A Semantic Parsing and Reasoning-Based Approach to Knowledge Base Question Answering

Ibrahim Abdelaziz*, **Srinivas Ravishankar***, **Pavan Kapanipathi***, **Salim Roukos**, **Alexander Gray**
IBM Research
IBM T.J. Watson Research Center
Yorktown Heights, NY, USA
{ibrahim.abdelaziz1, srini, alexander.gray}@ibm.com, {kapanipa, roukos}@us.ibm.com

Abstract

Knowledge Base Question Answering (KBQA) is a task where existing techniques have faced significant challenges, such as the need for complex question understanding, reasoning, and large training datasets. In this work, we demonstrate Deep Thinking Question Answering (`DTQA`), a semantic parsing and reasoning-based KBQA system. `DTQA` (1) integrates multiple, reusable modules that are trained specifically for their individual tasks (e.g. semantic parsing, entity linking, and relationship linking), eliminating the need for end-to-end KBQA training data; (2) leverages semantic parsing and a reasoner for improved question understanding. `DTQA` is a system of systems that achieves state-of-the-art performance on two popular KBQA datasets.

Introduction

The goal of Knowledge Base Question Answering is to answer natural language questions over Knowledge Bases (KB) such as DBpedia (Auer et al. 2007) and Wikidata (Vrandečić and Krötzsch 2014). Existing approaches to KBQA are either (i) graph-driven (Diefenbach et al. 2020; Vakulenko et al. 2019), or (ii) end-to-end learning (Sun et al. 2020) approaches. Graph-driven approaches attempt to find a best-match KB sub-graph for the given question, whereas end-to-end learning approaches, requiring large amounts of training data, directly predict a query sequence (i.e. SPARQL or SQL) from the input question. Both categories still suffer from a number of challenges, including complex question understanding, scarcity of end-to-end training data, and the need for reasoning.

In this work, we demonstrate `DTQA`; a semantic-parsing and reasoning-based KBQA system. `DTQA` begins by parsing the natural language question into Abstract Meaning Representation (AMR), which is a symbolic formalism that captures rich semantic information from natural language. The use of AMR enables task-independent language understanding, which `DTQA` leverages to answer structurally complex questions. Next, our system uses a triples-based approach that builds on state-of-the-art entity and relation linking modules to align the AMR graph with the underlying KB and produce logic forms. This step accurately maps AMR and its associated PropBank information to a KB such as DBpedia. Finally, the logic forms are used by Logical Neural Network (LNN) (Riegel et al. 2020), a neuro-symbolic reasoner that enables `DTQA` to perform different types of reasoning. `DTQA` comprises several modules which operate within a pipeline and are individually trained for their specific tasks.

This demonstration shows an interpretable methodology for KBQA that achieves state-of-the-art results on two prominent datasets (QALD-9, LC-QuAD-1). `DTQA` is a system of systems that demonstrates: (1) the use of task-general semantic parsing via AMR; (2) an approach that aligns AMR to KB triples; and (3) the use of a neuro-symbolic reasoner for the KBQA task.

`DTQA` System

Figure 1 shows an overview of `DTQA` with an example question. The input to `DTQA` is the question text, which is first parsed using the Abstract Meaning Representation (AMR) parser (Banarescu et al. 2013; Dorr, Habash, and Traum 1998). AMR is a semantic parse representation that solves the ambiguity of natural language by representing syntactically different sentences with the same underlying meaning in the same way. An AMR parse of a sentence is a rooted, directed, acyclic graph expressing "who is doing what to whom". We use the current state-of-the-art system for AMR that leverage transition-based (Naseem et al. 2019) parsing approach, parameterized with neural networks and trained in an end-to-end fashion. AMR parses are KB independent, however an essential task to solve KBQA using AMR is to align the AMR parses with the knowledge base. Therefore, in the next three modules we align the AMR parse to a DBpedia sub-graph that can be transformed into a SPARQL query.

