DERIVING LOCAL RELATIONAL SURFACE FORMS FROM DEPENDENCY-BASED ENTITY EMBEDDINGS FOR UNSUPERVISED SPOKEN LANGUAGE UNDERSTANDING

Yun-Nung Chen[⋆†] *Dilek Hakkani-Tür*[†] *Gokan Tur*[†]

[⋆]School of Computer Science, Carnegie Mellon University, PA, USA
[†]Microsoft Research, Mountain View, CA, USA
yvchen@cs.cmu.edu, {dilek,gokhan.tur}@ieee.org

ABSTRACT

Recent works showed the trend of leveraging web-scaled structured semantic knowledge resources such as Freebase for open domain spoken language understanding (SLU). Knowledge graphs provide sufficient but ambiguous relations for the same entity, which can be used as statistical background knowledge to infer possible relations for interpretation of user utterances. This paper proposes an approach to capture the relational surface forms by mapping dependency-based contexts of entities from the text domain to the spoken domain. Relational surface forms are learned from dependency-based entity embeddings, which encode the contexts of entities from dependency trees in a deep learning model. The derived surface forms carry functional dependency to the entities and convey the explicit expression of relations. The experiments demonstrate the efficiency of leveraging derived relational surface forms as local cues together with prior background knowledge.

Index Terms— spoken language understanding (SLU), relation detection, semantic knowledge graph, entity embeddings, spoken dialogue systems (SDS).

1. INTRODUCTION

Spoken language understanding (SLU) aims to detect the semantic frames that include domain-related information. Traditional spoken dialogue systems (SDS) are trained with annotated examples and support limited domains. Recently, structured semantic knowledge such as Freebase[1] [1], FrameNet[2] [2], etc. is utilized to obtain domain-related knowledge and help SLU for tackling open domain problems in SDSs [3, 4, 5, 6, 7].

Knowledge graphs, such as Freebase [1], usually carry rich information for named entities, which is encoded in triples about entity pairs and their relation. Such information is usually used for interpretation of natural language in SDSs [8, 9, 5]. However, the entity lists/gazatteers may bring noise and ambiguity to SLU, for example, the commonly used words "*Show me*" and "*Up*" can be movie names, and "*Brad Pitt*" can be an actor name or a producer name, which makes interpretation more difficult. Some works focused on assigning weights for entities or entity types to involve prior background knowledge of entities, where the probabilistic confidences offer better cues for SLU [8, 5]. Also, many works focused on mining natural language forms based on the ontology by web search or query click logs, which benefit discovering new relation types from large text corpora [10, 11]. The mined data can also be used to help SLU by adaptation from the text domain to the spoken domain [4].

[1]http://www.freebase.com
[2]http://framenet.icsi.berkeley.edu

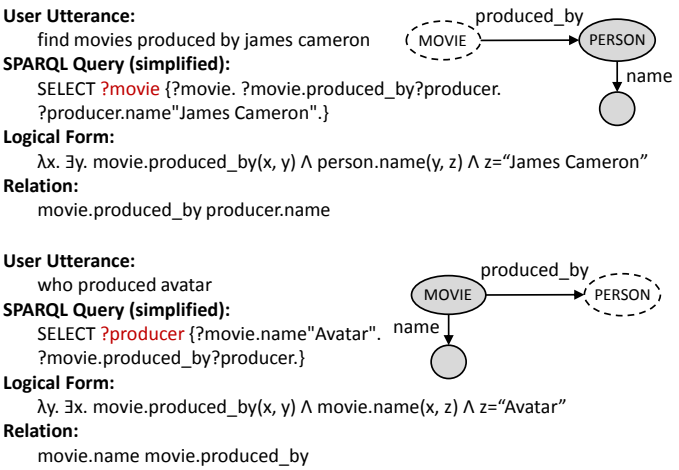

Fig. 1. The relation detection examples.

