom-Up Slot Importance Estimation

With clustered slots as the high-level semantics, we estimate the slot importance scores at different levels via a bottom-up method:

$$w^{h+1}(s) = \frac{1}{|\hat{C}^h(s)|} \sum_{\hat{c}^h(s_k) = \hat{c}^h(s)} w^h(s_k), \quad (4)$$

where $w^h(s)$ is the slot importance from the h-th level ($w^0(s)$ is estimated in (1)), $\hat{C}^h(s)$ includes all slots labelled as $\hat{c}^h(s)$ at the h-th level, so $|\hat{C}^h(s)|$ is the size of the cluster including the slot s. That is, $w^{h+1}(s)$ averages the slot importance within the cluster $\hat{c}^h(s)$ from the previous level as its high-level slot importance (at the $(h+1)$-th level), which considers not a single slot but multiple topically-related slots. Note that $w^{h+1}(s_i) = w^{h+1}(s_j)$ if $\hat{c}^h(s_i) = \hat{c}^h(s_j)$.

5. Multi-Level Slot Ranking

Considering that different slots may score differently even if they are topically related to each others, the final weight of the slot s is computed as

$$w(s) = \sum_{h=0}^{H} \lambda_h \cdot w^h(s). \quad (5)$$

Here the semantics from different levels are integrated for measuring the final prominence of the slot ($\sum_h \lambda_h = 1$). We rank the slot candidates by the final scores, and tune a threshold θ to output the induced slots in a fully unsupervised fashion. Note that the reason for integrating scores from different levels is to make the measurement more robust, since the performance of a high-level slot estimator relies on a good and accurate hierarchy. Considering that the learned hierarchy may not be perfect, combination of the estimators from different levels produces more robust results. Hence, $w^{h+1}(s_i) \neq w^{h+1}(s_j)$ even if $\hat{c}^h(s_i) = \hat{c}^h(s_j)$, because we differentiate them by involving their individual low-level importance.

6. Semantic Decoder Training

While semantic slot induction is essential for providing semantic categories and imposing semantic constraints, we are also interested in understanding the performance achieved by our induced slots. Therefore, after slot induction, we use the original corpus as the training data and the automatically extracted slots as the pseudo training labels for building a semantic decoder (a.k.a. SLU component). 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 [27, 28, 26]. We build a vector representation of an utterance as $\mathbf{u} = [x_1, ..., x_j, ...]$.

$$x_j = \mathbb{E}[C_u(n\text{-gram}_j)]^{1/|n\text{-gram}_j|}, \quad (6)$$

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 length of 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. The trained model can be used to decode the semantic representations from testing utterances.

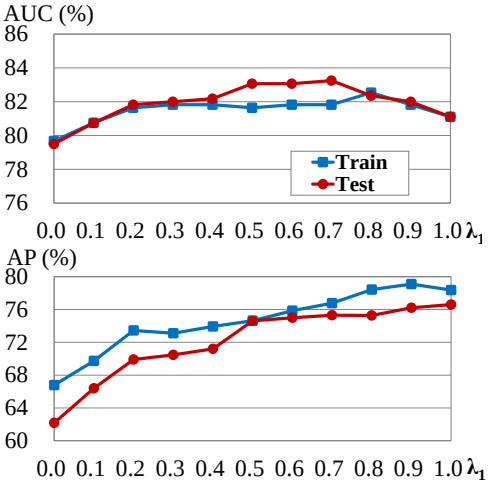
7. Experiments

To evaluate the effectiveness of our approach, we performed two experiments. First, we directly examine the slot induction performance by comparing the ranked list of frame-semantic parsing induced slots with the reference slots created by system developers [29]. Secondly, we examine the performance of the unsupervised SLU model to analyze whether induced slots can be used for practical SLU tasks.

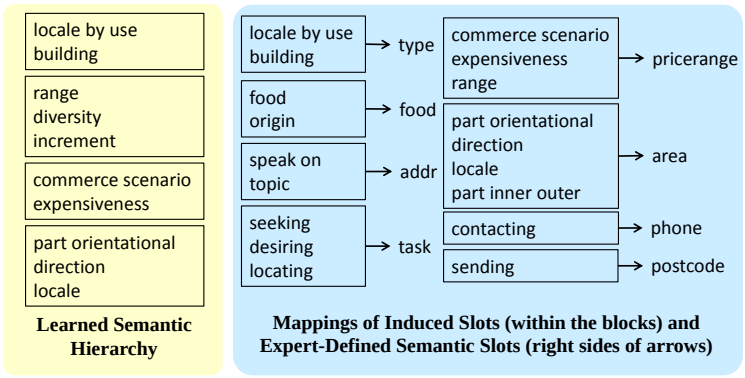
7.1. Experimental Setup

In this experiment, we used the Cambridge University SLU corpus [28, 30]. The domain of the corpus is about restaurant recommendation in Cambridge. The corpus contains a total number of 2,166 dialogues (1,522 for training; 644 for testing), and we only use 11,288 utterances with semantic tags (7,634 for training; 3,654 for testing) in the experiments. 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, pricerange, signature, task, and type. The parameters α in (1), λ_h in (5), θ for thresholding the slots, and the number of clusters M can be tuned on a development set (1/10 of training data). Note that our unsupervised approaches does not use any labelled data, and training data is for self-training. Here we set the number of semantic levels to 2 since the data has a relatively simple semantic hierarchy.

