Joint Relational Embeddings for Knowledge-based Question Answering

Min-Chul Yang[†]   **Nan Duan**[‡]   **Ming Zhou**[‡]   **Hae-Chang Rim**[†]

[†]Dept. of Computer & Radio Comms. Engineering, Korea University, Seoul, South Korea
[‡]Microsoft Research Asia, Beijing, China
`mcyang@nlp.korea.ac.kr`
`{nanduan, mingzhou}@microsoft.com`
`rim@nlp.korea.ac.kr`

Abstract

Transforming a natural language (NL) question into a corresponding logical form (LF) is central to the knowledge-based question answering (KB-QA) task. Unlike most previous methods that achieve this goal based on mappings between lexicalized phrases and logical predicates, this paper goes one step further and proposes a novel embedding-based approach that maps NL-questions into LFs for KB-QA by leveraging semantic associations between lexical representations and KB-properties in the latent space. Experimental results demonstrate that our proposed method outperforms three KB-QA baseline methods on two publicly released QA data sets.

1   Introduction

Knowledge-based question answering (KB-QA) involves answering questions posed in natural language (NL) using existing knowledge bases (KBs). As most KBs are structured databases, how to transform the input question into its corresponding structured query for KB (KB-query) as a logical form (LF), also known as semantic parsing, is the central task for KB-QA systems. Previous works (Mooney, 2007; Liang et al., 2011; Cai and Yates, 2013; Fader et al., 2013; Berant et al., 2013; Bao et al., 2014) usually leveraged mappings between NL phrases and logical predicates as lexical triggers to perform transformation tasks in semantic parsing, but they had to deal with two limitations: (i) as the meaning of a logical predicate often has different natural language expression (NLE) forms, the lexical triggers extracted for a predicate may at times are limited in size; (ii) entities detected by the named entity recognition (NER) component will be used to compose the

logical forms together with the logical predicates, so their types should be consistent with the predicates as well. However, most NER components used in existing KB-QA systems are independent from the NLE-to-predicate mapping procedure.

We present a novel embedding-based KB-QA method that takes all the aforementioned limitations into account, and maps NLE-to-entity and NLE-to-predicate simultaneously using simple vector operations for structured query construction. First, low-dimensional embeddings of n-grams, entity types, and predicates are jointly learned from an existing knowledge base and from entries <*entity_{subj}*, *NL relation phrase*, *entity_{obj}*> that are mined from NL texts labeled as KB-properties with weak supervision. Each such entry corresponds to an NL expression of a triple <*entity_{subj}*, *predicate*, *entity_{obj}*> in the KB. These embeddings are used to measure the semantic associations between lexical phrases and two properties of the KB, entity type and logical predicate. Next, given an NL-question, all possible structured queries as candidate LFs are generated and then they are ranked by the similarity between the embeddings of observed features (n-grams) in the NL-question and the embeddings of logical features in the structured queries. Last, answers are retrieved from the KB using the selected LFs.

The contributions of this work are two-fold: (1) as a smoothing technique, the low-dimensional embeddings can alleviate the coverage issues of lexical triggers; (2) our joint approach integrates entity span selection and predicate mapping tasks for KB-QA. For this we built independent entity embeddings as the additional component, solving the entity disambiguation problem.

2   Related Work

Supervised semantic parsers (Zelle and Mooney, 1996; Zettlemoyer and Collins, 2005; Mooney, 2007) heavily rely on the <*sentence, semantic an-*

notation> pairs for lexical trigger extraction and model training. Due to the data annotation requirement, such methods are usually restricted to specific domains, and struggle with the coverage issue caused by the limited size of lexical triggers.

Studies on weakly supervised semantic parsers have tried to reduce the amount of human supervision by using question-answer pairs (Liang et al., 2011) or distant supervision (Krishnamurthy and Mitchell, 2012) instead of full semantic annotations. Still, for KB-QA, the question of how to leverage KB-properties and analyze the question structures remains.

