Leveraging Graph Databases and Natural Langauge Processing for Explanation Regeneration Tasks

Joshua Salit

MSDS 459: Knowledge Engineering Assignment 4

November 22nd, 2020

Abstract

Question-answering systems application across industry is widespread however there are still aspects of this technology lagging profound success. Identifying explanations to the answer to each question proves to be a challenging task. This requires identifying all relevant information often not included in the initial question. The WorldTree corpus of over 2,000 elementary school science questions provides questions, answers, and facts to facilitate explanation regeneration. This research explores many techniques including various Natural Language Processing methods and graph database storage mechanisms to determine the most important explanations for each question.

Introduction

Question-answering systems relevance extends across industries with many successful applications built today. These chatbot-type interfaces boil-down to simply finding the best response to a question. While many have proven successful, explaining the answer to a question adds a level of complexity that has not shared the same achievement. This type of inference requires connecting and “hopping” across data to identify the most relevant information to explain an answer. Multiple approaches are available to traverse networks of data to retrieve and rank information. In this paper, I explore multi-hop inference for explanation regeneration of an elementary school science question corpus with a question-answering chatbox to interact with.

Chatbots operate for a range of purposes but the root foundation develops from a user’s ability to interact with a system and receive a response. The process begins with the system collecting some information from the user, understanding what it represents, querying a database of knowledge, and returning a response. This research implements the architecture with a couple of environments including Python, Neo4j, and a basic terminal. Multiple NLP techniques were leveraged for understanding the data and scoring responses. TFIDF and word embeddings scores identified the most relevant facts for each question. Reranking TFIDF scores was performed by calculating cosine similarity distances of pre-trained embeddings. The motivation was driven by the need to regularize biases in words from a relatively small corpus particularly addressing generic terms. Graph databases through Neo4j represented the storage infrastructure for the data making the retrieval process efficient and straightforward.

This project is based on the 2019 TextGraphs conference where the attendees were able to submit an attempt at the explanation regeneration task on the WorldTree corpus. The data includes over 2,000 elementary school science questions with metadata, annotations, and facts for additional information. The goal is given a question and answer, rank a set of facts to then

compare against the “gold” correct explanations provided. This paper focuses on using solely the question to fuel the responses which have proven to be a more difficult task.

Literature Review

A good summary and implementation of multi-hop inference can be seen from the Jensen paper working with the WordTree dataset for a question answering application (Zhengnan et al., 2020). Graphs can be leveraged where nodes represent facts or sentences and edges represent some connection between the facts (Jensen 2018). This basic construct provides a solid structure but could be expanded on through supplementing global dimensions developing clusters (Chang et al. 2018). Relationships for information retrieval tasks can be built based on co-occurrence when two words co-occur within a sentence (Zhang et al. 2018). This model could be improved with dynamic edges weighted by the number of entities sentences could share (Mesgar and Strube 2014). The benefits of graph structures with information retrieval are the ability to identify connections that not have been previously understood. (Luo et al. 2018). Other more advanced methods for chatbox applications utilize sophisticated NLP techniques such as BERT enhancing the power of the knowledge base (Das et al. 2019). Question-answering systems are used widely across industry and academia with many techniques available including NLP and methods of storing information.

Methods

Question-answering systems require many tools to facilitate interaction. As summarized, previously, a user interacts with the system to generate a question, the question is parsed and processed, a query is performed to the graph database to return results, the results are processed, and a response is provided back to the user. A walkthrough of the process can be seen in Figure 1 and the architecture for the question-answering system in Figure 2 in the appendix.

The system begins in a terminal with the user engaging in a conversation and asking whether they would like to generate a random question or provide a new question. Next, the user decides whether they would like a score of the question to evaluate the model or a paragraph of the top facts. Once the inputs are provided, the response is parsed and processed using various Python NLP tools to understand the question better. Irrelevant words are removed and the query to the database finds all term connections with the response as facts their ids.

The graph database represents a knowledge base of all facts and metadata information from the raw data. Graphs can be broken down into vertices and edges where the vertex or node represents an entity and the edge as a relationship that connects nodes. With this corpus, each fact was stored as a node, and the relationships are established by shared terms from each fact. This creates a network of information and is queried using information retrieval to find words that exist as facts. Facts were used to identify important terms and storing a sentence form into the database. The process involved tokenizing and cleaning terms and stored as lists and full coherent sentences. Neo4j's admin tool then uploaded the facts and relationships to create the database.

Now that there is a knowledge base of information, the queried facts needed to be ranked for identifying the most relevant based on the question. Two methods used for scoring facts were TFIDF (Term Frequency Inverse Document Frequency) and word embeddings. TFIDF is a technique that scores individual terms based on the frequency they occur in that document and the inverse of the frequency across documents. This intuitive logic can help determine the most important terms in each document. To enhance this process, TFIDF was reranked using pre-trained word embeddings through a target regularization of generic terms. Cosine similarity scores were calculated between each term and a random selection of terms from the pre-trained embedding. Higher scores often associated with generic terms so each penalized the TFIDF score by subtracting. In addition to TFIDF, custom word embeddings were trained on the corpus and cosine similarity was calculated between each term of the question and term in

the fact. Finally, TFIDF and cosine similarity scores were aggregated for each fact and ranked to determine the most relevant facts for each question.