To include distributional semantics information, we use pre-trained word embeddings [22, 31]. The word vectors are trained on 10^9 words from Google News using the continuous



(a) The performance with different interpolation weights



(b) The automatically learned and the reference hierarchies

Figure 2: The performance of integration of the high-level semantic hierarchy (number of levels is set to 2)

Table 1: The performance of induced slots and SLU models(%)

Approach		Slot Induction		SLU Model
		AP	AUC	F-Measure
Baseline	Low-Level	62.19	79.50	60.27
Proposed	High-Level	**76.60**	81.28	67.94
	Multi-Level	76.21	**82.00**	**68.13**

bag of words architecture. The resulting vectors have dimensionality 300, vocabulary size is 3×10^6; the entities contain both words and automatically derived phrases.

7.2. Slot Induction Evaluation

We perform our approaches on the training data to generate a slot ranking list. To evaluate the accuracy of the induced slots, we measure their quality as the proximity between induced slots and reference slots. The right part of Figure 2(b) shows the mappings that indicate semantically related induced slots and reference slots [1]. For example, "expensiveness $\rightarrow$ price", "food $\rightarrow$ food", and "direction $\rightarrow$ 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. With the ranked list of induced slots, we can use the standard average precision (AP) and area under the ROC curve (AUC) as our metrics, where the induced slot is counted as correct when it has a mapping to a reference slot.

Table 1 shows the results, where the baseline is the result using low-level slot importance [3]. Proposed approaches include the result using only the high-level slot importance and the one using multiple levels. We find that high-level semantics significantly helps slot induction performance in terms of both AP and AUC. Furthermore, Figure 2(a) analyzes the performance balancing the low-level and high-level slot importance, λ_0 and λ_1 respectively in (5), to examine the contribution of the high-level semantics on the slot induction task. It shows that for both AP and AUC, high-level semantics improve the performance significantly. Also, the best weight for λ_1 is about $0.7 - 0.9$ in terms of both metrics, and this means that the proposed hierarchical structure can induce domain-specific slots more effectively, since it considers the cross-slot relations. In addition, we show some slot-level clustering examples on the left part of Figure 2(b) to analyze high-level semantics. Here it is obvious that some slots with the same semantics can be successfully grouped together (referring to the oracle mapping), indicating that our hierarchical structure helps the semantics estimation effectively.

7.3. SLU Model Evaluation

An unsupervised SLU model trained on the training data with pseudo labels is used to decode the semantic representations on the test data, where an SVM with linear kernel is applied to classify if an utterance contain a certain slot or not. To evaluate our SLU model, we compute a micro F-measure by comparing the automatically-decoded and reference semantic representations.

From Table 1, by comparing the results of the proposed approaches and the baseline, high-level semantics significantly improve the performance (from 60% to 68% on F-measure). The difference between the high-level and the multi-level approach is not significant, which shows that our learned semantic hierarchy is accurate enough to provide a better high-level slot estimator. Overall, the proposed high-level and multi-level approaches estimate the slot importance more accurately and outperform that only using low-level semantics; this demonstrates the effectiveness of considering a semantic hierarchy to improve slot induction for unsupervised SLU. The 77% of AP and 82% of AUC indicate that our proposed approach can generate good coverage for domain-specific slots in a real-world SDS. The 68% of F-measure is close to the performance of hand-crafted grammars (about 69% of F-measure) on the same dataset [28]. Therefore, this paper shows the feasibility of applying our approach to SDS development with lower labor cost.

8. Conclusion

This paper proposes an unsupervised approach unifying semantics from different levels for automatic slot induction and SLU modeling. Our work makes use of a state-of-the-art semantic parser, and adapts the generic FrameNet representation to a semantic space characteristic of a domain-specific SDS using a hierarchical structure. With the incorporation of high-level semantics from a learned hierarchy, we show that our automatically induced semantic slots align better with reference slots. We also show the feasibility of training an SLU component based on automatically induced slots and its promising performance.

9. References

[1] Y.-N. Chen, W. Y. Wang, and A. 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.

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

[3] Y.-N. Chen, W. Y. Wang, and A. I. Rudnicky, "Leveraging frame semantics and distributional semantics for unsupervised semantic slot induction in spoken dialogue systems," in *Proceedings of SLT*, 2014.