Bordes et al. (2012) and Weston et al. (2013) designed embedding models that connect free texts with KBs using the relational learning method (Weston et al., 2010). Their inputs are often statement sentences which include subject and object entities for a given predicate, whereas NL-questions lack either a subject or object entity that is the potential answer. Hence, we can only use the information of a subject or object entity, which leads to a different training instance generation procedure and a different training criterion.

Recently, researchers have developed open domain systems based on large scale KBs such as FREEBASE[1] (Cai and Yates, 2013; Fader et al., 2013; Berant et al., 2013; Kwiatkowski et al., 2013; Bao et al., 2014; Berant and Liang, 2014; Yao and Van Durme, 2014). Their semantic parsers for Open QA are unified formal and scalable: they enable the NL-question to be mapped into the appropriate logical form. Our method obtains similar logical forms, but using only low-dimensional embeddings of n-grams, entity types, and predicates learned from texts and KB.

3 Setup

3.1 Relational Components for KB-QA

Our method learns semantic mappings between NLEs and the KB[2] based on the paired relationships of the following three components: $\mathcal{C}$ denotes a set of bag-of-words (or n-grams) as context features (c) for NLEs that are the lexical representations of a logical predicate (p) in KB; $\mathcal{T}$ denotes a set of entity types (t) in KB and each type can be used as the abstract expression of a subject entity

(s) that occurs in the input question; $\mathcal{P}$ denotes a set of logical predicates (p) in KB, each of which is the canonical form of different NLEs sharing an identical meaning (bag-of-words; c).

Based on the components defined above, the paired relationships are described as follows: $\mathcal{T}$-$\mathcal{P}$ can investigate the relationship between subject entity and logical predicate, as object entity is always missing in KB-QA; $\mathcal{C}$-$\mathcal{T}$ can scrutinize subject entity's attributes for the entity span selection such as its positional information and relevant entity types to the given context, which may solve the entity disambiguation problem in KB-QA; $\mathcal{C}$-$\mathcal{P}$ can leverage the semantic overlap between question contexts (n-gram features) and logical predicates, which is important for mapping NL-questions to their corresponding predicates.

3.2 NLE-KB Pair Extraction

This section describes how we extract the semantic associated pairs of NLE-entries and KB-triples to learn the relational embeddings (Section 4.1).

<**Relation Mention, Predicate**> **Pair** ($\mathcal{MP}$) Each relation mention denotes a lexical phrase of an existing KB-predicate. Following information extraction methods, such as PATTY (Nakashole et al., 2012), we extracted the <*relation mention, logical predicate*> pairs from English WIKIPEDIA[3], which is closely connected to our KB, as follows: Given a KB-triple <*entity$_{subj}$, logical predicate, entity$_{obj}$*>, we extracted NLE-entries <*entity$_{subj}$, relation mention, entity$_{obj}$*> where *relation mention* is the shortest path between *entity$_{subj}$* and *entity$_{obj}$* in the dependency tree of sentences. The assumption is that any *relation mention* (m) in the NLE-entry containing such entity pairs that occurred in the KB-triple is likely to express the *predicate* (p) of that triple.

With obtaining high-quality $\mathcal{MP}$ pairs, we kept only relation mentions that were highly associated with a predicate measured by the scoring function:

$$S(m,p) = \text{PMI}(e_m; e_p) + \text{PMI}(u_m; u_p) \quad (1)$$

where e_x is the set of total pairs of both-side entities of entry x (m or p) and u_x is the set of unique (distinct) pairs of both-side entities of entry x. In this case, the both-side entities indicate *entity$_{subj}$* and *entity$_{obj}$*. For a frequency-based probability, $\text{PMI}(x;y) = \log \frac{P(x,y)}{P(x)P(y)}$

---

[3]http://en.wikipedia.org/

(Church and Hanks, 1990) can be re-written as $\text{PMI}(x; y) = \log \frac{|x \bigcap y| \cdot C}{|x| \cdot |y|}$, where C denotes the total number of items shown in the corpus. The function is partially derived from the *support* score (Gerber and Ngonga Ngomo, 2011), but we focus on the correlation of shared entity pairs between relation mentions and predicates using the PMI computation.

