Semi-Supervised Lexicon Learning for Wide-Coverage Semantic Parsing

Bo Chen[†‡], **Bo An**[†‡], **Le Sun**[†], **Xianpei Han**[†]
[†]State Key Laboratory of Computer Science
Institute of Software, Chinese Academy of Sciences, Beijing, China
[‡]University of Chinese Academy of Sciences, Beijing, China
{chenbo,anbo,sunle,xianpei}@iscas.ac.cn

Abstract

Semantic parsers critically rely on accurate and high-coverage lexicons. However, traditional semantic parsers usually utilize annotated logical forms to learn the lexicon, which often suffer from the lexicon coverage problem. In this paper, we propose a graph-based semi-supervised learning framework that makes use of large text corpora and lexical resources. This framework first constructs a graph with a phrase similarity model learned by utilizing many text corpora and lexical resources. Next, graph propagation algorithm identifies the label distribution of unlabeled phrases from labeled ones. We evaluate our approach on two benchmarks: WEBQUESTIONS and FREE917. The results show that, in both datasets, our method achieves substantial improvement when comparing to the base system that does not utilize the learned lexicon, and gains competitive results when comparing to state-of-the-art systems.

Title and Abstract in Chinese

基于半监督词典学习的语义解析技术研究

语义解析器的性能往往依赖于词典的准确度和覆盖度。传统语义解析器利用标注好的句子-逻辑表达式对来学习词典，这通常会面临词典覆盖度不足的问题。本文提出了一种基于图的半监督学习框架，该框架能够充分利用容易获取的 大量文本语料和词典资源来进行词典扩充学习。该词典扩充学习方法首先利用大量文本语料和词典资源来学习词语与词语之间的相似度，并构建用于图传播的图；接着使用图传播算法从少量标注的词汇中学习新的词汇。本文 在两个公开数据集上进行了实验，实验结果表明：本文系统相比未使用新词汇的基准系统取得了显著提升，相比当前 最好的系统，也取得了具有竞争力的结果。

1 Introduction

Semantic parsing aims to map natural language sentences into formal meaning representations, e.g., Figure 1 shows an example of semantic parsing. Semantic parsing plays an important role in natural language understanding, and has attracted increasing attention in recent years (Zelle and Mooney, 1996; Wong and Mooney, 2007; Lu et al., 2008; Liang et al., 2011; Kwiatkowski et al., 2011; Artzi and Zettlemoyer, 2013; Krishnamurthy and Mitchell, 2014; Li et al., 2015; Chen et al., 2016; Xiao et al., 2016; Jia and Liang, 2016; Reddy et al., 2016; Liang et al., 2017).

The performance of semantic parsers critically depends on the quality of lexicon, including accuracy and coverage. Specifically, in order to construct the logical form from a sentence, we first need to learn a lexicon,[1] which contains the mappings from natural language phrases (e.g., "*born*") to logical predicates (e.g., `PlaceOfBirth`). From the example in Figure 1, we can see that lexicon is the foundation of parsing, and lexicon learning plays an important role in semantic parsing.

Traditional semantic parsers are usually domain-specific, which only contains a limited number of logical predicates (Zettlemoyer and Collins, 2005; Kwiatkowksi et al., 2010; Artzi et al., 2014). In this

[1]We follow the lexicon style defined in Berant et al. (2013).

Proceedings of the 27th International Conference on Computational Linguistics, pages 892–904
Santa Fe, New Mexico, USA, August 20-26, 2018.

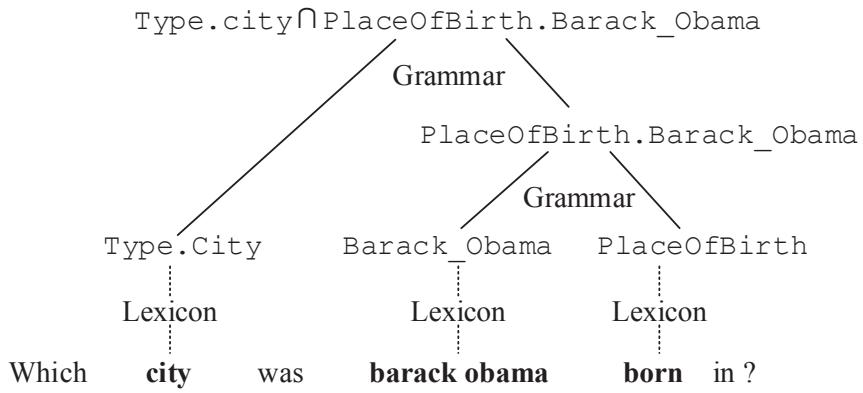

Figure 1: An example of semantic parsing, which uses lexicons to map phrases to predicates, and applies grammars to construct the logical form.

case, the mappings for each predicate can be learned relatively easily from training corpus. Recently, a growing body of research has scaled up semantic parsers to open domain (Cai and Yates, 2013a; Cai and Yates, 2013b; Berant et al., 2013; Krishnamurthy and Mitchell, 2012; Kwiatkowski et al., 2013) , where the number of predicates has increased substantially, making it hard to learn a lexicon with high coverage.

