An Overview of Microsoft Deep QA System on Stanford WebQuestions Benchmark

Zhenghao Wang, Shengquan Yan, Huaming Wang, and Xuedong Huang

Microsoft Corporation, Redmond WA 98052 USA
September 3rd, 2014

Abstract. Question answering (QA) over an existing knowledge base (KB) such as Microsoft Satori or open Freebase is one of the most important natural language processing applications. There are approaches based on web-search motivated statistic techniques as well as linguistically oriented knowledge engineering. Both methods face the key challenge on how to handle diverse ways of naturally expressing predicates and entities existing in the KB. The domain independent web information extracted from the massive amount of web usage data can be used with traditional semantic parsing through a unified framework. We provide such a unified framework utilizing both statistically motivated information-theoretic embeddings and logically driven proof-theoretic decoding to significantly improve Stanford's WebQuestions QA benchmark. In comparison to Stanford's state of the art ParaSempre 39.9 (F1-score), our Deep QA system achieves 45.3 on the same test data and protocol.

1 Introduction

Question Answering (QA) from an existing knowledge base (KB) has drawn significant interest from our industry and academia (Ferrucci et al. 2010, Kolomiyets et al. 2011, Cai and Yates 2013, Bao et al. 2014, Steedman 2014, Berant et al. 2014, Yao & Van Durme 2014, Bordes et al. 2014, Mooney 2014). There are many QA systems like IBM's Watson that is highly optimized for vertical domains (Ferrucci et al. 2010), UW's probabilistic CCG parsing with on-the-fly ontology matching (Kwiatkowski et al. 2013), Stanford's semantic parsing via paraphrasing (Berant & Liang 2014), and Facebook's subgraph embeddings (Bordes et al. 2014). It is one of the most important natural language processing applications to automatically answer open questions asked in a natural way. Large scale structured KBs have been widely adopted in modern web search engines such as Bing and Google. However, open-domain QA with complex natural language and logic reasoning remains a major scientific and engineering challenge. The scale of the KB and the difficulty for machines to interpret natural language accurately make QA indeed a challenging and interesting problem for Artificial Intelligence. The core component of QA is natural language understanding that can convert natural language questions into KB appropriate queries.

In open QA research systems, the UW system first maps utterances to a domain-independent intermediate logical form, and then performs ontology matching to produce the final logical form (Kwiatkowski et al. 2013). Fader et al. (2013 & 2014) also presented a system that maps questions onto simple queries by learning paraphrases from a large monolingual parallel corpus. In semantic parsing, typically a predefined set of logical constants, or an ontology, is used to construct meaning representations. In practice, few ontologies have sufficient coverage to support meaning representations. To associate unstructured natural language questions with logical predicates in the KB, we must learn that the constructions "What does X do for a living?", "What is X's profession?", and "Who is X?", should all map to the same logical predicate *Profession*. ParaSempre (Berant & Liang 2014) from Stanford offers the state of the art KB QA by using paraphrasing to exploit large amounts of text not covered by the KB. They approach the problem from the opposite end of (Kwiatkowski et al. 2013). They target factoid questions with a modest amount of compositionality. Given an input utterance, they first construct a manageable set of candidate logical forms. Next, they heuristically generate canonical utterances for each logical form based on the text descriptions of predicates from the KB. They finally choose the canonical utterance that best paraphrases the input utterance, and thereby the logical form that generated it. Both the association model based on aligned phrase pairs extracted from a monolingual parallel corpus, and retrained Google vector space model are used to jointly optimize for their QA training pairs.

Facebook recently reported their improved embedding model by providing the ability to answer more complicated questions via subgraph embeddings (Borders et al. 2014). They used a sophisticated inference procedure to consider longer paths and developed a richer representation of the answers which encodes the QA path and surrounding subgraph of the KB. Their approach achieved a similar F1 score on Stanford's benchmark, which demonstrated the viability of subgraph embeddings.

In this study, we focus on using the open Freebase (Bollacker et al. 2008) as illustrated in Figure 1. We focus on developing a portable system that can be easily scaled from Freebase to other KBs. The WebQuestions dataset contains 5,810 question-answer pairs with common questions asked by web users (Berant et al. 2013). This dataset is built using Freebase as the KB by crawling questions through the Google Suggest API, and then obtaining answers using Amazon Mechanical Turk. The WebQuestions task has been evaluated by many systems as reported by Liang in Figure 2.