Results

Results were not extremely compelling pointing to the difficulty with multi-hop inference tasks. Success was measured by the amount of “gold” explanation facts in the resulting top 20 facts ranked by the system. A significant part of the shortcoming could be attributed to the minimal amount of information obtained from questions to identify all relevant facts. Utilizing only shared words limited the amount of data to be traversed. If the question did not contain a term in the gold standard explanations, then it would not have been retrieved from the query.

Limitations of only shared words show the importance of developing a more robust understanding of the question and a more powerful network. Applying global spatial dimensions to the network could have helped identify relevant facts that were not included in the original question. This would enable facts to be obtained based on a hierarchy of information. For example, if the question asked about “dogs” then a more higher-level term such as “animals” could be used to gather a larger pool of facts. Another issue with relying on shared words is that all facts retrieved that contain that word so the system depends heavily on the scoring mechanism to filter out facts accurately. Questions with more generic terms run into this issue more frequently.

Word embeddings contributed greatly to the information retrieval task to rerank TFIDF scores and find similarities among individual terms from the question and each fact. One of the initial observations from TFIDF was that it was classifying many generic terms such as “can” or “should” as important terms yet from a human they could easily be seen as throw-out words not proving much value to understanding a question. Pre-trained word embeddings with cosine similarity scores helped identify these more generic terms and then reducing the impact of terms by how generic the terms were. This approach was not perfect as some other broader terms

such as “Technology” were penalized even though there is a value to them. Overall, the pre-trained embeddings helped score the TFIDF terms more appropriately.

Another contribution of word embeddings was from training the corpus with Word2Vec to develop word embeddings. This enabled words to be evaluated based on a vector of information more easily comparably to terms. This formed a network where question terms could search in the vector space to find the similarities. Facts were scored based on the similarity of each term to each term in the question. Combining this score and TFIDF helped provide an ensemble of values to determine the most important facts.

Graph databases proved to be very valuable in implementing this question-answering system. Facts could be easily connected and traversed to obtain desired ones from each query. The speed at which these facts could be queried showed the power of the system. With each question, question terms were sent to the database and iterated through individually gathering all of the facts. The query was executed extremely fast obtaining a host of information. Future work would be focused on enhancing the database of information and applying more queries to gather information and relationships.

Conclusion

Multi-hop inference has proven to be a challenging task because of the need to identify multiple correct responses across a knowledge base. The infrastructure demonstrated here provides a solid foundation for implementing this type of task. Graph databases with Neo4j make it easy to store and query data to represent relationships. Information retrieval utilizing TFIDF and word embeddings help determine and score important terms. There remain many other tools to use to improve a question-answering system. This includes incorporating global spatial dimensions to cluster data and more powerful NLP techniques such as BERT. Explaining elementary school science questions is still a trivial task for systems but the ability to master this will improve question-answering systems broadly.

References

Chang, Haw-Shiuan, Amol Agrawal, Ananya Ganesh, Anirudha Desai, Vinayak Mathur, Alfred Hough, and Andrew McCallum. “Efficient Graph-Based Word Sense Induction by Distributional Inclusion Vector Embeddings.” *arXiv.org*, (2018). https://arxiv.org/abs/1804.03257.

Das, Rajarshi, Ameya Godbole, Manzil Zaheer, Shehzaad Dhuliawala, and Andrew McCallum. “Chains-of-Reasoning at TextGraphs 2019 Shared Task: Reasoning over Chains of Facts for Explainable Multi-Hop Inference.” *ACL Anthology*, (2019). https://www.aclweb.org/anthology/D19-5313/.

Jansen, Peter. “Multi-Hop Inference for Sentence-Level TextGraphs: How Challenging Is Meaningfully Combining Information for Science Question Answering?” *ACL Anthology*, (2018). https://www.aclweb.org/anthology/W18-1703/.

Jansen, Wainwright, Marmorstein, and Morrison (2018). Worldtree: A corpus of Explanation Graphs for Elementary Science Questions supporting Multi-hop Inference. *Proceedings of the Language Resource and Evaluation Conference* (2018).

Luo, Fan, Marco A. Valenzuela-Escárcega, Gus Hahn-Powell, and Mihai Surdeanu. “Scientific Discovery as Link Prediction in Influence and Citation Graphs.” *ACL Anthology*, (2018). https://www.aclweb.org/anthology/W18-1701/.

Mesgar, Mohsen, and Michael Strube. “Normalized Entity Graph for Computing Local Coherence.” *ACL Anthology*, (2014). https://www.aclweb.org/anthology/W14-3701/.