To resolve the lexicon coverage problem, there have been several papers on lexicon learning for semantic parsing. Cai and Yates (2013a) learns lexicons by pattern matching. Berant et al. (2013) learns lexicons by aligning Freebase [2] predicates with relations from ClueWeb [3], and then the alignments are used as lexicons. However, the lexicon coverage of these alignment-based methods highly depends on entity co-occurrences, and they mostly can only learn predicates which indicating relations between entities. It is still hard to cover all expressions and all predicates using alignment-based methods.

In this paper, we propose a semi-supervised lexicon learning algorithm for semantic parsing, which can increase the lexicon coverage by exploiting easily obtained text corpora and lexical resources. [4] The intuition behind our approach is that similar phrases should map to similar predicates, thus the phrase similarity can be used to propagate known predicate mappings to unknown mappings. For example, assuming we have a seed mapping: "*currency*" :: `currency`, and we know "*money*" is strongly related to "*currency*", we then can predict "*money*" should also map to `currency`. To achieve the above goal,we employ a graph-based semi-supervised learning framework, which learns lexicons not only used the alignments between Freebase and text, but also the semantic relatedness between phrases in the text side. Specifically, we use the abundant lexical resources for high coverage lexicon learning (Figure 2 shows the difference). There are three main tasks in this process. (1) we need a seed lexicon; (2) we need to measure the similarity between words; (3) we need to smooth the mappings to unlabeled words. For the similarity measure between words, we learn them from large text resources. This similarity plays two roles in our lexicon learning: (1) it is used for label propagation; (2) the similarity is used as a constraint on smoothing. That is, since we assume similar words will map to similar labels, the similarity can then strengthen the correct mapping and weaken the wrong mapping. Once we have seed lexicons and similarities between words, we smooth the lexicon by using a graph-based semi-supervised learning framework.

[2]https://en.wikipedia.org/wiki/Freebase

[3]https://www.lemurproject.org/clueweb12.php/

[4]We use *text corpora* to refer text resources from web, e.g., Wikipedia and WikiAnswers, and *lexical resources* to refer organized resources related to lexical items, e.g., WordNet and PPDB.

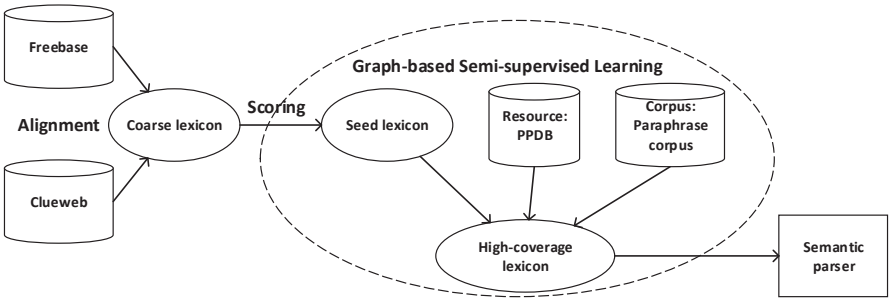

Figure 3: The framework of lexicon learning for semantic parsing in this paper. We can utilize large amount of text corpora and lexical resources to extend the lexicon for wide-open semantic parsing.

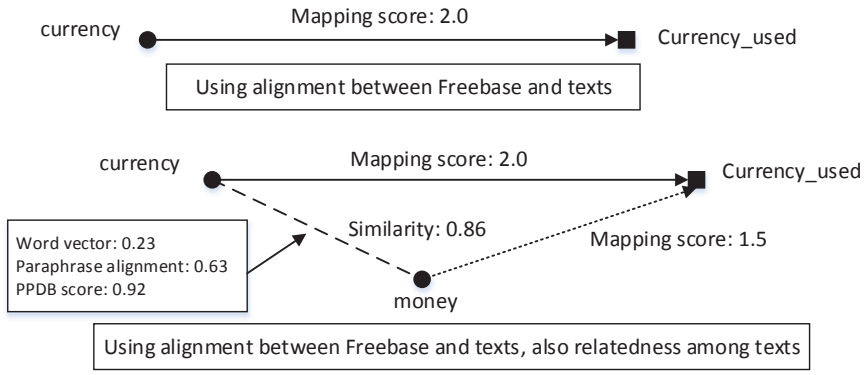

Figure 2: Example of our approach (the below one) and the previous approach (the above one). The previous approach only utilizes the alignments between text corpora and Freebase. By contrast, our approach further makes use of alignments between text and text. In this way, we can learn wide-coverage lexicon from several labeled lexicon.

The framework of our approach is shown in Figure 3. First, we make use of Freebase and ClueWeb to gain a coarse lexicon, and then we score these lexicons to gain the seed lexicon for following smooth. Next, we utilize easily obtained text corpora and text resources to learn the wide-coverage lexicon in a graph-based semi-supervised learning framework. Finally, we use the extended lexicon in the semantic parser to evaluate our approach.

We evaluate our lexicon learning algorithm on two benchmark datasets: WEBQUESTIONS and FREE917. The results show that our method can learn lexicon with higher coverage, and enhance the performance of semantic parsing system, especially its recall.

The contributions of this paper can be summarized as follows:

1. We propose a new semi-supervised learning framework for wide-coverage lexicon learning. Different to previous work, our approach can improve the lexicon coverage by further exploiting the easily obtained text corpora and lexical resources.