First, `DTQA` uses an Entity Extraction and Linking (EL) module that extracts entities and types and links them to their corresponding KB URIs. Given a question such as *Which actors starred in Spanish movies produced by Benicio del Toro?*, the EL module will link the AMR nodes of *Benicio del Toro*, and *Spanish* to *dbr:Benicio_del_Toro* and *dbr:Spain*, respectively. In `DTQA`, we trained a BERT-based neural mention detection and used BLINK (Devlin et al. 2018) for disambiguation of named entities.

Aligning the AMR graph to the underlying KB is a challenge, particularly due to structural and granularity mismatch between AMR and KBs such as DBPedia. The PropBank

*Equal contribution

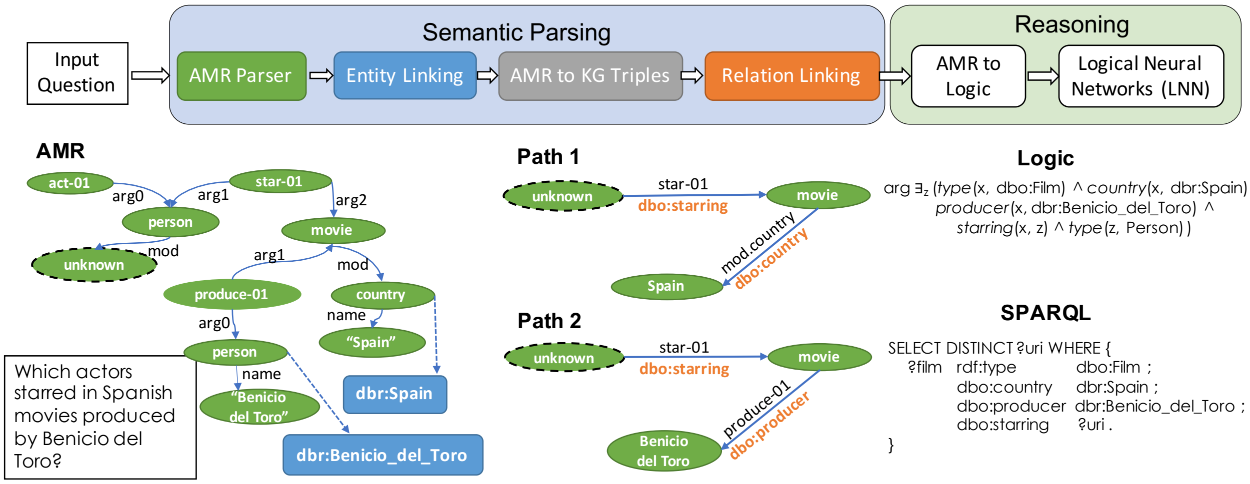

Figure 1: DTQA architecture, illustrated via examples

frames in AMR are n-ary predicates whereas relationships in KBs are predominantly binary. In order to transform the n-ary predicates to binary, we use a novel 'Path-based Triple Generation' module that transforms the AMR graph (with linked entities) into a set of triples where each has one to one correspondence with constraints in the SPARQL query. The main idea of this module is to find paths between the question amr-unknown to every linked entity and hence avoid generating unnecessary triples. For the example in Figure 1, this approach will generate the following two paths:

1. *amr-unknown* $\rightarrow$ *star-01* $\rightarrow$ *movie* $\rightarrow$ *country* $\rightarrow$ *Spain*
2. *amr-unknown* $\rightarrow$ *star-01* $\rightarrow$ *movie* $\rightarrow$ *produce-01* $\rightarrow$ *Benicio del Toro*.

Note that the predicate *act-01* in the AMR graph is irrelevant, since it is not on the path between *amr-unknown* and the entities, and hence ignored.

Furthermore, AMR parse provides relationship information that is generally more fine-grained than the KB. To address this granularity mismatch, the path-based triple generation collapses multiple predicates occurring consecutively into a single edge. For example, in the question "In which ancient empire could you pay with cocoa beans? ", the path consists of three predicates between *amr-unknown* and *cocoa bean*; (location, pay-01, instrument). These are collapsed into a single edge, and the result is *amr-unknown* $\rightarrow$ *location* | *pay-01*| *instrument* $\rightarrow$ *cocoa-bean*.