On the other hand, the distributional view of semantics hypothesizes that words occurring in the same contexts may have similar meanings, and words can be represented as high dimensional vectors. Recently, with the advancement of deep learning techniques, the continuous word representations that represent words as dense vectors have been shown to perform well in many applications [12, 13, 14, 15, 16, 17]. Furthermore, dependency-based word embeddings were proposed to capture more functional similarity based on the dependency-based contexts instead of the linear contexts using the similar training procedure [18].

Following the successes, we leverage dependency-based entity embeddings to learn relational information including entity surface forms and entity contexts from the text data. Integrating derived relational information as local cues and gazetteers as background knowledge performs well for the relation detection task in a fully unsupervised fashion.

2. KNOWLEDGE GRAPH RELATIONS

Given an utterance, we can form a set of relations that encode the user's intent for informational queries based on the semantic graph ontology. Fig. 1 presents two user utterances and their invoked relations, which can be used to create requests in query languages (i.e., SPARQL Query Language for RDF[3]). The two examples in this figure include two nodes and the same relation `movie.produced_by`, and we differentiate these examples by

[3]http://www.w3.org/TR/ref-sparql-query/

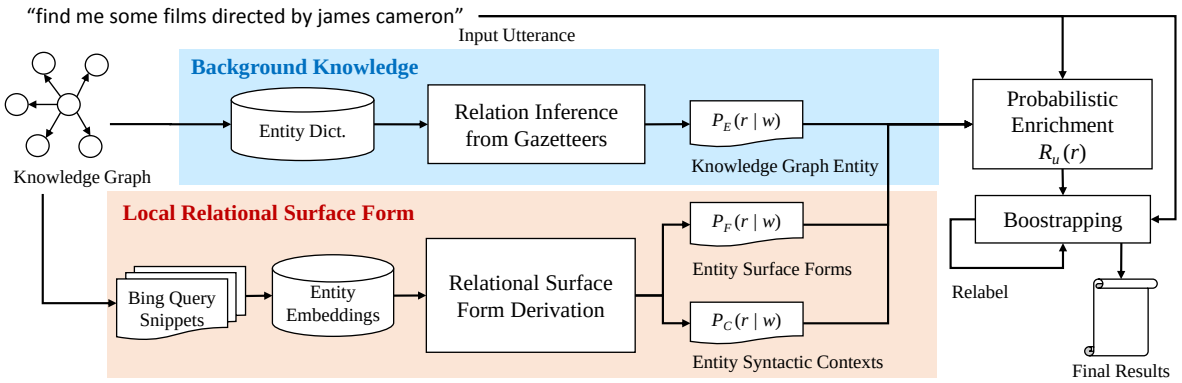

Fig. 2. The proposed framework.

including the originating node types in the relation (`movie.name`, `producer.name`) instead of just plain names (the nodes with gray color denote specified entities). Given all these, this task is richer than the regular relation detection task. The motivation to do that is, since we are trying to write queries to the knowledge source, we need to make sure that the queries are well-formed and relation arcs originate from the correct nodes in them. Therefore, this paper focuses on detecting not only `movie.produced_by` but also the specified entities `producer.name` and `movie.name`, so that we can obtain a better understanding of user utterances.

3. PROPOSED FRAMEWORK

The whole system framework is shown in Fig 2. There are two major components: 1) we first utilize background knowledge as a prior to infer relations, and 2) we capture natural language surface forms for detecting local observations, which are described in Sections 4 and Section 5 respectively.

Then probabilistic enrichment is used to integrate probabilistic information from background knowledge and local relational observations given the input utterance. Finally an unsupervised learning approach is proposed to boost the performance. The detail is presented in Section 6.