2. We design a graph-based learning algorithm to learn a wide-coverage lexicon from a seed lexicon.

3. We evaluate our approach on two benchmark datasets. Our system outperforms baseline systems significantly, and achieves competitive results with state-of-the-art systems.

2 Related Work

Lexicon learning is fundamental for semantic parsing. Traditional semantic parsers usually utilize annotated logical forms to learn the lexicon (Zettlemoyer and Collins, 2005; Kwiatkowksi et al., 2010; Kwiatkowski et al., 2011; Krishnamurthy and Mitchell, 2012; Berant et al., 2013; Krishnamurthy, 2016). Zettlemoyer and Collins (2005) utilizes the alignment between phrases in sentences and predicates in

annotated logical forms, and then assigned a confidence score to each lexical entry. Obviously, these approaches are limited by the annotated data.

Recently, many researchers begin to scale up semantic parsers to open domain. Learning high coverage lexicon in open domain requires large amount of annotated data, which is quite expensive even they only use question-answer pairs for supervision. There are several papers that focus on extending the lexicon for open-domain semantic parsing.

Cai and Yates (2013a) first extend the semantic parser to open domains. They utilize pattern matching to extend the lexicon. Specifically, they define knowledge patterns from knowledge bases, and text patterns from a search engine. Then they learn the lexicon based on the assumption that the phrase between two entities may map to the predicate if these two entities are also found under the predicate in knowledge bases. Berant et al. (2013) learn the lexicon by similar idea, but they use annotated ClueWeb corpus as the text side. Besides, they propose what they call a bridge operation, which is in fact a type-shifting which can bring in a predicate using minimum information. Compared to these approaches, our approach not only utilizes the alignments between knowledge bases and text corpora, but also makes use of text corpora and text resources to get the phrase similarity and phrase co-occurrence. In this way, we can learn more lexicon from little seed lexicon. Krishnamurthy (2016) also learned a lexicon for semantic parsing. However, they aim to extend the predicate side as they think the predicates have limited coverage for new sentences. Our aim is to extend the phrase that can trigger the predicates.

Graph-based semi-supervised learning algorithm has been used to resolve the OOV problem in machine translation (Razmara et al., 2013; Saluja et al., 2014; Zhao et al., 2015; Mehdizadeh Seraj et al., 2015). frame semantic parsing (Das and Smith, 2011), sentiment lexicon induction (Hamilton et al., 2016), and morph-syntactic lexicon induction (Faruqui et al., 2016).

3 Graph-based Lexicon Induction

Lexicon learning aims to learn the mapping from natural language phrases to predicates in knowledge base. There are three types of lexicons, including entity lexicon (e.g., "*city*" :: `Type.City`), unary lexicon (e.g., "*barack obama*" :: `Barack_Obama`) and binary lexicon (e.g., "*born*" :: `PlaceOfBirth`). In most cases entity lexicons are learned using entity linking techniques, therefore we usually only consider unary and binary lexicons in lexicon learning.

In open-domain semantic parsing, it is hard to learn high-coverage lexicon from annotated data for lexicon learning. In this paper, we use the (phrase, predicate) mappings as seeds, then learn new (phrase, predicate) mappings by propagating mapping information through similarities between phrases. Specifically, we propose a graph-based semi-supervised approach to resolve this problem. Our approach makes use of easily obtained text corpora and lexical resources to learn a similarity between words,[5] and then use a graph-based semi-supervised learning framework to smooth the lexicon graph. In this way, we can learn a new lexicon from labeled ones. Our method consists of three main steps:

1. Construct seed lexicon using alignments between Freebase and text corpora.
2. Learn similarities between words using both text corpora and text resources.
3. Learn new lexicon using label propagation.

3.1 Seed Lexicon Construction

We propagate information from seed lexicon to unknown ones. However, as we do not have enough annotated logical forms to learn the lexicon, and the number of predicates is too large, It is impossible to hand-make the lexicon, To obtain high-quality phrases mappings for each predicate, we construct the seed lexicon in two steps.

First, we gain a coarse lexicon by aligning Freebase with text corpora, using the lexicon learning methods as the same as Berant et al. (2013), with a slight difference that we only use unigram. We do this for two reasons: one is that a word (e.g., `born`) in a phrase (e.g., `born in`) can trigger a

[5]We use *phrase* and *word* alternately in this paper, since we use unigram for the lexicon and a *phrase* contains a single word.

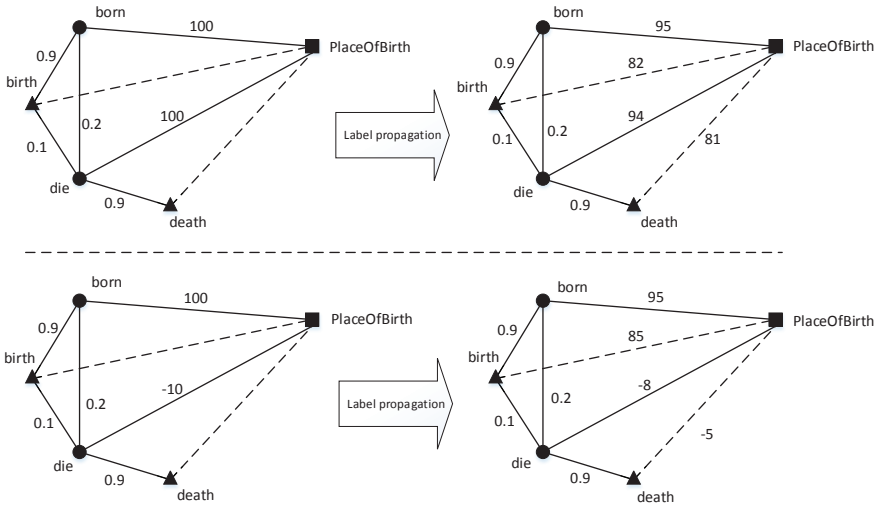

Figure 4: An example of propagation. This example shows that we need to score the seed lexicon first. Otherwise, the label propagation will bring more noise to lexicon learning.