Next, the (collapsed) AMR predicates from the triples are mapped to relationships in the KB using the Relation Extraction and Linking (REL) module. As seen in Figure 1, the REL module maps *star-01* and *produce-01* to DBpedia relations *dbo:starring* and *dbo:producer*, respectively. For this task, DTQA uses SLING (Mihindukulasooriya et al. 2020); the state-of-the-art REL system for KBQA. SLING takes as input the AMR graph with linked entities and extracted paths obtained from the upstream modules. It relies on an ensemble of complementary relation linking approaches to produce a top-k list of KB relations for each AMR predicate. The most relevant and novel approach in SLING is its statistical mapping of an AMR predicate to its semantically equivalent relation from the underlying KB using a distant supervision dataset. Apart from this statistical co-occurrence information, the rich lexical information in the AMR predicate and question text is also leveraged by SLING. This is captured by a neural-model and an unsupervised semantic similarity module that measures similarity of each AMR predicate with a list of possible relations using GloVe (Pennington, Socher, and Manning 2014) embeddings.

The AMR to Logic module is a rule-based system that transforms the KB-aligned AMR paths to a logical formalism to support binary predicates and higher-order functional predicates (e.g. for aggregation and manipulation of sets). This logic formalism has a deterministic mapping to SPARQL by introducing SPARQL constructs such as COUNT, ORDER BY, and FILTER.

Some questions in KBQA datasets require reasoning. However, existing KBQA systems (Diefenbach et al. 2020; Vakulenko et al. 2019; Sun et al. 2020) do not have a modular reasoner to perform logical reasoning using the information available in the knowledge base. In DTQA, the final module is a logical reasoner called Logical Neural Networks (LNN) (Riegel et al. 2020). LNN is a neural network architecture in which neurons model a rigorously defined notion of weighted fuzzy or classical first-order logic. LNN takes as input the logic forms from AMR to Logic and has access to the KB to provide an answer. LNN retrieves predicate groundings via its granular SPARQL integration and performs type-based, and geographic reasoning based on the ontological information available in the knowledge base. LNN allowed us to manually add generic axioms for the geographic reasoning scenarios.

Experimental Evaluation

To evaluate DTQA, we use two prominent KBQA datasets; QALD - 9 (Ngomo 2018) and LC-QuAD 1.0 (Trivedi et al. 2017). The latest version of QALD-9 contains 150 test questions whereas LC-QuAD test set contains 1,000 questions.

	Dataset	P	R	F	F1 QALD
WDAqua	QALD	26.09	26.7	24.99	28.87
gAnswer	QALD	29.34	32.68	29.81	42.96
DTQA	QALD	31.41	32.16	**30.88**	**45.33**
WDAqua	LC-QuAD	22.00	38.00	28.00	–
QAMP	LC-QuAD	25.00	50.00	33.00	–
DTQA	LC-QuAD	33.94	34.99	**33.72**	–

Table 1: DTQA performance on QALD-9 and LC-QuAD 1.0

We compare DTQA against three KBQA systems: 1) GAnswer (Zou et al. 2014) is a graph-driven approach and the state-of-the-art on QALD dataset, 2) QAmp (Vakulenko et al. 2019) is another graph-driven approach based on message passing. QAmp is also state-of-the-art on LC-QuAD dataset, and 3) WSDAqua-core1 (Diefenbach et al. 2020) which to the best of our knowledge, is the only approach evaluated on both QALD and LC-QuAD.

Results: Table 1 shows the performance of DTQA on QALD and LC-QuAD datasets. It shows the standard precision, recall and F-score metrics for each system. We also report Macro-F1-QALD (Usbeck et al. 2015), a recommended metric to use for evaluating QALD dataset. DTQA achieves state-of-the-art performance on all metrics, outperforming existing graph-driven approaches such as gAnswer, WDAqua, and QAmp on both datasets.

By utilizing AMR to get the semantic representation of the question, DTQA is able to generalize to sentence structures that come from very different distributions, and achieve state-of-the-art performance on both datasets. DTQA enables opportunities for reasoning by transforming text to a logic form that can be used by LNN (a neuro-symbolic reasoner).