4. RELATION INFERENCE FROM GAZETTEERS

Due to ambiguity of entity mentions, we utilize prior knowledge from gazetteers to estimate the probability distribution of associated relations for each entity [5]. For example, "*James Cameron*" can be a director or a producer, which infers `movie.directed_by` or `movie.produced_by` relations respectively. Given a word w_j, the estimated probability of inferred relation r_i is defined as

$$P_E(r_i \mid w_j) = P_E(t_i \mid w_j) = \frac{C(w_j, t_i)}{\sum_{t_k \in T(w_j)} C(w_j, t_k)}, \quad (1)$$

where t_i is the type corresponding to the relation r_i (e.g. the entity type `director.name` infers the relation `movie.directed_by`), $T(w_j)$ denotes the set of all possible entity types of the word w_j, and $C(w_j, t_i)$ is the number of times the specific entity w_j is observed with a specific type t_i in the knowledge graph. For example, the number of movies James Cameron has directed.

5. RELATIONAL SURFACE FORM DERIVATION

5.1. Web Resource Mining

Based on the ontology of knowledge graph, we extract all possible entity pairs that are connected with specific relations. Following the previous work, we get search snippets for entity pairs tied with specific relations by web search[4] [3, 10]. Then we mine the patterns used in natural language realization of the relations. With the query snippets, we use dependency relations to learn natural language surface forms about each specific relation by dependency-based entity embeddings introduced below.

5.2. Dependency-Based Entity Embeddings

Most neural embeddings use linear bag-of-words contexts, where a window size k is defined to produce contexts of $2k$ words, k words before and after the target word [12, 13, 14]. 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 [18].

An example sentence "*Avatar is a 2009 American epic science fiction film directed by James Cameron.*" and its dependency parsing result are illustrated in Fig. 3. Here the sentence comes from snippets returned by searching the entity pair, "*Avatar*" (movie) and "*James Cameron*" (director). The arrows denote the dependency relations from headwords to their dependents, and words on arcs denote type of the dependency relations. Relations that include a preposition are "collapsed" prior to context extraction (dashed arcs in Fig. 3), by directly connecting the head and the object of the preposition, and subsuming the preposition itself into the dependency label. Before training embeddings, we replace entities with their entity tags such as *$movie* for "*Avatar*" and *$director* for "*James Cameron*".

The dependency-based contexts extracted from the example are given in Table 1, 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 entity embeddings for all target words [15, 16, 17].

For training dependency-based entity embeddings, each word w is associated with a word vector $v_w \in \mathbb{R}^d$ and each context c is

[4] `http://www.bing.com`

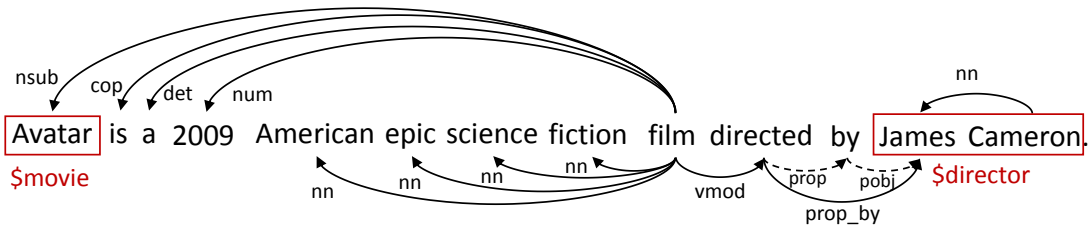

Fig. 3. An example of dependency-based contexts.

Table 1. The contexts extracted for training dependency entity embeddings in the example of the Fig. 3.

Word	Contexts
$movie	film/nsub^{-1}
is	film/cop^{-1}
a	film/det^{-1}
2009	film/num^{-1}
american, epic, science, fiction	film/nn^{-1}
film	avatar/nsub, is/cop, a/det, 2009/num, american/nn, epic/nn, science/nn, fiction/nn, directed/vmod
directed	$director/prep_by
$diretor	directed/prep_by^{-1}

represented as a context vector $v_c \in \mathbb{R}^d$, where d is the embedding dimensionality. We learn vector representations for both words and contexts such that the dot product $v_w \cdot v_c$ associated with "good" word-context pairs belonging to the training data D is maximized, leading to the objective function:

$$\arg\max_{v_w, v_c} \sum_{(w,c)\in D} \log \frac{1}{1+\exp(-v_c \cdot v_w)}, \tag{2}$$

which can be trained using stochastic-gradient updates [18].