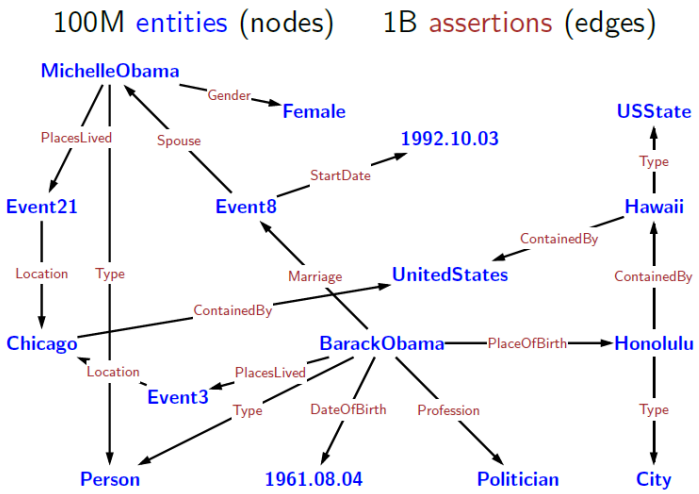

Figure 1. Freebase KB (Bollacker et al. 2008) has over 100m entities and the Stanford system used 41m of them for the WebQuestions benchmark.

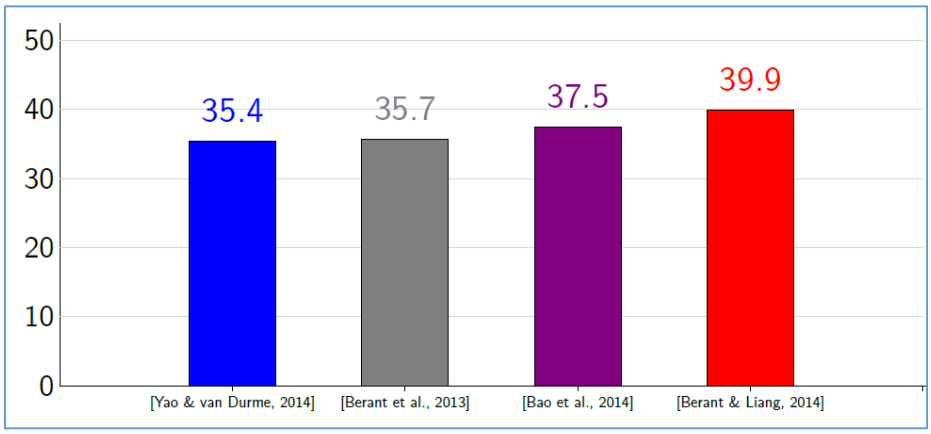

Figure 2. F1 score of several state of the art QA systems on Freebase. Questions include "What did Obama study in school?", "Where to fly into bali?", and "What was tupac name in juice?". (Liang 2014)

Our approach is similar to the Stanford system with a number of key differences. Both of these differences entail gains over the best published Stanford benchmark:

1) Microsoft's proprietary Affinity Intent Map (AIM) is used as one of our key matching and ranking features. AIM was developed to improve Bing's intent relevance (Huang et al. 2012). Shannon's information-theoretic mutual information is used as the objective function to derive embeddings (Pierce 1980). AIM offers semantic embeddings for most frequent words and entities.

2) Like the Stanford system, we also approach semantic parsing from the opposite end, i.e., from the KB rather than from the input question. Instead of parsing the input question into logical form with a weak ontology, our approach has the advantage of utilizing the rich KB facts from the beginning. One potential drawback of Stanford's approach is it may not scale to handle more complicated logic. A logic driven A* decoder is used to enhance the candidate generation. The decoder tries to construct the most probable proof, using the ground facts in the KB, for the input question.

2 Microsoft DeepQA System Architecture

A significant amount of crafting efforts or a sufficient amount of QA training pairs is often needed for most of the modern QA systems. Traditional QA systems such as IBM Watson require dealing with multiple steps: speech recognition or intelligent text input, syntactical parsing, semantic understanding, KB querying, and answer ranking/output. To answer questions such as "What is current US president's brother in law's profession?" as well as many diverse ways to paraphrase such a question like "What job did the brother of the first lady of the United States in 2014 have?", the key task is to understand language's massive variations, some of which may be ambiguous for relevant semantic representation. We also need to combine atomic pieces of knowledge in KB's subject-predicate-object triples to answer many challenging natural language questions. For questions listed here, many KB triples need to be stitched together via logic reasoning as illustrated in Figure 3.