<Question Pattern, Predicate> Pair ($\mathcal{QP}$) Since WIKIPEDIA articles have no information to leverage interrogative features which highly depend on the object entity (answer), it is difficult to distinguish some questions that are composed of only different *5W1H* words, e.g., {*When|Where*} *was Barack Obama born?* Hence, we used the method of collecting question patterns with human labeled predicates that are restricted by the set of predicates used in $\mathcal{MP}$ (Bao et al., 2014).

4 Embedding-based KB-QA

Our task is as follows. First, our model learns the semantic associations of $\mathcal{C}$-$\mathcal{T}$, $\mathcal{C}$-$\mathcal{P}$, and $\mathcal{T}$-$\mathcal{P}$ (Section 3.1) based on NLE-KB pairs (Section 3.2), and then predicts the semantic-related KB-query which can directly find the answer to a given NL-question.

For our feature space, given an NLE-KB pair, the NLE (*relation mention* in $\mathcal{MP}$ or *question pattern* in $\mathcal{QP}$) is decomposed into n-gram features: $\mathbb{C} = \{c \mid c \text{ is a segment of NLE}\}$, and the KB-properties are represented by entity type t of *entity$_{subj}$* and predicate p. Then we can obtain a training triplet $w = [\mathbb{C}, t, p]$. Each feature ($c \in \mathcal{C}, t \in \mathcal{T}, p \in \mathcal{P}$) is encoded in the distributed representation which is n-dimensional embedding vectors ($\mathbb{E}^n$): $\forall x, x \overset{encode}{\Rightarrow} \mathbb{E}(x) \in \mathbb{E}^n$.

All n-gram features ($\mathbb{C}$) for an NLE are merged into one embedding vector to help speed up the learning process: $\mathbb{E}(\mathbb{C}) = \sum_{c \in \mathbb{C}} \mathbb{E}(c)/|\mathbb{C}|$. This feature representation is inspired by previous work in embedding-based relation extraction (Weston et al., 2013), but differs in the following ways: (1) entity information is represented on a separate embedding, but its positional information remains as symbol ⟨entity⟩; (2) when the vectors are combined, we use the average of each index to normalize features.

For our joint relational approach, we focus on the set of paired relationships $\mathbb{R} = \{\mathbb{C}$-$t$, $\mathbb{C}$-p, t-$p\}$ that can be semantically leveraged. Formally, these features are embedded into the same latent

space ($\mathbb{E}^n$) and their semantic similarities can be computed by a dot product operation:

$$Sim(a, b) = Sim(r_{ab}) = \mathbb{E}(a)^{\mathsf{T}} \mathbb{E}(b) \quad (2)$$

where r_{ab} denotes a paired relationship a-b (or (a, b)) in the above set $\mathbb{R}$. We believe that our joint relational learning can smooth the surface (lexical) features for semantic parsing using the aligned entity and predicate.

4.1 Joint Relational Embedding Learning

Our ranking-based relational learning is based on a ranking loss (Weston et al., 2010) that supports the idea that the similarity scores of observed pairs in the training set (*positive instances*) should be larger than those of any other pairs (*negative instances*):

$$\forall i, \forall y' \neq y_i, Sim(x_i, y_i) > 1 + Sim(x_i, y') \quad (3)$$

More precisely, for each triplet $w_i = [\mathbb{C}_i, t_i, p_i]$ obtained from an NLE-KB pair, the relationships $\mathbb{R}_i = \{\mathbb{C}_i$-$t_i$, $\mathbb{C}_i$-p_i, t_i-$p_i\}$ are trained under the soft ranking criterion, which conducts Stochastic Gradient Descent (SGD). We thus aim to minimize the following:

$$\forall i, \forall y' \neq y_i, \max(0, 1 - Sim(x_i, y_i) + Sim(x_i, y')) \quad (4)$$

Our learning strategy is as follows. First, we initialize embedding space $\mathbb{E}^n$ by randomly giving mean 0 and standard deviation $1/n$ to each vector. Then for each training triplet w_i, we select the negative pairs against positive pairs ($\mathbb{C}_i$-t_i, $\mathbb{C}_i$-p_i, and t_i-p_i) in the triplet. Last, we make a stochastic gradient step to minimize Equation 4 and update $\mathbb{E}^n$ at each step.