predicate (e.g., *PlaceOfBirth*) if the phrase triggers the predicate. The other is that it is easy to compute the similarity between unigrams. Although this technique will raise more ambiguity, we use an extra feature template to handle this.

Next, we select high quality lexicons by scoring each lexical entry. Specifically, we use the lexicon gained in first step to train a semantic parser, and define several features to measure the quality of each lexical entry. After training, we can compute the score for each lexical entry. Since the higher the score of a lexical entry, the better its quality. We pick top K (K=5) lexical entries for each predicates as our seed lexicon. It is important to assign score to each lexical entry in the seed lexicon. As Figure 4 shows if we don't assign score, as there are many incorrect lexical entries in the seed lexicon, and these lexical entries will bring more noise to the lexicon when doing graph propagation.

3.2 Graph Construction

We construct a graph over all phrases in the seed lexicons and words which occur in the whole data. Besides, we also consider bridge words, which are both near to labeled node and unlabeled node. There are three types of nodes in the graph (As Figure 5 shows): Labeled nodes represent words in the seed lexicon; unlabeled nodes represent words in the whole data; bridge nodes represent the shared nearest neighbor nodes for the labeled nodes and unlabeled nodes.

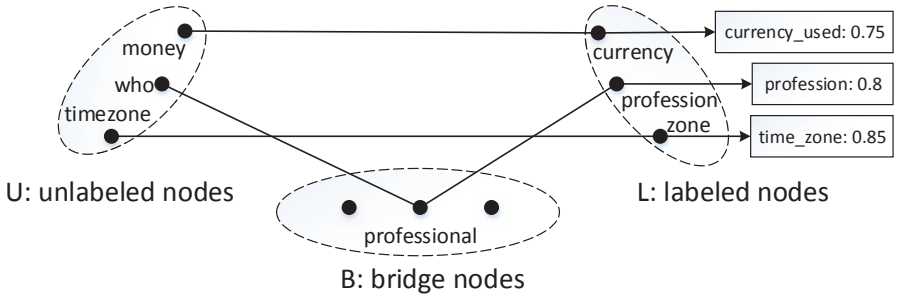

Figure 5: The graph between labeled nodes, unlabeled nodes and bridge nodes. Mapping can propagate either directly from labeled nodes to unlabeled nodes or indirectly via bridge nodes.

We use three resources to compute similarity for graph construction.

First, we use distributional representations for phrases to compute the similarity. Recently, a fair amount of research has showed that the word vector is quite useful for natural language process, especially for tasks related to similarity. Our purpose is to find similar words for the labeled ones, and label

neighbor words	scores
guilder	3.05
coin	3.03
taxa	3.00
les	2.89
exchange	2.85
monetary	2.76
money	2.63

Table 1: Nearest neighbor words for "*currency*" from PPDB-2.0.

them with the same labels. We use the published word vector (Huang et al., 2012) directly, and use cosine distance for similarity.

The second resource we used is paraphrase tables. In this part, we want to utilize paraphrase pairs, like "*money*" and "*currency*". We construct these pairs using the Paralex corpus (Fader et al., 2013). Paralex is a large question paraphrases corpus from WikiAnswers,[6] and each clique questions were tagged as expressing the same meaning by users. Paraphrase pairs in Paralex are word-aligned using standard machine translation methods. We use the word alignments to construct a word table by applying the consistent word pair heuristic to only unigram. This paraphrase tables is suitable for our needs since it focuses on question paraphrases.

The third resource we used is PPDB. Specifically, we use PPDB-2.0 (Pavlick et al., 2015) to calculate the similarity between two phrases by utilizing their scores, which consider many aspects. As we argued before, we only consider single word. So we use the lexical part of PPDB-2.0. Moreover, we pre-process the PPDB dataset by lemming the words. Table 1 shows several nearest neighbor words for the word "*currency*" from PPDB-2.0.

In fact, we can also use lexicon resources like synset from WordNet,[7] and Allen (2014) has used the VerbNet for learning a lexicon for broad-coverage semantic parsing. However, we find that the synsets for words are almost covered by the resources mentioned before. So we don't use these resources here.

In order to limit the graph size, we consider the top 10 nearest labeled nodes and top 5 nearest bridge nodes for each unlabeled one; for the each bridge node, we consider the top 5 nearest labeled nodes. Moreover, we also consider edges between labeled nodes and labeled nodes.