Xie, Zhengnan, Sebastian Thiem, Jaycie Martin, Elizabeth Wainwright, Steven Marmorstein, and Peter Jansen. “WorldTree V2: A Corpus of Science-Domain Structured Explanations and Inference Patterns Supporting Multi-Hop Inference.” *ACL Anthology*, (2020). https://www.aclweb.org/anthology/2020.lrec-1.671/.

Zhang, Zheng, Pierre Zweigenbaum, and Ruiqing Yin. “Efficient Generation and Processing of Word Co-Occurrence Networks Using corpus2graph.” *ACL Anthology*, (2018). https://www.aclweb.org/anthology/W18-1702/.

Appendix

Figure 1a - Sample interaction scoring response

```
C:\Users\Joshua\Northwestern\Knowledge Engineering\Salit_Assignment_4>python entrypoint.py
INFO:__main__:Hello! How are you? I have mastered elementary school education.
        Please ask me a question.
If you have a question, please ask otherwise respond "no".no
INFO:__main__:Okay let me find a good question for you.
Would you like to score it or just see responses?score
INFO:__main__:That is a good question, the top results are
INFO:__main__:Ok How about question: how do most fish get the oxygen they need to survive
INFO:__main__:Getting facts for term ['fish', 'get', 'oxygen', 'need', 'survive']
INFO:__main__:The explanation ID f59c-9a1b-f57b-d689 was not ranked at all
INFO:__main__:The explanation ID 9526-aecc-22b2-4b6a was ranked at 2
INFO:__main__:The explanation ID 686d-e4be-a123-314f was ranked at 3
INFO:__main__:The explanation ID 571a-0a31-ad7b-e097 was not ranked at all
INFO:__main__:The explanation ID f32e-0424-d7c1-e558 was not ranked at all
INFO:__main__:The explanation ID 65bb-ee0e-f67d-f6d9 was ranked at 6
INFO:__main__:The explanation ID bf2f-e380-be4e-ca06 was not ranked at all
INFO:__main__:0.42857142857142855 is not too bad
```

Figure 1b - Sample interaction with custom question

```
C:\Users\Joshua\Northwestern\Knowledge Engineering\Salit_Assignment_4>python entrypoint.py
INFO:__main__:Hello! How are you? I have mastered elementary school education.
        Please ask me a question.
If you have a question, please ask otherwise respond "no".How much of the earth is made up of the ocean?
INFO:__main__:Getting facts for term ['much', 'earth', 'made', 'ocean']
INFO:__main__:earth is made of rock. an ocean is a kind of body of water. the ground is part of the earth.
earth is a kind of planet. the crust is a layer of the earth. the sun is the  star that is closest to earth.
the mantle is a layer of the earth. the arctic ocean is a kind of ocean. the water cycle occurs on earth.
the surface of the earth is made of 70% ocean
```

Figure 2 - Process flow diagram

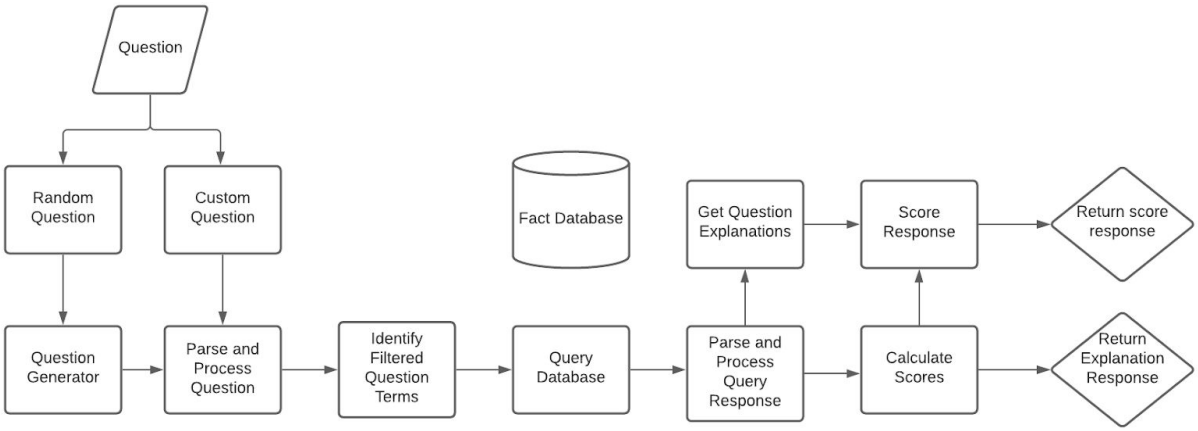

Figure 3 - Neo4j Admin Import Tool Upload data. This should be executed in the local “bin” folder of the Neo4j database created.

```
neo4j-admin import ^
   --nodes ..\import\sentence_node_table.csv ^
   --relationships ..\import\shared_term_relationships.csv
```