4.2 KB-QA using Embedding Models

Our goal for KB-QA is to translate a given NL-question to a KB-query with the form *<subject entity, predicate, ?>*, where *?* denotes the answer entity we are looking for. The decoding process consists of two stages. The first stage involves generating all possible KB-queries ($\mathcal{K}^q$) for an NL-question q. We first extract n-gram features ($\mathbb{C}^q$) from the NL-question q. Then for a KB-query k^q, we find all available entity types (t^q) of the identified subject entities (s^q) using the dictionary-based entity detection on the NL-question q (all of spans can be candidate entities), and assign all items of predicate set ($\mathcal{P}$) as the candidate predicates (p^q). Like the training triplets,

q	*where is the city of david?*
$\hat{k}(q)$	[The City of David, contained_by, ?]
$\mathbb{C}^q$	n-grams of "where is ⟨entity⟩ ?"
t^q	location
p^q	contained_by

Table 1: The corresponding KB-query $\hat{k}(q)$ for a NL-question q and its decoding triplet w^q.

	# Entries	Accuracy
$\mathcal{MP}$ pairs	291,585	89%
$\mathcal{QP}$ pairs	4,764	98%

Table 2: Statistics of NLE-KB pairs

we also represent the above features as the triplet form $w_i^q = [\mathbb{C}_i^q, t_i^q, p_i^q]$ which is directly linked to a KB-query $k_i^q = [s_i^q, p_i^q, ?]$. The second stage involves ranking candidate KB-queries based on the similarity scores between the following paired relationships from the triplet w_i^q: $\mathbb{R}_i^q = \{\mathbb{C}_i^q\text{-}t_i^q, \mathbb{C}_i^q\text{-}p_i^q, t_i^q\text{-}p_i^q\}$. Unlike in the training step, the similarities of $\mathbb{C}_i^q\text{-}t_i^q$ and $\mathbb{C}_i^q\text{-}p_i^q$ are computed by summation of all pairwise elements (each context embedding $\mathbb{E}(c)$, not $\mathbb{E}(\mathbb{C})$, with each paired $\mathbb{E}(t)$ or $\mathbb{E}(p)$) for a more precise measurement. Since similarites of $\mathbb{R}^q$ are calculated on different scales, we normalize each value using Z-score ($Z(x) = \frac{x-\mu}{\sigma}$) (Kreyszig, 1979). The final score is measured by:

$$Sim_{q2k}(q, k^q) = \sum_{r \in \mathbb{R}^q} Z(Sim(r)) \qquad (5)$$

Then, given any NL-question q, we can predict the corresponding KB-query $\hat{k}(q)$:

$$\hat{k}(q) = \arg\max_{k \in \mathcal{K}^q} Sim_{q2k}(q, k) \qquad (6)$$

Last, we can retrieve an answer from the KB using a structured query $\hat{k}(q)$. Table 1 shows an example of our decoding process.

Multi-related Question Some questions include two-subject entities, both of which are crucial to understanding the question. For the question *who plays gandalf in the lord of the rings?* Gandalf (character) and The Lord Of The Rings (film) are explicit entities that should be joined to a pair of the two entities (implicit entity). More precisely, the two entities can be combined into one concatenated entity (character-in-film) using our manual rule, which compares the possible pairs of entity types in the question with the list of pre-defined entity type pairs that can be merged into a concatenated entity. Our solution enables a multi-related question to be transformed to a single-related question which can be directly translated to a KB-query. Then, the two entity

mentions are replaced with the symbol ⟨entity⟩ (who play ⟨entity⟩ in ⟨entity⟩ ?). We regard the result of this transformation as one of the candidate KB-queries in the decoding step.