5.3. Surface Form Derivation

In addition to named entities detected by gazetteers, there are two different relational surface forms used in natural languages, entity surface forms and entity syntactic contexts, which are derived from trained embeddings by following approaches.

5.3.1. Entity Surface Forms

With only background knowledge gazetteers provided in Section 4, the unspecified entities cannot be captured because knowledge graph does not contain such information like the words "*film*" and "*director*". This procedure is to discover the words that play the same role and carry similar functional dependency as the specified entities. For example, the entity *$character* may derive the word "*role*", and *$movie* may derive "*film*", "*movie*" as their entity surface forms. The unspecified entities provide important cues for inferring corresponding relations.

We first define a set of entity tags $E = \{e_i\}$ and a set of words $W = \{w_j\}$. Based on the trained dependency-based entity embeddings, for each entity tag e_i, we compute the score of the word w_j as

$$S_i^F(w_j) = \frac{\text{FormSim}(w_j, e_i)}{\sum_{e_k \in E} \text{FormSim}(w_j, e_k)}, \tag{3}$$

where $\text{FormSim}(w, e)$ is the cosine similarity between the embeddings of the word w and the entity tag e. $S_i^F(w_j)$ can be viewed as the normalized weights of the words indicating the importance for discriminating different entities. Based on $S_i^F(w_j)$, we propose to extract top N similar words for each entity tag e_i, to form a set of entity surface forms F_i, where F_i includes surface form candidates of entity e_i. The derived words may have similar embeddings as the target entity, for example, "*director*" and *$director* may encode the same context information such as *directed/prep_by*$^{-1}$ in their embeddings. Therefore, the word "*director*" can be extracted by the entity tag *$director* to serve its surface form. With derived words F_i for entity tag e_i, we can normalize the relation probabilities the word $w_j \in F_i$ infers.

$$P_F(r_i \mid w_j) = P_F(e_i \mid w_j) = \frac{S_i^F(w_j)}{\sum_{k, w_j \in F_k} S_k^F(w_j)}, \tag{4}$$

where r_i is the relation inferred from the entity tag e_i, $S_k^F(w_j)$ is the score of the word w_j that belongs to the set F_k extracted by the entity tag e_k, and $P_F(r_i \mid w_j)$ is similar to $P_E(r_i \mid w_j)$ in (1) but based on derived words instead of specified entities.

5.3.2. Entity Syntactic Contexts

Another type of relational cues comes from contexts of entities; for example, a user utterance "*find movies produced by james cameron*" includes an unspecified movie entity "*movies*" and a specified entity "*james cameron*", which may be captured by entity surface forms via P_F and gazetteers via P_E respectively. However, it doesn't consider local observations "*produced by*". In this example, the most likely relation of the entity "*james cameron*" from the background knowledge is `director.name`, which infers `movie.directed_by`, and the local observations are not be used to derive the correct relation `movie.produced_by` for this utterance.

This procedure is to discover the relational entity contexts based on syntactic dependency. With dependency-based entity embeddings and their context embeddings, for each entity tag e_i, we extract top N syntactic contexts to form a set of entity contexts C_i, which includes the words that are the most activated by a given entity tag e_i. The extraction procedure is similar to one in Section 5.3.1; for each entity tag e_i, we compute the score of the word w_j as

$$S_i^C(w_j) = \frac{\text{CxtSim}(w_j, e_i)}{\sum_{e_k \in E} \text{CxtSim}(w_j, e_k)}, \tag{5}$$

where $\text{CxtSim}(w_j, e_i)$ is the cosine similarity between the context embeddings of the word w_j and the embeddings of the entity tag e_i.