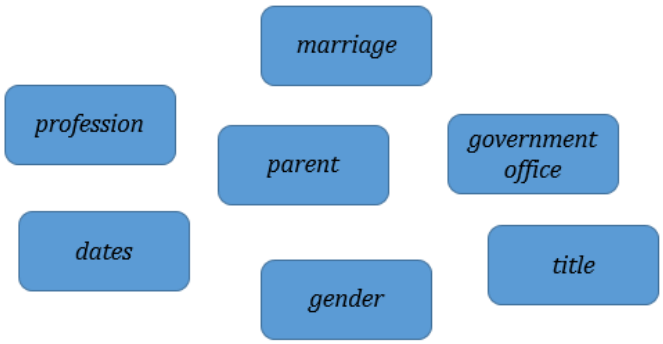

Figure 3. Freebase KB triples needed via logic reasoning to answer "What is current US president's brother in law's profession?" or "What job did the brother of the first lady of the United States in 2014 have?"

These atomic facts often need to go through a proof-theoretic reasoning process to reach the answer that does not preexist directly in the KB (Carpenter 1997). These combinatorial explosion challenges in the parsing or decoding procedure need to be carefully addressed. An efficient A* decoder with a discriminative evaluation function is needed to construct the most probably proof of a canonical utterance to answer the given question from the ground facts.

We borrowed many core concepts adopted in modern speech recognition for our Deep QA system (Huang et al. 2014). We rely on a very similar Bayesian framework to treat many disparate layers (like in a typical speech recognition system) to derive the best parse (A^*) for a given question Q:

$$A^* = \operatorname*{argmax}_A P(A|Q)$$
$$= \operatorname*{argmax}_A P(Q|A)P(A)$$

Here $P(Q|A)$ is the posterior probability generated from the underlying model and $P(A)$ is the prior probability that is similar to the language model used in most spoken language systems. A parse is a proof, and it naturally results in a possible answer of the question.

To effectively manage a massive search space under the Bayesian framework, the system is designed to have a three layered architecture as shown in Figure 4. We adopt the principle to gradually add broader contextual features that are often more expensive to derive. In general, lower layers in the system are

mostly focused on reducing the search space with efficiency and recall as the key objective function. Higher layers use more accurate and contextual features to improve precision.

Facts in the KB need to be utilized much more effectively in comparison to traditional semantic parsing. As in most speech recognition systems, early hard decisions must be avoided to consider obscure yet correct constructs. We need to use the question being asked to guide the knowledge reasoning process and connect (semi-)structured KB with unstructured text (question). The decoder also need to derive knowledge on demand instead of using predetermined templates or the keyword-based web search framework.

One of the most distinctive features in our approach is syntactical parsing, semantic understanding, logic reasoning, and answer ranking are treated in a truly unified manner. A large number of hypotheses in the form of logical formulae are explored from the KB that represent candidate semantic representations of possible answers leading to the given question. For each hypothesis, the given question is parsed and evaluated against it using a semantic scoring function through statistically motivated deep learning.