Finally, the overall similarity score between two given phrases w_1 and w_2 is computed as follow:

$$sim(w_1, w_2) = \alpha sim_1(w_1, w_2) + \beta sim_2(w_1, w_2) + (1 - \alpha - \beta) sim_3(w_1, w_2) \quad (1)$$

Before the final computing, we normalize each similarity score which are obtained using three resources separately. The hyperparameters are turned by development training.

3.3 Graph Propagation

Graph propagation is used to propagate the labels from labeled nodes to unlabeled ones by following the graph's structure. This approach is based on the smoothness assumption: similar nodes in the graph have similar labels. This paper utilizes the modified Adsorption algorithm (Talukdar and Crammer, 2009).

$$\min_{\widehat{Y}} \quad \mu_1 \sum_{v \in V_L} p_1 \, ||Y_v - \widehat{Y}_v||_2^2 \; + \mu_2 \sum_{v,u} p_2 \, W_{v,u} \, ||\widehat{Y}_v - \widehat{Y}_u||_2^2 \; + \mu_3 \sum_{v} p_3 \, ||\widehat{Y}_v - R_u||_2^2 \quad (2)$$

There are three parts in Formula (2), the first part enforces the labels of the seed nodes to keep unchanged. The second part enforces the smoothness, making similar nodes have similar labels. The third part enforces an uniform distribution for the unlabeled nodes. We use the Junto label propagation toolkit[8] for label propagation.

[6] http://www.answers.com/Q/

[7] https://wordnet.princeton.edu/

[8] https://github.com/parthatalukdar/junto

4 Semantic Parsing with Extended Lexicon

After graph propagation, each unlabeled phrase is labeled with a distribution over the set of predicates. We use SEMPRE (Berant et al., 2013; Berant and Liang, 2015) as our base semantic parser. In order to use the learned lexicon, we add a feature which indicates the final score for each lexical entry. The semantic parser will train on the training data with the learned lexicon as its initial lexicon. Following Berant and Liang (2015), we also use the feature template that conjoins predicates and content lemmas, and this feature template has been proved very helpful in Berant and Liang (2015).

5 Experiments

We evaluate our method on two benchmark datasets: WEBQUESTIONS and FREE917.

Dataset: WEBQUESTIONS dataset (Berant et al., 2013) contains 5,810 question-answer pairs. These questions are collected by crawling the Google Suggest API, and the answers are obtained using Amazon Mechanical Turk. This dataset covers several popular topics and its questions are commonly asked on the web. In our experiments, we use the standard train-test split (Berant et al., 2013), i.e., 3,778 questions (65%) for training and 2,032 questions (35%) for testing.

The FREE917 dataset (Cai and Yates, 2013a) contains 917 questions, annotated with logical forms. This dataset covers a wide range of domains. One example is "*what fuel does an internal combustion engine use*". Following Cai and Yates (2013a), we use the original split of the questions into 70% questions (641) to train and 30% questions (276) to test.

Setup: In our experiments, we use the Freebase Search API for entity lookup in WEBQUESTIONS dataset, and build a Lucene index over the 41M Freebase entities to map entities in FREE917 dataset. We load Freebase using Virtuoso, and execute logical forms by converting them to SPARQL and querying using Virtuoso. We learn the parameters of our system by making several passes (3 for WEBQUESTIONS and 6 for FREE917) over the training dataset, with the beam size (200 in WEBQUESTIONS and 500 for FREE917).

For the similarity computation, we set $\alpha = 0.05$, $\beta = 0.85$. For the parameters in Junto, we set $\mu_1 = 0.55$, $\mu_2 = 0.44$, $\mu_3 = 0.01$, $\beta = 2$.

Comparing systems: To evaluate our method, we mainly compare our system (Base + lexicon) to the base system (Base) which does not use the learned wide-coverage lexicon, also to system (Base + bridge) which utilize bridge operator to serve as lexicon (Berant et al., 2013). We also compare to several nearly published systems, including semantic parsing based system (Kwiatkowski et al., 2013; Berant and Liang, 2015), information extraction based systems (Yao and Van Durme, 2014; Yao, 2015), machine translation based systems (Bao et al., 2014), embedding based systems (Bordes et al., 2014; Yang et al., 2014), and QA based system (Bast and Haussmann, 2015).

5.1 Experimental Results

Table 2 and Table 3 provide the performances of all baselines[9] and our method in WEBQUESTIONS and FREE917. From Table 2 and Table 3, we can see that:

1. Our method achieves competitive performance: Our system outperforms base system (Base) greatly and gets a better performance when comparing to the base system with a bridge operator (Base + bridge).

2. The learned lexicon has wider coverage than the seed one: Our system obtains higher recall than the Base. By utilizing large amount of text corpora and lexical resources, the extended lexicon improves the semantic parsing system. For FREE917, our system gains the highest recall. This indicates that our lexicon really has wider coverage, especially for dataset with more domains like FREE917.

3. The bridge operation from Berant and Liang (2015) is quite powerful. It can resolve the problem of lexicon coverage to some degree. And our approach, which learns the lexicon directly, can gain a better performance.

[9]We collect the results of other systems from https://nlp.stanford.edu/software/sempre/.

