Named Entity Recognition and and Relationship Extraction using a Knowledge Graph

Raktim Dey (MDS202132)

Project Report

September 5, 2024

Acknowledgements

I would like to express my sincere gratitude to Dr. Arun Ayyar, my project supervisor, for his valuable guidance, advice and support throughout this project. His expertise and encouragement have been instrumental in the successful completion of this work.

I would also like to thank Professors Sourish Das and Priyavrat Deshpande for their assistance with various aspects of the project.

Furthermore, I would like to express my appreciation to my friends and family for their unwavering support and encouragement throughout this project.

Thank you all for your valuable support and encouragement.

Abstract

Named Entity Recognition (NER) and Relationship Extraction (RE) are important tasks in natural language processing (NLP) that involve identifying entities and the relationships between them in unstructured text data. In this project, we propose a methodology for NER and RE using a knowledge graph to represent the extracted entities and their relationships. We leverage the power of state-of-the-art NLP techniques, such as pre-trained language models, to identify entities and relationships in the input text. We then map these entities and relationships to nodes and edges in a knowledge graph, which captures the semantic relationships between the entities. Finally, we use graph algorithms to extract additional insights from the knowledge graph, such as identifying key entities and relationships, clustering entities based on their relationships, and predicting missing relationships.

We evaluate our methodology on a publicly available dataset and demonstrate its effectiveness in accurately identifying entities and relationships, as well as in generating meaningful insights from the resulting knowledge graph. Our approach has potential applications in various domains, such as information retrieval, question answering, and recommendation systems. Overall, our project contributes to the growing field of NLP and knowledge representation, and provides a practical framework for NER and RE using a knowledge graph.

Introduction

The SEC filing is a financial statement or other formal document submitted to the U.S. Securities and Exchange Commission (SEC). Public companies, certain insiders, and broker-dealers are required to make regular SEC filings. The most commonly filed SEC forms are the 10-K and the 10-Q. These forms are composed of four main sections: The business section, the F-pages, the Risk Factors, and the MDA.
Documents like these contain a lot of information that cannot be directly fed into a machine. We need to make our machine understand the data beforehand. For this purpose, we need sentence segmentation, NER and various other NLP tasks to perform on the data. On that note, we can use the idea of Knowledge graphs to extract company names, its products, competitors, locations and link them using relationship extraction. Transformer models like

BERT, Zshot and the usual NLP models, namely spaCy and NLTK can be used to extract information and build the knowledge graph.

The llama index is a database of all the filings made to the SEC since 1994. It contains information about the filing date, company name, and other relevant data. The SEC Downloader API provides access to the SEC's database of filings, allowing users to download the full text of the reports in various formats, including PDF and HTML.

To begin, one can use the llama index to identify the specific filing(s) they are interested in, such as the Form 10-K of a particular company. Once identified, the user can then use the SEC Downloader API to retrieve the filing and store it in a local database or file system.

Once the filings have been downloaded, they can be processed to extract useful information. One approach is to use natural language processing (NLP) techniques to build a question answering model that can answer questions about the company's financial performance, business operations, and other relevant topics.

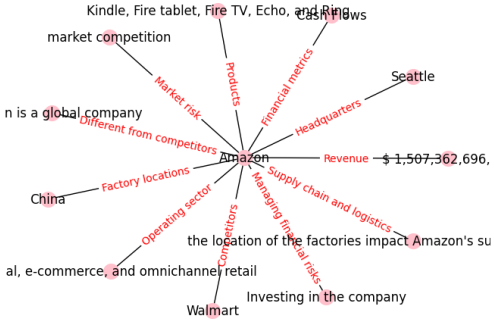
Methodology

After installing all the modules necessary to generate the knowledge graphs, namely SEC-Downloader, langchain, pyvis,trasnformers, etc. , we use the Downloader function of the SEC-Downloader module to download the SEC-10K filings of the company whose graph we want to build. But this data is available in a .txt format. So, after several steps of pre-processing which includes scraping the text, restoring the non-alphanumeric parts of the text, we extract the necessary 10K data that we need. Note that there are sections from 1 through 15 in the 10K filing, each of which contains unique financial information about the company.

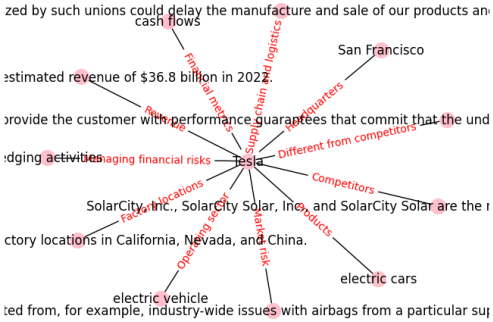
Now we define our Language model class for question answering, where we use google/flan-t5-large model for text-2-text generation.
After defining our queries and labelling them with the section number of the SEC-10K data where we know the answer would be found, we use the llama index api to generate our answers from the text. Then we use the Networkx module to generate our knowledge graphs.

Results

Knowledge graphs generated using the aforementioned techniques for some popular companies can be visualised below:

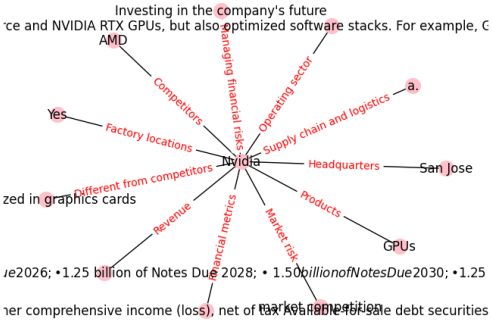


(a) Amazon



(b) Tesla

(c) Microsoft



(d) NVIDIA

Conclusions

The use of the SEC-Downloader API and the LangChain Transformer model can greatly enhance the process of generating knowledge graphs and answering complex questions related to financial data. The SEC-Downloader API provides a reliable source of financial data, which can be used to create a knowledge graph that represents the relationships between different entities in the financial world. On the other hand, the LangChain Transformer model can effectively process natural language queries and provide accurate answers by analyzing the information in the knowledge graph.

By combining these two technologies, we can create a powerful system that can help investors, analysts, and other financial professionals to quickly access relevant information and make informed decisions. For instance, the system can be used to analyze financial reports, track market trends, identify potential risks, and evaluate investment opportunities. Moreover, the system can be customized to suit specific requirements, such as filtering data by industry, region, or other criteria.

In summary, the use of the SEC-Downloader API and the LangChain Transformer model can greatly improve the efficiency and accuracy of financial analysis and decision-making. As these technologies continue to evolve, we can expect to see more sophisticated systems that can handle even more complex queries and provide more insightful answers.