5 Experiments

Experimental Setting We first performed pre-processing, including lowercase transformation, lemmatization and tokenization, on NLE-KB pairs and evaluation data. We used 71,310 n-grams (uni-, bi-, tri-), 990 entity types, and 660 predicates as relational components shown in Section 3.1. The sum of these three numbers (72,960) equals the size of the embeddings we are going to learn. In Table 2, we evaluated the quality of NLE-KB pairs ($\mathcal{MP}$ and $\mathcal{QP}$) described in Section 3.2. We can see that the quality of $\mathcal{QP}$ pairs is good, mainly due to human efforts. Also, we obtained $\mathcal{MP}$ pairs that have an acceptable quality using *threshold 3.0* for Equation 1, which leverages the redundancy information in the large-scale data (WIKIPEDIA). For our embedding learning, we set the embedding dimension n to 100, the learning rate (λ) for SGD to 0.0001, and the iteration number to 30. To make the decoding procedure computable, we kept only the popular KB-entity in the dictionary to map different entity mentions into a KB-entity.

We used two publicly released data sets for QA evaluations: **Free917** (Cai and Yates, 2013) includes the annotated lambda calculus forms for each question, and covers 81 domains and 635 Freebase relations; **WebQ.** (Berant et al., 2013) provides 5,810 question-answer pairs that are built by collecting common questions from Web-query logs and by manually labeling answers. We used the previous three approaches (Cai and Yates, 2013; Berant et al., 2013; Bao et al., 2014) as our baselines.

Experimental Results Table 3 reports the overall performances of our proposed KB-QA method on the two evaluation data sets and compares them with those of the three baselines. Note that we did not re-implement the baseline systems, but just borrowed the evaluation results reported in their

Methods	Free917	WebQ.
Cai and Yates (2013)	59.00%	N/A
Berant et al. (2013)	62.00%	31.40%
Bao et al. (2014)	N/A	37.50%
Our method	71.38%	41.34%

Table 3: Accuracy on the evaluation data

Methods	Free917	WebQ.
Our method	71.38%	41.34%
w/o $\mathcal{T}$-$\mathcal{P}$	70.65%	40.55%
w/o $\mathcal{C}$-$\mathcal{T}$	67.03%	38.44%
w/o $\mathcal{C}$-$\mathcal{P}$	31.16%	19.24%

Table 4: Ablation of the relationship types

papers. Although the KB used by our system is much larger than FREEBASE, we still think that the experimental results are directly comparable because we disallow all the entities that are not included in FREEBASE.

Table 3 shows that our method outperforms the baselines on both **Free917** and **WebQ.** data sets. We think that using the low-dimensional embeddings of n-grams rather than the lexical triggers greatly improves the coverage issue. Unlike the previous methods which perform entity disambiguation and predicate prediction separately, our method jointly performs these two tasks. More precisely, we consider the relationships $\mathcal{C}$-$\mathcal{T}$ and $\mathcal{C}$-$\mathcal{P}$ simultaneously to rank candidate KB-queries. In Table 1, the most independent NER in KB-QA systems may detect David as the subject entity, but our joint approach can predict the appropriate subject entity The City of David by leveraging not only the relationships with other components but also other relationships at once. The syntax-based (grammar formalism) approaches such as Combinatory Categorial Grammar (CCG) may experience errors if a question has grammatical errors. However, our bag-of-words model-based approach can handle any question as long as the question contains keywords that can help in understanding it.

Table 4 shows the contributions of the relationships ($\mathbb{R}$) between relational components $\mathcal{C}$, $\mathcal{T}$, and $\mathcal{P}$. For each row, we remove the similarity from each of the relationship types described in Section 3.1. We can see that the $\mathcal{C}$-$\mathcal{P}$ relationship plays a crucial role in translating NL-questions to KB-queries, while the other two relationships are slightly helpful.

Result Analysis Since the majority of questions in **WebQ.** tend to be more natural and diverse, our method cannot find the correct answers to many questions. The errors can be caused by any of the following reasons. First, some NLEs cannot be easily linked to existing KB-predicates, making it difficult to find the answer entity. Second, some entities can be mentioned in several different ways, e.g., nickname (*shaq*→Shaquille O'neal) and family name (*hitler*→Adolf Hitler). Third, in terms of KB coverage issues, we cannot detect the entities that are unpopular. Last, feature representation for a question can fail when the question consists of rare n-grams.