System	Prec.	Rec.	F1 (avg)
Berant et al. (2013)	48.0	41.3	35.7
Yao and Van Durme (2014)	51.7	45.8	33.0
Berant and Liang (2014)	40.5	46.6	39.9
Bao et al. (2014)	–	–	37.5
Bordes et al. (2014)	–	–	39.2
Yang et al. (2014)	–	–	41.3
Bast and Haussmann (2015)	49.8	60.4	49.4
Yao (2015)	52.6	54.5	44.3
Berant and Liang (2015)	50.5	55.7	49.7
Yih et al. (2015)	52.8	60.7	52.5
Base	51.0	47.6	40.5
Base + bridge	50.0	58.5	50.0
Our approach	51.6	59.7	51.2

Table 2: The results of our system and recently published systems on WEBQUESTIONS.

System	Prec.	Rec.	F1
Cai and Yates (2013a)	67.0	59.0	63.0
Kwiatkowski et al. (2013)	76.7	68.0	72.1
Bast and Haussmann (2015)	72.0	67.8	69.8
Base	71.2	59.5	64.8
Base + bridge	69.4	64.4	66.8
Our approach	71.5	67.9	69.6

Table 3: The results of our system and recently published systems on FREE917.

4. Compared to all baselines, our system gets a competitive recall. This result indicates that our parser can parse more sentences when the lexicon has wider coverage. Interestingly, for WEBQUESTIONS, both the two systems with the highest recall (Bast and Haussmann, 2015; Yih et al., 2015) rely on extra-techniques such as entity linking and relation matching.

In Section 3.1, we normalize the seed lexicon using the unigram for the lexeme. By this way, the final graph for label propagation will not be too large, and it is convenient to compute the similarity between word and word using text corpora and lexical resources. We design some experiments to evaluate the new seed lexicon (unigram for lexeme). Table 4 and Table 5 shows the results. We can see that the system using the new seed lexicon has similar performance to the system using the original seed lexicon.

System	Prec.	Rec.	F1 (avg)
Original seed	40.6	47.5	**40.6**
New seed	**51.0**	**47.6**	40.5

Table 4: The results of using different seed lexicons on WEBQUESTIONS dataset.

System	Prec.	Rec.	F1 (avg)
Original seed	69.8	59.0	63.9
New seed	**71.2**	**59.5**	**64.8**

Table 5: The results of using different seed lexicons on FREE917 dataset.

System	Prec.	Rec.	F1 (avg)
Only word vector	41.8	45.9	39.7
Only paraphrase table	50.0	46.3	39.4
Only PPDB-2.0	51.2	58.7	50.6
All	**51.6**	**59.7**	**51.2**

Table 6: The results of using different resources for measuring similarities on WEBQUESTIONS dataset.

To evaluate the text corpora and lexical resources we used, we also conduct several experiments on WEBQUESTIONS. Table 6 shows the results. We can see that:

1. Only using word vector for similarity computation, the final result is not ideal. We think that the word vector consider many aspects in similarity, and in lexicon learning, what we expect for similarity is paraphrasing.

2. The paraphrase table pairs help a little. We think that is due to we use simple alignment for scoring.

3. The PPDB-2.0 serves quite well, even only use PPDB-2.0 score for similarity computation. We think that the PPDB-2.0 was extracted from paraphrase corpus, so the similar lexicon are almost paraphrase to each other.

4. Using word vector, paraphrase align table pairs and PPDB-2.0 score together achieves the best performance.

5.2 Analysis

Our aim is to learn a wide-coverage lexicon for semantic parsing. Our approach utilizes text corpora and lexical resources to extend seed lexicon. Table 7 shows several learned new mappings with the final score from the semantic parser. We can find that after label propagation, we can obtain new lexical entries which can improve the coverage of semantic parser. The results proved our intuition, i.e., the unlabeled phrase maps to the same predicate of its nearest labeled phrases.

Predicate	Seed phrase	Learned phrase	score
`Currency`	*currency*	*money*	4.91
`Education`	*education*	*school*	4.75
`Religion`	*religion*	*believe*	2.30
`Profession`	*professional*	*who*	2.30

Table 7: Several learned lexical entries with un-normalized scores on WEBQUESTIONS dataset.

The learned new lexicon has wide coverage, however, this means the accuracy for the lexicon will be influenced. Berant and Liang (2015) added new lexicalized features (lemmaAndBinary) that connect natural language phrases to binary predicates. For example, given the utterance "*What countries have german as the official language?*", the predicate for phrase "*language*" can be `Language-spoken` and `Offical-language`. The added feature will conjoin binaries with all content word lemmas. After observing enough examples, the phrase "*language*" will map to `Offical-language` when has "*official*" as its content, because the feature, which corresponds to "*offical*" and `Offical-language`, will be up-weighted. Berant and Liang (2015) have proved this feature is really helpful. In our experiments, we also use this feature. Table 8 shows the ablation test results. We can see that, this feature improves our system greatly, especially for precision. We think that as we only use the unigram as our lexeme for lexical entry, the lexicon has more noise. And the alignments between predicates with content words will help the parser to choose the right lexicon during parsing.

System	Prec.	Rec.	F1 (avg)
Our system - feature	48.0	48.9	41.8
Our system	**51.6**	**59.7**	**51.2**

Table 8: The results of ablation test for the lemmaAndBinary feature.

Manual error analysis To better understand our system, we manually inspected the errors our system made. We found that many errors are due to mistakes in labeling. The rest of the errors are mainly complicate cases, like N-ary predicate (event in Freebase), superlative, temporal clause etc. We argue that more attention should be given to these complicated cases.