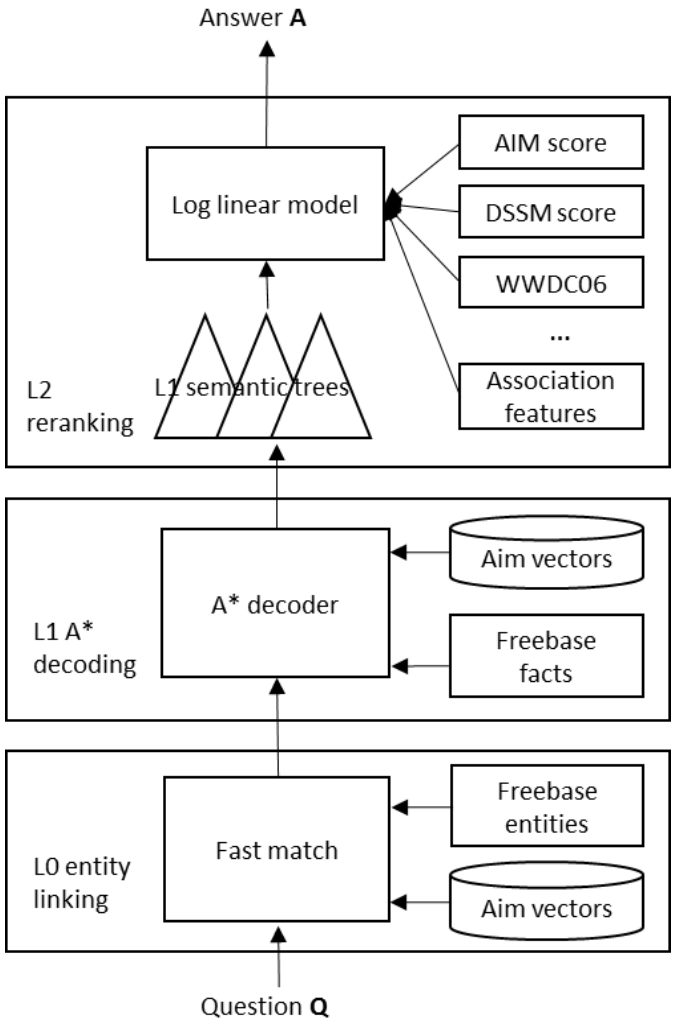

Figure 4. Three layered Microsoft Deep QA System Architecture Diagram

3 Fast Match

The first layer in our system (L0) is to identify a set of possible relevant entities for the input question. It tries to identify top N entities (E) with highest probability $P(E|Q)$ given the input question (Q). These entities are critical for the next level of service (L1) to further recognize the correct answer. The recall

performance is the most important objective function for L0 since it is the gate through which the right entities pass towards subsequent layers. If it misses the right entity, there is no chance that subsequent layers can recover L0 errors. On the other hand, if L0 returns too many entities, it adversely affects precision and latency of downstream components.

The simplest implementation is to use question surface form as our matching features. The naïve approach may miss correct entities implied in the query due to the diversity of natural questions. In a manner similar to Google's word2vec embeddings (Mikolov et al. 2013), we used Microsoft's internal Affinity Intent Map (AIM) that was developed to improve Bing's intent matching. AIM used Shannon's information-theoretic mutual information (Pierce 1980) to derive semantic embeddings for words, sentences and entities. AIM embeddings are used as one of our core features in L0. These AIM features were previously trained with a massive amount of web usage data for Bing to improve web ranking. We did not retrain AIM embeddings for our QA tasks so these results should be reasonably scalable for other open domains.

AIM scores are used to rank entities based on NL parser-identified noun phrases in the input question. AIM-based L0 recall results are listed in Table 1. More than 10pt F1 score recall improvement is observed over the surface form based entity recognition. Similarly, the Stanford system also reported that word2vec embedding can improve Stanford F1 score significantly.

Table 1. L0 recall results in Microsoft Deep QA system with and without AIM features

WebQuestions dev set	L0: surface form only	L0: AIM as features
Recall Accuracy (F1)	66.1	77.3

4 Unified A* Decoding

Our unified framework combines syntactical parsing, semantic understanding, logic reasoning, and answer ranking in one process. A large number of hypotheses in the form of logical formulae are explored from the KB to represent candidate semantic representations of possible answers.

For each hypothesis, the given question is parsed and evaluated using a semantic scoring function. An optimized A* decoder is used to derive n-best candidate answers with the corresponding syntactical and semantic parses for the given input question, i.e. it generates top N candidate parses (A) of the given input question (Q) based on the probability $P(A|Q)$ assigned by the model. The key metrics of this layer are top-one F1 score and oracle F1 score for picking the parse with the best F1 prediction for each n-best list (i.e., each question).

Our decoder is inspired by Stanford's paraphrasing approach (Berant & Liang 2014). Candidate logical forms are generated with corresponding canonical utterances. We also decomposed logical rules into limited amount of atomic pieces that are pieced together to form a myriad of rich logical forms. There are three major differences:

1) While Berant & Liang used only five templates, we took advantage of full-strength proof-theoretic logical programming to form much richer logical forms.
2) While Berant & Liang enumerated candidate logical forms from a candidate entity, which is possible due to limited number of templates, we used an A* decoder to search for arbitrarily complicated logical formula for the input question. Figure 5 shows one possible deduction tree for such an example.
3) While Berant & Liang used Google's word2vec embeddings (Mikolove et al. 2013), we used Microsoft's own maximum mutual information derived AIM embeddings. Stanford system also retrained word2vec embeddings for QA but we relied on domain independent AIM embeddings.