The derived contexts may serve the indicators of possible relations. For instance, for the entity tag *$producer*, the most activated contexts include "*produced/prep_by*$^{-1}$", so the word "*produced*" can be extracted by this procedure for detecting local observations other than entities. Then we can normalize the relation probabilities the

Table 2. An example of three different methods in probabilistic enrichment ($w =$ "*pitt*").

r	actor	produced_by	location
$P_E(r \mid w)$	0.7	0.3	0
$P_F(r \mid w)$	0.4	0	0.6
$P_C(r \mid w)$	0	0	0
Unweighted $R_w(r)$	1	1	1
Weighted $R_w(r)$	0.7	0.3	0.6
Highest Weighted $R_w(r)$	0.7	0	0.6

contexts imply to compute $P_C(r_i \mid w_j)$ similar to (4):

$$P_C(r_i \mid w_j) = P_C(e_i \mid w_j) = \frac{S_i^C(w_j)}{\sum_{k, w_j \in C_k} S_k^C(w_j)}. \quad (6)$$

6. PROBABILISTIC ENRICHMENT AND BOOTSTRAPPING

Hakkani-Tür *et al.* proposed to use probabilistic weights for unsupervised relation detection [5]. We extend the approach to integrate induced relations from prior knowledge $P_E(r_i \mid w_j)$ and from local relational surface forms $P_F(r_i \mid w_j)$ and $P_C(r_i \mid w_j)$ to enrich the relation weights for effectively detecting relations given the utterances. This paper experiments to integrate multiple distributions in three ways:

- Unweighted

$$R_w(r_i) = \begin{cases} 1 & \text{, if } P_E(r_i \mid w) > 0 \text{ or} \\ & P_F(r_i \mid w) > 0 \text{ or} \\ & P_C(r_i \mid w) > 0. \\ 0 & \text{, otherwise.} \end{cases} \quad (7)$$

  This method combines possible relations from all sources, which tends to capture as many as possible relations (higher recall).

- Weighted

$$R_w(r_i) = \max(P_E(r_i \mid w), P_F(r_i \mid w), P_C(r_i \mid w)) \quad (8)$$

  This method assumes that the relation r_i invoked in word w comes from the source that carries the highest probability, so it simply selects the highest one among the three sources.

- Highest Weighted

$$R_w(r_i) = \max(P'_E(r_i \mid w), P'_F(r_i \mid w), P'_C(r_i \mid w)),$$

$$P'(r_i \mid w) = \mathbb{1}[i = \arg\max_i P(r_i \mid w)] \cdot P(r_i \mid w). \quad (9)$$

  This method only combines the most likely relation for each word, because $P'(r_i \mid w) = 0$ when the relation r_i is not the most likely relation of the word w.

An example of relation weights about the word "*pitt*" with three different methods is shown in Table 2. The final relation weight of the relation r_i given an utterance u, $R_u(r_i)$, can be compute as

$$R_u(r_i) = \max_{w \in u} R_w(r_i). \quad (10)$$

With enriched relation weights, $R_u(r_i)$, we train a multi-class, multi-label classifier in an unsupervised way, where we learn ensemble of weak classifiers by creating pseudo training labels in each iteration for boosting the performance [10, 19, 20, 21]. The detail of the algorithm is shown in Algorithm 1. Then the returned classifier E^k can be used to detect relations given unseen utterances.

Algorithm 1 Bootstrapping

Input: the set of user utterances $U = \{u_j\}$; the relation weights for the utterances, $R_{u_j}(r_i)$, $u_j \in U$;

Output: the multi-class multi-label classifier E that estimates relations given an utterance

1: Initializing relation labels $L^0(u_j) = \{r_i \mid R_{u_j}(r_i) \geq \delta\}$;
2: **repeat**
3: Training ensemble of M weak classifiers E^k on U and $L^k(u_j)$;
4: Classifying the utterance u_j by E^k and output probability distribution of relations as $R_{u_i}^{(k+1)}(r_i)$;
5: Creating relation labels $L^{(k+1)}(u_j) = \{r_i \mid R_{u_j}^{(k+1)}(r_i) \geq \delta\}$;
6: **until** $L^{(k+1)}(u_j) \sim L^k(u_j)$
7: **return** E^k;

Table 3. Relation detection datasets used in the experiments.

