Generating Simple Language-Based Templates from a Knowledge Graph to Completely Cover its Question-Answer Space.

Alex Gagnon
alex.gagnon@carleton.ca
Carleton University
Ottawa, Ontario

CCS Concepts • **Information systems** → *Information integration*; *Data mining*; **Web data description languages**; **Information retrieval**; • **Computing methodologies** → **Natural language processing**.

Keywords datasets, information retrieval, data description languages

ACM Reference Format:
Alex Gagnon. 2020. Generating Simple Language-Based Templates from a Knowledge Graph to Completely Cover its Question-Answer Space.. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

Introduction

The domain of Question-Answering boils down to an inconsistency between the unstructured natural language based question and the structured schema of the knowledge base that contains the answer. Foe example, given the question 'where was Michael Jordan born?', a system is expected to be able to return the answer 'Brooklyn, New York', in a time manner. The concept of question and answering itself is vital for two reasons. First, a store containing potentially the entirety of collected human knowledge is, by itself, useless. A mechanism to extract facts in a generic manner, such as through voice or written text, is essential to give meaning to the endeavor. Secondly, as more information is collected over time, the search space containing the possible answers becomes enormous. A strategy for question answering that performs quickly and accurately at scale, and ideally in near-real time, is the desired outcome for most applications.

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

When a question is asked in human language, a processing step must convert its semantics into a formal query suitable to be run against a datastore such as DBpedia, Freebase, and Wikidata. The information in these stores is represented by a graph containing facts in the form of subject/predicate/object triples, known as RDF (e.g. "(Michael Jordan, bornIn, Brooklyn NY)"). The primary mechanism for accessing these knowledge graphs is through specialized query languages (e.g. SPARQL), that traverse the graph and retrieve triples matching the request.

A failure to convert the question into the appropriate formal query will lead to incorrect answers. The likelihood of an erroneous conversion increases as the question becomes more complex. This can occur in several situations. Firstly, when a question is composed of multiple clauses (i.e. a conjuctive "and" or through nested questions where the answer of one clause is used sequentially in the next). Secondly, the intricacies of the language itself can cause ambiguities, such as when pronoun usage makes the subject difficult or impossible to identify (e.g. 'John has a son named Tom. He went to Harvard'.)

There are two standard approaches to solving this conversion problem: semantic parsing, and templates.

Semantic Parsing. Semantic parsing deconstructs the question into one or more subgraphs based on the grammatical and syntactical structure of the sentence, and then attempts to find matches in the knowledge graph. This strategy can be effective as it is, in essence, 'real-time' and does not require a large bank of previously computed examples. However, it is prone to incorrect subgraph creation. This is due to the fact that neural networks are typically used, and these require a large and diverse set of training data to account for all topologies of subgraphs that exist in the knowledge base.

Templates. Template-based approaches instead try to convert the question into one or more simpler questions, of which an equivalent structured query has already been generated. For example, a question such as 'what is the name of the person who has won the most NBA MVP awards ever' into 'who won the most NBA MVP awards'. This simpler question is directly mapped to a query pattern: '?award,

wonBy, <Person>'. Templates have been found to be effective in returning high quality answers in a timely fashion, however they often suffer in coverage. Zheng et al. were able to outperform state-of-the-art implementations using binary templates. However, the source of the natural language templates was gathered through examining a single text corpus. As such, it suffered a lack of generalizability to other sources. It also uses neural networks to create the simplified questions, which, as previously mentioned, depends on having accurate and diverse training data which can be limiting given that only a sole source in a specific domain is used.

We seek to address the issues of limited coverage for template-based approaches by instead generating natural language/query pattern pairs by starting from the knowledge base itself. By traversing the triple graph, we can effectively create simple questions that map to the current fact. For example, given the triple (Michael Jordan, bornIn, Brooklyn), we can create simple questions to represent it such as 'where was Michael Jordan born', and its corresponding SPARQL queries 'Michael Jordan, bornIn, <City>'. Other more general questions stemming from this fact can also be produced, such as 'who is Michael Jordan', and 'where is Brooklyn', which further complete the search space. The use of question words (e.g. when, where, who, what, why, how) can be injected based on the types of entities and classification of predicates found in the fact. Similar 'bottom-up' approaches of starting from the knowledge base itself include work by Serban et al., where a large corpora of question-answer pairs was created. The goal in this case, however, was to produce question-answer pairs for use in benchmarking and not for use as templates.

Difficulties in the approach include ambiguities and laxness in the language (i.e. where was <Person> born, in which city was <Person> born), and the sheer volume of templates generated. The vast number of which can be generated is orders of magnitude larger than the number of facts in the knowledge base, and as such, requires indexing/space reduction methodologies such as those in the implementation by Zheng et al.. We investigate indexing and filtering strategies to reduce the space complexity produced and to ensure the natural language templates can be matched in a timely manner.

We will not be investigating the actual conversion of a natural language question into its corresponding set of simple questions, however, for more information on the topic see [1, 6]

References

[1] Abdalghani Abujabal, Mirek Riedewald, and Gerhard Weikum. 2017. Automated Template Generation for Question Answering over Knowledge Graphs. 1191–1200. https://doi.org/10.1145/3038912.3052583

[2] Nikita Bhutani, Xinyi Zheng, and H V Jagadish. 2019. Learning to Answer Complex Questions over Knowledge Bases with Query Composition. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management (CIKM '19)*. Association for Computing Machinery, New York, NY, USA, 739–748. https://doi.org/10.1145/3357384.3358033

[3] Jiwei Ding, Wei Hu, Qixin Xu, and Yuzhong Qu. 2019. Leveraging Frequent Query Substructures to Generate Formal Queries for Complex Question Answering. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Association for Computational Linguistics, Hong Kong, China, 2614–2622. https://doi.org/10.18653/v1/D19-1263

[4] Iulian Serban, Alberto García-Durán, Caglar Gulcehre, Sungjin Ahn, Sarath Chandar, Aaron Courville, and Y. Bengio. 2016. Generating Factoid Questions With Recurrent Neural Networks: The 30M Factoid Question-Answer Corpus. 588–598. https://doi.org/10.18653/v1/P16-1056

[5] Weiguo Zheng, Jeffrey Yu, Lei Zou, and Hong Cheng. 2018. Question answering over knowledge graphs: question understanding via template decomposition. *Proceedings of the VLDB Endowment* 11 (07 2018), 1373–1386. https://doi.org/10.14778/3236187.3236192

[6] Weiguo Zheng, Lei Zou, Xiang Lian, Jeffrey Xu Yu, Shaoxu Song, and Dongyan Zhao. 2015. How to Build Templates for RDF Question/Answering: An Uncertain Graph Similarity Join Approach. In *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data (SIGMOD '15)*. Association for Computing Machinery, New York, NY, USA, 1809–1824. https://doi.org/10.1145/2723372.2747648