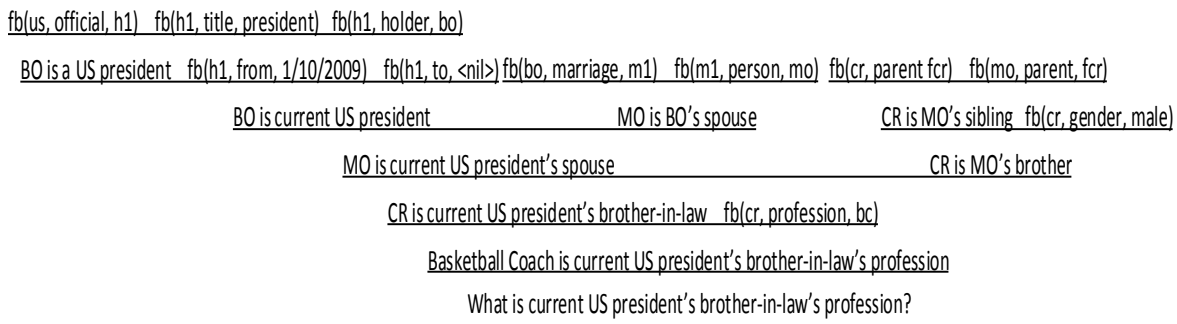

Figure 5. One possible deduction tree for "What is current US president's brother in law's profession?" fb(subject, predicate, object) stands for Freebase fact. For brevity, a few inferences are dropped and the inference rule names are omitted without loss of readability.

In comparison to Stanford's latest results (Berant & Liang 2014), our WebQuestions development set results are shown in Table 2.

Table 2. Microsoft Deep QA and Stanford Berant & Liang 2014 System

WebQuestions dev set	Stanford ACL14 Cand Gen	Microsoft Deep QA L1	Deep QA L1 (surface form L0)
Recall Accuracy (F1)	63.0	77.3	66.1
Top-one Accuracy (F1)	31.3 (JACCARD)	40.8	37.5

Our A* decoder currently used only simple surface-form features as A* decoding evaluation function to handle the massive search space. Without ranking, Stanford results are also shown in the table. With AIM features in L0, we significantly outperformed Stanford's simple candidate generation both in oracle and top-one accuracies. Without AIM features in L0, the results are still measurably better (63.0 vs 66.1).

5 Post Ranking

The post ranking layer (L2) is to utilize more powerful contextual features that are too expensive to use in the A* decoder to further refine candidate answers. Once again, we used the same Bayesian framework to optimize for the best answer.

The L2 ranker is currently implemented as a log-linear model using a large number of lexicalized features such as AIM and DSSM scores (Gao et al. 2014). The parameters of the log linear model is learned using L-BFGS, with L1-regulation to avoid over fitting (Andrew & Gao 2007). By utilizing features such as WDDC06 match (Wan et al. 2006), AIM, and a subset of association model features, the final answer for the given input question is derived.

6 Experimental Results

Microsoft Deep QA system is compared with many other systems in terms of F1 score using the Stanford evaluation script[1]. Cross validation is used to find optimal parameters of our model. Facebook recently considered an ensemble of its approach (Bordes et al. 2014) with the Stanford system (Berant & Liang 2014) to achieve the best reported number of 41.8 on the WebQuestions test set. For the individual system, the Stanford system (Berant & Liang 2014) still has the best F1 score (39.9) among all systems reported. Table 3 illustrates our Deep QA results (45.3) in comparison to these state of the art QA systems. Interestingly, our n-best F1 score is approaching 77.3 in comparison to our top 1 F1 score of 45.3. There is a very significant room for future improvement.

[1] Available from www-nlp.stanford.edu/software/sempre/

Table 3. Results on the WebQuestions test set using Berant & Liang's F1 evaluation method. Other systems were reported by (Bordes et al 2014)

Method	F1 Score (Berant & Liang 2014)
Stanford (Berant et al. 2013)	31.4
Borders et al. 2013	29.7
Fader et al. 2014	35.0
Yao and Van Durme 2014	33.0
Stanford (Berant & Liang 2014)	**39.9**
Facebook (Bordes et al. 2014)	**39.2**
Ensemble of Facebook & Stanford (Borders et al. 2014)	41.8
Our Approach: Microsoft Deep QA	**45.3**
Our Approach: Microsoft Deep QA N-best from L1	(Upper bound) 77.3

7 Conclusion

This paper presented a new unified framework motivated by both statistically-driven information theoretic mutual information embeddings and logically-driven proof-theoretic semantics to address one of the most challenging QA tasks for open-domain and open-topic QA. Our Deep QA system provides a parsed structure and proof tree among answers, and can achieve very encouraging performance on the competitive Stanford benchmark WebQuestions.