6 Conclusion

In this paper, we make use of low-cost, easily obtained text corpora and lexical resources in a graph-based semi-supervised learning framework to learn lexicon for semantic parsing. Experiments demonstrate that our method improves the semantic parsing system, especially, when the lexicon is not covered in the training data. Our method can learn wide-coverage lexicon for open-domain semantic parsing.

Traditionally, a semantic parser needs a lexicon first, and then parses the sentence in a bottom-up way. For these parsers, the lexicon is extremely important, and it is hard to learn lexicon with high coverage. Currently, some semantic parsers use the knowledge base in advance, and utilize entity linking and relation matching during parsing, and these methods parse the sentence like a top-down way. As the knowledge base is huge, the searching space is usually quite large. In future work, We want to design parsing algorithm which can take advantages from both sides.

Acknowledgments

This work is supported by the National Natural Science Foundation of China under Grants no. 61433015, 61572477 and 61772505, and the Young Elite Scientists Sponsorship Program no. YESS20160177. Moreover, we sincerely thank the reviewers for their valuable comments.

References

James Allen. 2014. Learning a lexicon for broad-coverage semantic parsing. In *Proceedings of the ACL 2014 Workshop on Semantic Parsing*, pages 1–6, Baltimore, MD, June. Association for Computational Linguistics.

Yoav Artzi and Luke Zettlemoyer. 2013. Weakly supervised learning of semantic parsers for mapping instructions to actions. *Transactions of the Association for Computational Linguistics*, 1(1):49–62.

Yoav Artzi, Dipanjan Das, and Slav Petrov. 2014. Learning compact lexicons for ccg semantic parsing. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1273–1283, Doha, Qatar, October. Association for Computational Linguistics.

Junwei Bao, Nan Duan, Ming Zhou, and Tiejun Zhao. 2014. Knowledge-based question answering as machine translation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 967–976, Baltimore, Maryland, June. Association for Computational Linguistics.

Hannah Bast and Elmar Haussmann. 2015. More accurate question answering on freebase. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management, CIKM 2015, Melbourne, VIC, Australia, October 19 - 23, 2015*, pages 1431–1440.

Jonathan Berant and Percy Liang. 2014. Semantic parsing via paraphrasing. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1415–1425, Baltimore, Maryland, June. Association for Computational Linguistics.

Jonathan Berant and Percy Liang. 2015. Imitation learning of agenda-based semantic parsers. *Transactions of the Association for Computational Linguistics*, 3:545–558.

Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. 2013. Semantic parsing on Freebase from question-answer pairs. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1533–1544, Seattle, Washington, USA, October. Association for Computational Linguistics.

Antoine Bordes, Sumit Chopra, and Jason Weston. 2014. Question answering with subgraph embeddings. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 615–620, Doha, Qatar, October. Association for Computational Linguistics.

Qingqing Cai and Alexander Yates. 2013a. Large-scale semantic parsing via schema matching and lexicon extension. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 423–433, Sofia, Bulgaria, August. Association for Computational Linguistics.

Qingqing Cai and Alexander Yates. 2013b. Semantic parsing freebase: Towards open-domain semantic parsing. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 1: Proceedings of the Main Conference and the Shared Task: Semantic Textual Similarity*, pages 328–338, Atlanta, Georgia, USA, June. Association for Computational Linguistics.

Bo Chen, Le Sun, Xianpei Han, and Bo An. 2016. Sentence rewriting for semantic parsing. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 766–777, Berlin, Germany, August. Association for Computational Linguistics.

Dipanjan Das and Noah A. Smith. 2011. Semi-supervised frame-semantic parsing for unknown predicates. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 1435–1444, Portland, Oregon, USA, June. Association for Computational Linguistics.

Anthony Fader, Luke Zettlemoyer, and Oren Etzioni. 2013. Paraphrase-driven learning for open question answering. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1608–1618, Sofia, Bulgaria, August. Association for Computational Linguistics.

Manaal Faruqui, Ryan McDonald, and Radu Soricut. 2016. Morpho-syntactic lexicon generation using graph-based semi-supervised learning. *Transactions of the Association for Computational Linguistics*, 4:1–16.

William L. Hamilton, Kevin Clark, Jure Leskovec, and Dan Jurafsky. 2016. Inducing domain-specific sentiment lexicons from unlabeled corpora. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 595–605, Austin, Texas, November. Association for Computational Linguistics.

Eric Huang, Richard Socher, Christopher Manning, and Andrew Ng. 2012. Improving word representations via global context and multiple word prototypes. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 873–882, Jeju Island, Korea, July. Association for Computational Linguistics.

Robin Jia and Percy Liang. 2016. Data recombination for neural semantic parsing. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12–22, Berlin, Germany, August. Association for Computational Linguistics.

Jayant Krishnamurthy and Tom Mitchell. 2012. Weakly supervised training of semantic parsers. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 754–765, Jeju Island, Korea, July. Association for Computational Linguistics.

Jayant Krishnamurthy and Tom M. Mitchell. 2014. Joint syntactic and semantic parsing with combinatory categorial grammar. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1188–1198, Baltimore, Maryland, June. Association for Computational Linguistics.

