CSI 5180 Project Definition (Group 11)

Zhikun Yuen (300323972), Zidu Yin (300309036)

Project Title: Improving a simple knowledge graph question answering (KGQA) model to answer complicated questions on Harry Potter wiki's knowledge graph through ChatGPT

Project Description

1. What will you achieve (learn + produce) and what is your prior knowledge of this task?

   We want to have an understanding of how KGQA works and how is the workflow of a KGQA process. A KGQA model is pre-trained on the Harry Potter wiki's knowledge graph already. However, this model can only answer simple questions (single-hop questions), e.g. In which house is Harry Potter? For some more complicated questions, e.g. Who is the Minister for Magic during Hector Fawley's term? Our goal is to improve the simple model and make it answer some more complicated questions, i.e. we want the model to translate some complicated questions to SPARQL queries instead of translating simple questions only.

   Prior knowledge we need in this task is knowledge graph, SPARQL queries and seq2seq model, e.g. transformer.

2. What algorithms/approaches will you be testing/developing?

   We will use a pre-trained BART model. It is a seq2seq transformer, the encoder can be BERT and the decoder can be a lightweight GPT. The task of the model in our project is to translate text questions into SPARQL. The BART model is already pre-trained on the LC-QuAD 1.0 dataset and then fine-tuned on the Harry Potter wiki's knowledge graph with some very simple questions and SPAQR query pairs only.

   We will use ChatGPT to generate more complicated questions and SPARQL pairs. Our aim is to generate around > 200 question pairs after filtering (some question pairs generated by ChatGPT may be wrong, we need to filter those wrong data). The new data will be used to fine-tune the BART model and let it to translated complicated questions to SPARQL queries.

3. What will be the final deliverable and by whom could it be used, or what would be its contribution to the field?

   Most of the KGQA datasets are based on DBpedia, Wikidata and Freebase. If someone wants to custom a KGQA model on some other data or knowledge graphs, there are no datasets for fine-tuning. The most common way to fine-tune the model for a custom knowledge graph is to generate enough questions and SPARQL queries. Then, use these new data to fine-tune the KGQA model. However, generating corresponding text questions and SPARQL queries by humans is very time-consuming. We want to improve the KGQA model to answer complicated questions on a custom knowledge graph with the data generated by ChatGPT. This can help to reduce the time of annotation. We hope our model can answer some complicated questions on a custom knowledge graph and be general to unseen questions on the training data after fine-tuning. The developer can customize the KGQA system or search engine and can use our approach to fine-tune the seq2seq model on their own KG.

Customizing a KGQA is very difficult because we need to generate new data from the custom knowledge graph. Our contribution is to make the process of fine-tuning KGQA model to be simpler with the help of ChatGPT and shows that the data generated by ChatGPT can be useful as a data augmentation approach to improve the KGQA model.

4. What are the project boundaries that you are setting to be able to achieve your project within 25 hours * number of people in the group? In other words, what is included/excluded to make your achievement realistic.
   We will create a dataset with around 200-250 pairs of text questions and SPARQL queries. Instead of using the full graph (the large graph can’t be inputted to ChatGPT), we will sample several small subgraphs with DFS or BFS and use them to ask ChatGPT to generate training pairs. Then, we will use the data to fine-tune the pre-trained BART model. This is all that we plan to do in our plan.

   What we will not do is we will not focus on the modelling part. Modelling is not our consideration in this project. Also, we will only generate a small new dataset. To let BART translate simple text questions to SPARQL queries on Harry Potter’s wiki KG, the BART model is only fine-tuned on very few simple questions. To make the data generation easier, we will also use a small dataset even though our BART model needs to answer some complicated questions. These can help us save time and finish the project within time.

Description and justification of the software platform / programming involved / dataset involved

Programming language:

- Python

Knowledge graph data:

- Harry Potter Wiki: a wiki page to contain all the information of the story of Harry Potter. The knowledge graph is built on top of this wiki page with 118,543 triples.

  The following are some example triplets from the Harry Potter’s knowledge graph:
  hp:Gryffindor rdf:type hp:House_ .

hp:Gryffindor hp:name hp:Gryffindor .
hp:Gryffindor hp:founder hp:Godric_gryffindor .
hp:Gryffindor hp:colors hp:Scarlet_and_gold .
hp:Gryffindor hp:animal hp:Lion .
hp:Gryffindor hp:element hp:Fire .
hp:Gryffindor hp:traits hp:Courage .

Query for the knowledge graph:

- SPARQL query: the standard query language to query results from the knowledge graph in RDF format. RDF is a graph format for our knowledge graph.

Packages/tool:

- Haystack: an end-to-end framework that enables us to build powerful and production-ready pipelines for different search use cases. KGQA is the very first version of this package. They provided the Harry Potter wiki's knowledge graph and the pre-trained BART model to answer very simple questions only.
- ChatGPT: A large pre-trained language model (LLM) for question answering
- HuggingFace: a platform to provide package to download and fine-tune the LLM, e.g. BART

Activity Table

Activity	Why	Time planned	Deliverables
Read article	Gather knowledge about Knowledge Graph Question Answering	6h	Fundamental understanding of the model and summary of important ideas used for project
Understand the package	Have an idea of what package can be used	5h	Know when and how to use the package in an appropriate way
Learn the SPARQL	It is an important language to understand the relationships in the graph	6h	Understand the result in terms of SPARQL
Write code to generate dataset and filter the improper data	The dataset would be large enough for model to train and we have to delete meaningless data	10h	Refined dataset for model
Write code to fine-tune the seq2seq model	Required to training models using HuggingFace and PyTorch	8h	Training function and dataset and dataloader class based on the HuggingFace and PyTorch
Training the model	Develop an improved model	10h	A KGQA model to answer more complicated questions

Evaluation the model	Evaluate the performance of our model on complicated questions and its generalization	5h	Show our approach works and can improve the model to answer complicated model with a simple data augmentation/generation approach