Acknowledgement

We thank Guihong Cao, David Ku, Yi Li, Qi Lu, Jan Pederson, Chris Quirk, Jordi Ribas, Harry Shum, Jinxi Xu, and Zijian Zheng for providing web-related data, tools, support, and computing resources that enabled Deep QA and AIM R&D. We also appreciate the support and collaboration with many colleagues: Jianfeng Gao for providing us with the DSSM feature, Zheng Chen, Gavin Hu & Jun Yan for RNN R&D, Jianwen Zhang for joint knowledge and text embedding, Nan Duan & Ming Zhou for the QA framework and NLP utilities. Last but not least, Zheng Chen, JP Martin, Jiaping Wang, Lingfeng Wu, Wayne Xiong, Gu Xu, Jun Yan, Steven Yao, Yuan Yu, and Michael Zeng played a key role to help our internal AIM embeddings used in this study.

References

1. G. Andrew, J. Gao, Scalable training of L_1-regularized log-linear models. In *ICML*, 2007
2. J. Bao, N. Duan, M. Zhou, T. Zhao, Knowledge-based Question Answering as Machine Translation, In *Proceedings of the 52nd Annual Meeting of the ACL*, 2014
3. J. Berant, A. Chou, R. Frostig, and P. Liang. Semantic parsing on Freebase from question-answer pairs. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, October 2013.
4. J. Berant and P. Liang. Semantic parsing via paraphrasing. In *Proceedings of the 52nd Annual Meeting of the ACL*, 2014.
5. K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*. ACM, 2008.
6. A. Bordes, S Chopra, and J Weston, Question Answering with Subgraph Embeddings, In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, 2014
7. B. Carpenter. Type-Logical Semantics, MIT Press, 1997.
8. Q. Cai and A. Yates. Large-scale semantic parsing via schema matching and lexicon extension. In Association for Computational Linguistics (ACL) 2013.

9. A. Fader, L. Zettlemoyer, and O. Etzioni. Paraphrase-driven learning for open question answering. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pages 1608–1618, Sofia, Bulgaria, 2013.
10. A. Fader, L. Zettlemoyer, and O. Etzioni. Open question answering over curated and extracted knowledge bases. In *Proceedings of KDD'14*. ACM, 2014.
11. Ferrucci, D, et al., Building Watson: An Overview of the DeepQA Project, AI Magazine (AI Magazine) 31 (3), 2010.
12. J. Gao, P. Pantel, M. Gamon, X. He, L. Deng, Y. Shen. 2014. Modeling interestingness with deep neural networks. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, Oct, 2014.
13. X. Huang, et al. Project Affinity Intent Map (AIM), Microsoft Internal Memo, 2012
14. X. Huang, J. Baker, R. Reddy, A Historical Perspective of Speech Recognition, Communications of the ACM, 57 (1), 94-103, 2014
15. O. Kolomiyets and M.-F. Moens. A survey on question answering technology from an information retrieval perspective. *Information Sciences*, 181(24):5412–5434, 2011.
16. T. Kwiatkowski, E. Choi, Y. Artzi, and L. Zettlemoyer. Scaling semantic parsers with on-the-fly ontology matching. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, October 2013.
17. P. Liang, Computing with Natural Language, In *ACL 2014 Workshop on Semantic Parsing*, 2014
18. T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient Estimation of Word Representations in Vector Space. In *Proceedings of Workshop at ICLR*, 2013
19. R. Mooney, Semantic Parsing, Past, Present and Future, In *ACL 2014 Workshop on Semantic Parsing*, 2014
20. J. Pierce, An Introduction to Information Theory: Symbols, Signals and Noise, Dover Books on Mathematics, 1980
21. M. Steedman, Robust Semantics of Semantic Parsing, In *ACL 2014 Workshop on Semantic Parsing*, 2014
22. S. Wan, M. Dras, R. Dale, and C. Paris. Using dependency-based features to take the "para-farce" out of paraphrase. In Australasian Language Technology Workshop. 2006
23. X. Yao and B. Van Durme. Information extraction over structured data: Question answering with freebase. In *Proceedings of the 52nd Annual Meeting of the ACL*, 2014.