Query Statistics	Train	Test
% entity only	8.9%	10.7%
% rel only with specified movie names	27.1%	27.5%
% rel only with specified other names	39.8%	39.6%
% more complicated relations	15.4%	14.7%
% not covered	8.8%	7.6%
#utterance with SPARQL annotations	3338	1084

7. EXPERIMENTS

7.1. Dataset

The experiments use a list of entities/gazetteers from the publicly available Freebase knowledge graph. The list includes 670K entities of 78 entity types, including movie names, actors, release dates, etc. after filtering out the movie entities with lower confidences [8].

The relation detection datasets include crowd-sourced utterances addressed to a conversational agent and are described in Table 3. Both train and test sets are manually annotated with SPARQL queries, which are used to extract relation annotations. Most of data includes the relations with either specified movie names or specified other names. In addition, the relations only with specified movie names are difficult to capture by gazetteers, which emphasizes the contribution of this task. We use 1/10 training data as a development set to tune the parameters δ, M, and the optimal number of iterations in Algorithm 1. The training set is only used to train the classifier of Algorithm 1 for bootstrapping in an unsupervised way; note that the manual annotations are not used here.

For retrieving the snippets, we use 14 entity pairs from the knowledge graph related to movie entities, which include director, character, release date, etc. We extract snippets related to each pair from web search results, and we end up with 80K snippets, where the pairs of entities are marked in the returned snippets[5]. For all query snippets, we parse all with the Berkeley Parser [22], and then convert the output parse trees to dependency parses using the LTH Constituency-to-Dependency Conversion toolkit[6] for training dependency-based entity embeddings [23]. The trained entity embeddings have dimension 200 and vocabulary size is 1.8×10^5.

[5] In this work, we use top 10 results from Bing for each entity pair.

[6] http://nlp.cs.lth.se/software/treebank_converter

Table 4. The relation detection performance of all proposed approaches ($N = 15$).

Micro F-measure (%)		Unweighted		Weighted		Highest Weighted	
		Ori.	Bootstrap	Ori.	Bootstrap	Ori.	Bootstrap
Baseline	(a) Gazetteer	35.21	36.91	37.93	40.10	36.08	38.89
	(b) Gazetteer + Weakly Supervised	25.07	37.39	39.04	39.07	39.40	39.98
Bag-of-Words	(c) Gazetteer + Entity Surface Form	34.23	34.91	36.57	38.13	34.69	37.16
Dep.-Based	(d) Gazetteer + Entity Surface Form	37.44	38.37	**41.01**	41.10	39.19	42.74
	(e) Gazetteer + Entity Context	35.31	37.23	38.04	38.88	37.25	38.04
	(f) Gazetteer + Entity Surface Form + Context	**37.66**	**38.64**	40.29	**41.98**	**40.07**	**43.34**

Table 5. The examples of derived entity surface forms based on dependency-based entity embeddings.

Entity Tag	Derived Word
$character	character, role, who, girl, she, he, officer
$director	director, dir, filmmaker
$genre	comedy, drama, fantasy, cartoon, horror, sci
$language	language, spanish, english, german
$producer	producer, filmmaker, screenwriter

7.2. Results

In the experiments, we train multi-class, multi-label classifiers using icsiboost [24], a boosting-based classifier, where we extract word unigrams, bigrams, and trigrams as classification features. The evaluation metric we use is micro F-measure for relation detection [5]. The performance with all of the proposed approaches with $N = 15$ (top 15 similar words of each tag) is shown in Table 4.

The first baseline here (row (a)) uses gazetteers to detect entities and then infers the relations by background knowledge described in Section 4. Row (b) is another baseline [10], which uses the retrieved snippets and their inferred relations as labels to train a multi-class multi-label classifier, then outputs the relation probabilities for each utterance as $R_u(w)$, and integrates with the first baseline. Here the data for training only uses the patterns between entity pairs in the paths of dependency trees. Row (c) is the results of adding entity surface forms derived from original embeddings, which is shown for demonstrating the effectiveness of dependency-based entity embeddings (row (d)). Row (e) is the results of adding entity contexts, and row (f) combines both of entity surface forms and entity contexts. Below we analyze the effectiveness of proposed approaches.