[4] Y.-N. Chen and A. I. Rudnicky, "Two-stage stochastic natural language generation for email synthesis by modeling sender style and topic structure," in *Proceedings of INLG*, 2014, pp. 152–156.

[5] M. Korpusik, N. Schmidt, J. Drexler, S. Cyphers, and J. Glass, "Data collection and language understanding of food descriptions," in *Proceedings of SLT*, 2014, pp. 560–565.

[6] B. Hixon, P. Clark, and H. Hajishirzi, "Learning knowledge graphs for question answering through conversational dialog," in *Proceedings of NAACL-HLT*, 2015, pp. 851–861.

[7] Y.-N. Chen, W. Y. Wang, and A. I. Rudnicky, "Jointly modeling inter-slot relations by random walk on knowledge graphs for unsupervised spoken language understanding," in *Proceedings of NAACL-HLT*, 2015, pp. 619–629.

[8] Y.-N. Chen, W. Y. Wang, A. Gershman, and A. I. Rudnicky, "Matrix factorization with knowledge graph propagation for unsupervised spoken language understanding," in *Proceedings of ACL-IJCNLP*, 2015.

[9] H. Cuayáhuitl, N. Dethlefs, H. Hastie, and X. Liu, "Training a statistical surface realiser from automatic slot labelling," in *Proceedings of SLT*, 2014, pp. 112–117.

[10] G. Tur, M. Jeong, Y.-Y. Wang, D. Hakkani-Tür, and L. P. Heck, "Exploiting the semantic web for unsupervised natural language semantic parsing," in *Proceedings of INTERSPEECH*, 2012.

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

[12] Y.-N. Chen, D. Hakkani-Tür, and G. Tur, "Deriving local relational surface forms from dependency-based entity embeddings for unsupervised spoken language understanding," in *Proceedings of SLT*, 2014.

[13] A. El-Kahky, D. Liu, R. Sarikaya, G. Tür, D. Hakkani-Tür, and L. 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.

[14] Y.-N. Chen and A. I. Rudnicky, "Dynamically supporting unexplored domains in conversational interactions by enriching semantics with neural word embeddings," in *Proceedings of SLT*, 2014.

[15] N. Camelin, B. Detienne, S. Huet, D. Quadri, and F. Lefevre, "Unsupervised concept annotation using latent dirichlet allocation and segmental methods," in *Proceedings of the First Workshop on Unsupervised Learning in NLP*, 2011, pp. 72–81.

[16] A. Lorenzo, L. Rojas-Barahona, and C. Cerisara, "Unsupervised structured semantic inference for spoken dialog reservation tasks," in *Proceedings of SIGDIAL*, 2013, pp. 12–20.

[17] K. Yoshino, S. Mori, and T. Kawahara, "Spoken dialogue system based on information extraction using similarity of predicate argument structures," in *Proceedings of SIGDIAL*, 2011, pp. 59–66.

[18] D. Das, N. Schneider, D. Chen, and N. A. Smith, "Probabilistic frame-semantic parsing," in *Proceedings of NAACL-HLT*, 2010, pp. 948–956.

[19] D. Das, D. Chen, A. F. T. Martins, N. Schneider, and N. A. Smith, "Frame-semantic parsing," *Computational Linguistics*, 2013.

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

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

[22] T. Mikolov, W.-t. Yih, and G. Zweig, "Linguistic regularities in continuous space word representations," in *Proceedings of NAACL-HLT*, 2013, pp. 746–751.

[23] M. Steinbach, G. Karypis, V. Kumar *et al.*, "A comparison of document clustering techniques," in *KDD workshop on text mining*, vol. 400, no. 1. Boston, 2000, pp. 525–526.

[24] M. L. Zepeda-Mendoza and O. Resendis-Antonio, "Hierarchical agglomerative clustering," in *Encyclopedia of Systems Biology*. Springer, 2013, pp. 886–887.

[25] D. Beeferman and A. Berger, "Agglomerative clustering of a search engine query log," in *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2000, pp. 407–416.

[26] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[27] D. Hakkani-Tür, F. Béchet, G. Riccardi, and G. Tur, "Beyond asr 1-best: Using word confusion networks in spoken language understanding," *Computer Speech & Language*, vol. 20, no. 4, pp. 495–514, 2006.

[28] M. Henderson, M. Gasic, B. Thomson, P. Tsiakoulis, K. Yu, and S. Young, "Discriminative spoken language understanding using word confusion networks," in *Proceedings of SLT*, 2012, pp. 176–181.

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

[30] Y.-N. Chen, W. Y. Wang, and A. I. Rudnicky, "An empirical investigation of sparse log-linear models for improved dialogue act classification," in *Proceedings of ICASSP*, 2013, pp. 8317–8321.

[31] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proceedings of NIPS*, 2013, pp. 3111–3119.