The two training sets shown in Section 3.2 are complementary: $\mathcal{QP}$ pairs provide more opportunities for us to learn the semantic associations between interrogative words and predicates. Such resources are especially important for understanding NL-questions, as most of them start with such 5W1H words; on the other hand, $\mathcal{MP}$ pairs enrich the semantic associations between context information (n-gram features) and predicates.

6 Conclusion

In this paper, we propose a novel method that transforms NL-questions into their corresponding logical forms using joint relational embeddings. We also built a simple and robust KB-QA system based on only the learned embeddings. Such embeddings learn the semantic associations between natural language statements and KB-properties from NLE-KB pairs that are automatically extracted from English WIKIPEDIA using KB-triples with weak supervision. Then, we generate all possible structured queries derived from latent logical features of the given NL-question, and rank them based on the similarity scores between those relational attributes. The experimental results show that our method outperforms the latest three KB-QA baseline systems. For our future work, we will build concept-level context embeddings by leveraging latent meanings of NLEs rather than their surface n-grams with the aligned logical features on KB.

References

Junwei Bao, Nan Duan, Ming Zhou, and Tiejun Zhao. 2014. Knowledge-based question answering as machine translation. Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, pages 967–976. Association for Computational Linguistics.

Jonathan Berant and Percy Liang. 2014. Semantic parsing via paraphrasing. Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, pages 1415–1425. Association for Computational Linguistics.

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, Xavier Glorot, Jason Weston, and Yoshua Bengio. 2012. Joint learning of words and meaning representations for open-text semantic parsing. In *In Proceedings of 15th International Conference on Artificial Intelligence and Statistics*.

Qingqing Cai and Alexander Yates. 2013. Large-scale semantic parsing via schema matching and lexicon extension. In *Association for Computational Linguistics (ACL)*, pages 423–433. The Association for Computer Linguistics.

Kenneth Ward Church and Patrick Hanks. 1990. Word association norms, mutual information, and lexicography. *Comput. Linguist.*, 16(1):22–29, March.

Anthony Fader, Luke S. Zettlemoyer, and Oren Etzioni. 2013. Paraphrase-driven learning for open question answering. In *Association for Computational Linguistics (ACL)*, pages 1608–1618. The Association for Computer Linguistics.

Daniel Gerber and Axel-Cyrille Ngonga Ngomo. 2011. Bootstrapping the linked data web. In *1st Workshop on Web Scale Knowledge Extraction @ ISWC 2011*.

E. Kreyszig. 1979. *Advanced Engineering Mathematics*. Wiley.

Jayant Krishnamurthy and Tom M. 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*, EMNLP-CoNLL '12, pages 754–765, Stroudsburg, PA, USA. 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.

Percy Liang, Michael I. 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 - Volume 1*, HLT '11, pages 590–599, Stroudsburg, PA, USA. Association for Computational Linguistics.

RaymondJ. Mooney. 2007. Learning for semantic parsing. In Alexander Gelbukh, editor, *Computational Linguistics and Intelligent Text Processing*, volume 4394 of *Lecture Notes in Computer Science*, pages 311–324. Springer Berlin Heidelberg.

Ndapandula Nakashole, Gerhard Weikum, and Fabian Suchanek. 2012. Patty: A taxonomy of relational patterns with semantic types. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, EMNLP-CoNLL '12, pages 1135–1145, Stroudsburg, PA, USA. Association for Computational Linguistics.

Jason Weston, Samy Bengio, and Nicolas Usunier. 2010. Large scale image annotation: Learning to rank with joint word-image embeddings. *Machine Learning*, 81(1):21–35, October.

Jason Weston, Antoine Bordes, Oksana Yakhnenko, and Nicolas Usunier. 2013. Connecting language and knowledge bases with embedding models for relation extraction. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1366–1371, Seattle, Washington, USA, 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*, pages 956–966, Baltimore, Maryland, June. Association for Computational Linguistics.

John M. Zelle and Raymond J. Mooney. 1996. Learning to parse database queries using inductive logic programming. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence - Volume 2*, AAAI'96, pages 1050–1055. AAAI Press.

Luke S. Zettlemoyer and Michael Collins. 2005. Learning to map sentences to logical form: Structured classification with probabilistic categorial grammars. In *UAI*, pages 658–666. AUAI Press.