Demonstration and Conclusion

DTQA is a system of systems trained for different sub-problems that has achieved the state-of-the art performance on two prominent KBQA datasets. The system demonstrates an interpretable methodology for understanding the natural language question and reasoning to infer the answer from the KB. To the best of our knowledge, DTQA is the first system that successfully harnesses a generic semantic parser with a neuro-symbolic reasoner for a KBQA task and a novel path-based approach to map AMR to the underlying KG.

References

Auer, S.; Bizer, C.; Kobilarov, G.; Lehmann, J.; Cyganiak, R.; and Ives, Z. 2007. Dbpedia: A nucleus for a web of open data. In *The semantic web*, 722–735. Springer.

Banarescu, L.; Bonial, C.; Cai, S.; Georgescu, M.; Griffitt, K.; Hermjakob, U.; Knight, K.; Koehn, P.; Palmer, M.; and Schneider, N. 2013. Abstract meaning representation for sembanking. In *Proceedings of the 7th linguistic annotation workshop and interoperability with discourse*, 178–186.

Devlin, J.; Chang, M.; Lee, K.; and Toutanova, K. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *CoRR* abs/1810.04805. URL http://arxiv.org/abs/1810.04805.

Diefenbach, D.; Both, A.; Singh, K.; and Maret, P. 2020. Towards a question answering system over the semantic web. *Semantic Web* (Preprint): 1–19.

Dorr, B.; Habash, N.; and Traum, D. 1998. A thematic hierarchy for efficient generation from lexical-conceptual structure. In *Conference of the Association for Machine Translation in the Americas*, 333–343. Springer.

Mihindukulasooriya, N.; Rossiello, G.; Kapanipathi, P.; Abdelaziz, I.; Ravishankar, S.; Yu, M.; Gliozzo, A.; Roukos, S.; and Gray, A. 2020. Leveraging Semantic Parsing for Relation Linking over Knowledge Bases. In *Proceedings of the 19th International Semantic Web Conference (ISWC2020)*.

Naseem, T.; Shah, A.; Wan, H.; Florian, R.; Roukos, S.; and Ballesteros, M. 2019. Rewarding Smatch: Transition-Based AMR Parsing with Reinforcement Learning. *arXiv preprint arXiv:1905.13370* .

Ngomo, N. 2018. 9th challenge on question answering over linked data (QALD-9). *language* 7(1).

Pennington, J.; Socher, R.; and Manning, C. D. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 1532–1543.

Riegel, R.; Gray, A.; Luus, F.; Khan, N.; Makondo, N.; Akhalwaya, I. Y.; Qian, H.; Fagin, R.; Barahona, F.; Sharma, U.; Ikbal, S.; Karanam, H.; Neelam, S.; Likhyani, A.; and Srivastava, S. 2020. Logical Neural Networks.

Sun, H.; Arnold, A. O.; Bedrax-Weiss, T.; Pereira, F.; and Cohen, W. W. 2020. Faithful Embeddings for Knowledge Base Queries.

Trivedi, P.; Maheshwari, G.; Dubey, M.; and Lehmann, J. 2017. Lc-quad: A corpus for complex question answering over knowledge graphs. In *International Semantic Web Conference*, 210–218. Springer.

Usbeck, R.; Röder, M.; Ngonga Ngomo, A.-C.; Baron, C.; Both, A.; Brümmer, M.; Ceccarelli, D.; Cornolti, M.; Cherix, D.; Eickmann, B.; et al. 2015. GERBIL: general entity annotator benchmarking framework. In *Proceedings of the 24th international conference on World Wide Web*, 1133–1143.

Vakulenko, S.; Fernandez Garcia, J. D.; Polleres, A.; de Rijke, M.; and Cochez, M. 2019. Message passing for complex question answering over knowledge graphs. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, 1431–1440.

Vrandečić, D.; and Krötzsch, M. 2014. Wikidata: a free collaborative knowledgebase. *Communications of the ACM* 57(10): 78–85.

Zou, L.; Huang, R.; Wang, H.; Yu, J. X.; He, W.; and Zhao, D. 2014. Natural language question answering over RDF: a graph data driven approach. In *Proceedings of the 2014 ACM SIGMOD international conference on Management of data*, 313–324.