Jayant Krishnamurthy. 2016. Probabilistic models for learning a semantic parser lexicon. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 606–616, San Diego, California, June. Association for Computational Linguistics.

Tom Kwiatkowksi, Luke Zettlemoyer, Sharon Goldwater, and Mark Steedman. 2010. Inducing probabilistic CCG grammars from logical form with higher-order unification. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 1223–1233, Cambridge, MA, October. Association for Computational Linguistics.

Tom Kwiatkowski, Luke Zettlemoyer, Sharon Goldwater, and Mark Steedman. 2011. Lexical generalization in ccg grammar induction for semantic parsing. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1512–1523, Edinburgh, Scotland, UK., July. Association for Computational Linguistics.

Tom Kwiatkowski, Eunsol Choi, Yoav Artzi, and Luke Zettlemoyer. 2013. Scaling semantic parsers with on-the-fly ontology matching. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1545–1556, Seattle, Washington, USA, October. Association for Computational Linguistics.

Junhui Li, Muhua Zhu, Wei Lu, and Guodong Zhou. 2015. Improving semantic parsing with enriched synchronous context-free grammar. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1455–1465, Lisbon, Portugal, September. Association for Computational Linguistics.

Percy Liang, Michael Jordan, and Dan Klein. 2011. Learning dependency-based compositional semantics. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 590–599, Portland, Oregon, USA, June. Association for Computational Linguistics.

Chen Liang, Jonathan Berant, Quoc Le, Kenneth D. Forbus, and Ni Lao. 2017. Neural symbolic machines: Learning semantic parsers on freebase with weak supervision. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 23–33, Vancouver, Canada, July. Association for Computational Linguistics.

Wei Lu, Hwee Tou Ng, Wee Sun Lee, and Luke S. Zettlemoyer. 2008. A generative model for parsing natural language to meaning representations. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 783–792, Honolulu, Hawaii, October. Association for Computational Linguistics.

Ramtin Mehdizadeh Seraj, Maryam Siahbani, and Anoop Sarkar. 2015. Improving statistical machine translation with a multilingual paraphrase database. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1379–1390, Lisbon, Portugal, September. Association for Computational Linguistics.

Ellie Pavlick, Pushpendre Rastogi, Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. 2015. Ppdb 2.0: Better paraphrase ranking, fine-grained entailment relations, word embeddings, and style classification. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 425–430, Beijing, China, July. Association for Computational Linguistics.

Majid Razmara, Maryam Siahbani, Reza Haffari, and Anoop Sarkar. 2013. Graph propagation for paraphrasing out-of-vocabulary words in statistical machine translation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1105–1115, Sofia, Bulgaria, August. Association for Computational Linguistics.

Siva Reddy, Oscar Täckström, Michael Collins, Tom Kwiatkowski, Dipanjan Das, Mark Steedman, and Mirella Lapata. 2016. Transforming Dependency Structures to Logical Forms for Semantic Parsing. *Transactions of the Association for Computational Linguistics*, 4:127–140.

Avneesh Saluja, Hany Hassan, Kristina Toutanova, and Chris Quirk. 2014. Graph-based semi-supervised learning of translation models from monolingual data. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 676–686, Baltimore, Maryland, June. Association for Computational Linguistics.

Partha Pratim Talukdar and Koby Crammer. 2009. New regularized algorithms for transductive learning. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 442–457. Springer.

Yuk Wah Wong and Raymond Mooney. 2007. Learning synchronous grammars for semantic parsing with lambda calculus. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 960–967, Prague, Czech Republic, June. Association for Computational Linguistics.

Chunyang Xiao, Marc Dymetman, and Claire Gardent. 2016. Sequence-based structured prediction for semantic parsing. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1341–1350, Berlin, Germany, August. Association for Computational Linguistics.

Min-Chul Yang, Nan Duan, Ming Zhou, and Hae-Chang Rim. 2014. Joint relational embeddings for knowledge-based question answering. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 645–650, Doha, Qatar, October. Association for Computational Linguistics.

Xuchen Yao and Benjamin Van Durme. 2014. Information extraction over structured data: Question answering with freebase. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 956–966, Baltimore, Maryland, June. Association for Computational Linguistics.

Xuchen Yao. 2015. Lean question answering over freebase from scratch. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, pages 66–70, Denver, Colorado, June. Association for Computational Linguistics.

Wen-tau Yih, Ming-Wei Chang, Xiaodong He, and Jianfeng Gao. 2015. Semantic parsing via staged query graph generation: Question answering with knowledge base. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1321–1331, Beijing, China, July. Association for Computational Linguistics.

John M. Zelle and Raymond J. Mooney. 1996. Learning to parse database queries using inductive logic programming. In *AAAI/IAAI*, pages 1050–1055, Portland, OR, August. AAAI Press/MIT Press.

Luke S. Zettlemoyer and Michael Collins. 2005. Learning to map sentences to logical form: Structured classification with probabilistic categorial grammars. In *UAI '05, Proceedings of the 21st Conference in Uncertainty in Artificial Intelligence, Edinburgh, Scotland, July 26-29, 2005*, pages 658–666.

Kai Zhao, Hany Hassan, and Michael Auli. 2015. Learning translation models from monolingual continuous representations. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1527–1536, Denver, Colorado, May–June. Association for Computational Linguistics.