7.2.1. *Effectiveness of Entity Surface Forms*

The row (c) and row (d) show the performance of using entity surface forms derived from bag-of-words and dependency-based embeddings respectively. It can be found that the words derived from original embeddings do not successfully capture the surface forms of entity tags, and the results cannot be improved. On the other hand, the results from dependency-based embeddings outperform the baselines for all enrichment methods, which demonstrate the effectiveness of including entity surface forms based on dependency relations for relation detection. To analyze results of entity surface forms, we show some examples about derived words in Table 5. It can be shown that the functional similarity carried by dependency-based entity embeddings effectively benefits relation detection task.

7.2.2. *Effectiveness of Entity Contexts*

Row (e) shows the results of adding entity contexts learned from dependency-based contexts. It does not show improvement compared to baselines. Nevertheless, combining with dependency-based entity surface forms, the F-measure achieves 43% by highest weighted probabilistic enrichment, which implies that including local observations based on syntactic contexts may help relation detection, but the influence is not significant.

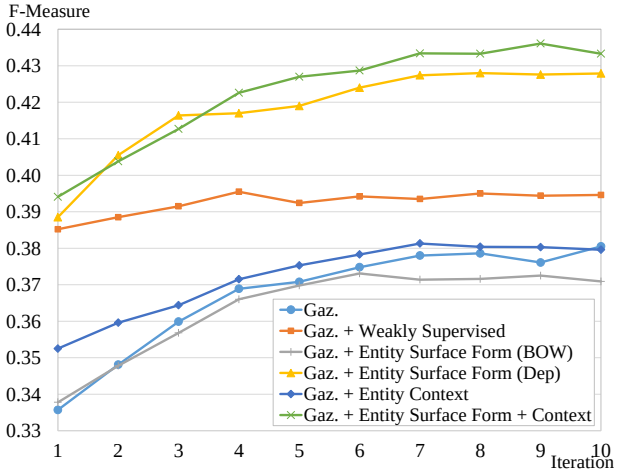

Fig. 4. Learning curves over incremental iterations of bootstrapping.

7.2.3. *Comparison of Probabilistic Enrichment Methods*

From Table 4, among the three probabilistic enrichment methods, unweighted method performs worst, because it does not differentiate the relations with higher and lower confidence, and some relations with lower probabilities will be mistakenly outputted. Comparing between weighted and highest weighted methods, the first baseline using the weighted method performs better, while other approaches using the highest weighted method perform better. The reason probably is that weighted method can provide more possible relations for the baseline only using gazetteers to increase the recall, so the weighted method benefits the first baseline. On the other hand, proposed approaches have higher recall and the highest weighted method provides more precise relations, resulting in better performance when applying the highest weighted method.

7.2.4. *Effectiveness of Bootstrapping*

The F-measure learning curves of all results using highest weighted probabilistic enrichment on the test set are presented in Fig. 4. The light blue line marked with circles is the first baseline, which applies only gazetteers with probability distribution of entity types to relation detection. After bootstrapping, the performance is significantly improved and achieves about 39% of F-measure. Another baseline using a weakly supervised classifier (orange line marked with squares) performs well before bootstrapping, while the performance

cannot be significantly improved with increased iterations. All other results show significant improvements after bootstrapping. The best result is the combination of all approaches (green line marked with crosses), and the curve shows the effectiveness and efficiency of bootstrapping. The reason probably is that the probabilities came from different sources can complement each other, and then benefit the classifiers. Also, only adding dependency-based entity surface forms (yellow line marked with triangles) performs similar to the combination result, showing that the major improvement comes from relational entity surface forms. The figure demonstrates the effectiveness of bootstrapping for improving relation detection.

7.2.5. Overall Results

The proposed approaches successfully capture local information other than background knowledge, where the relational surface forms can be learned by dependency-based entity embeddings trained on query snippets. After combining with prior relations induced by gazetteers, the relational information from the text domain can benefit the relation detection for the spoken domain. Also, the fully unsupervised approach shows the effectiveness of applying structured knowledge to SLU for tackling open domain problems.

8. CONCLUSION

This paper proposes an unsupervised approach to capture the relational surface forms including entity surface forms and entity contexts based on dependency-based entity embeddings. The detected relations viewed as local observations can be integrated with background knowledge by probabilistic enrichment methods. Experiments show that involving derived entity surface forms as local cues together with prior knowledge can significantly improve the relation detection task and help open domain SLU.

9. REFERENCES

[1] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor, "Freebase: a collaboratively created graph database for structuring human knowledge," in *Proceedings of International Conference on Management of Data*, 2008.

[2] Collin Baker, "Framenet, present and future," *Programme Committee 7*, 2008.

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

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

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

[6] 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.

[7] Yun-Nung Chen, William Yang Wang, and Alexander I. Rudnicky, "Leveraging frame semantics and distributional semantics for unsupervised semantic slot induction in spoken dialogue systems," in *Proceedings of SLT*, 2014.

[8] Dustin Hillard, Asli Celikyilmaz, Dilek Z Hakkani-Tür, and Gokhan Tur, "Learning weighted entity lists from web click logs for spoken language understanding.," in *Proceedings of INTERSPEECH*, 2011.

[9] Gokhan Tur, Dilek Z Hakkani-Tür, Dustin Hillard, and Asli Celikyilmaz, "Towards unsupervised spoken language understanding: Exploiting query click logs for slot filling.," in *Proceedings of INTERSPEECH*, 2011.

[10] 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.

[11] Dilek Hakkani-Tür, Asli Celikyilmaz, Larry P Heck, and Gokhan Tur, "A weakly-supervised approach for discovering new user intents from search query logs.," in *Proceedings of INTERSPEECH*, 2013.

[12] Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig, "Linguistic regularities in continuous space word representations.," in *Proceedings of HLT-NAACL*, 2013.

[13] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean, "Distributed representations of words and phrases and their compositionality," in *Proceedings of Advances in Neural Information Processing Systems*, 2013.

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

[15] Wen-tau Yih, Xiaodong He, and Christopher Meek, "Semantic parsing for single-relation question answering," in *Proceedings of ACL*, 2014.

[16] Antoine Bordes, Jason Weston, Ronan Collobert, Yoshua Bengio, et al., "Learning structured embeddings of knowledge bases.," in *Proceedings of AAAI*, 2011.

[17] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko, "Translating embeddings for modeling multi-relational data," in *Proceedings of NIPS*, 2013.

[18] Omer Levy and Yoav Goldberg, "Dependency-based word embeddings," in *Proceedings of ACL*, 2014.

[19] Renato De Mori, Frédéric Bechet, Dilek Hakkani-Tür, Michael McTear, Giuseppe Riccardi, and Gokhan Tur, "Spoken language understanding," *Signal Processing Magazine, IEEE*, vol. 25, no. 3, 2008.

[20] Fabrice Lefevre, François Mairesse, and Steve Young, "Cross-lingual spoken language understanding from unaligned data using discriminative classification models and machine translation.," in *INTERSPEECH*, 2010.

[21] Gokhan Tur, "Multitask learning for spoken language understanding," in *Proceedings of ICASSP*, 2006.

[22] Slav Petrov and Dan Klein, "Learning and inference for hierarchically split pcfgs," in *Proceedings of the National Conference on Artificial Intelligence*, 2007.

[23] Richard Johansson and Pierre Nugues, "Extended constituent-to-dependency conversion for english," in *16th Nordic Conference of Computational Linguistics*. University of Tartu, 2007.

[24] Benoit Favre, Dilek Hakkani-Tür, and Sebastien Cuendet, "Icsiboost," `http://code.google.come/p/icsiboost`, 2007.