Article semantic visualization

1. Introduction

The objective of this project is to visualize the **semantic relationships** and logical structures across multiple articles. By connecting the central points or keywords of these articles using nodes and links, we aim to create a cohesive and interactive representation of the content. This visualization serves several practical **purposes**:

1. **Operational Insights**: Clients will be able to easily comprehend the current operational data of their company through a comprehensive visual chart, providing clear and immediate insights.
2. **Data Selection and Modeling**: The data team can utilize the data modules to select different models across various domains, as well as adjust the parameters to fine-tune these models for better outcomes.
3. **Training Data Refinement**: By extracting key content and modules, we can efficiently distill and refine training datasets, optimizing the data for further analysis and model training.

This project not only enhances data comprehension but also supports strategic decision-making processes through intuitive visualization and data-driven insights.

1. Method
   1. **TF-IDF**
      1. Algo:

A graph of a line

Description automatically generated

This diagram represents the probability of a term appearing within a document, which can also be interpreted as the relative frequency of the term within that document.

The X-axis represents the term frequency (TF), calculate how many times the words show up.

The Y-axis represents TF-IDF, how many time the term appears in other documents. Inverse (IDF) stands for if the term appears less time in other documents, then the character will be given a higher score, as it might not be some common words, such as ‘a, the’ etc.

The Z-axis represents dispersion (p), where a lower value of p indicates lower dispersion

In the diagram, the top right corner is indicative of terms with a high frequency compared to average words, suggesting these might be related to general topics. The top left corner represents terms with low frequency compared to average words, which might be significant or specialized terms related to the major subject. The lower left corner reflects terms that have low frequency and appear an average number of times across the documents, likely indicating less important or context-specific terms.

A graph showing a line of dots

Description automatically generated

This diagram represents the probability of a term appearing within a document, which can also be interpreted as the relative frequency of the term within that document.

The X-axis represents the term frequency (TF), calculate how many times the words show up.

The Y-axis represents TF-IDF, how many time the term appears in other documents.

* + 1. Reference:

Code:

<https://github.com/roverbird/corpus_utils?tab=readme-ov-file>

Paper:

<http://siba-ese.unisalento.it/index.php/ejasa/article/view/12119>

* + 1. Evaluation:

This method proves ineffective for several reasons. Firstly, it fails to capture the connections between paragraphs across different articles. Additionally, the visualization does not provide meaningful insights, as the extracted terms are predominantly prepositions, pronouns, and other non-informative words. Therefore, it does not effectively convey the semantic relationships or key concepts within the articles.

* 1. **Embedding**
     1. **Word to vector embedding** (muti dimension to 2D or 3D dimension)

A diagram of a computer program

Description automatically generated

Data Processing and structure:

1. Divide text into chapters/paragraph
2. data scraping and cleansing
   1. remove number, html format
   2. tokenize
   3. POS tagging (Noun. + Noun. combo)
   4. Remove duplicate combo
3. Extract Embedding
4. Visualization

Result:

* + 1. Nvidia Financial Report

A graph of a graph with many different colored numbers

Description automatically generated with medium confidence

Explanation: We can see the keyword of each paragraph: expense, yearly revenue, income earnings, earnings share, stock compensation.

The x,y,z axis don’t have specific meaning. The algorithm applied PCA techniques to make the 50 dimensions vector to a 3D interface.

1. Novel: pride and prejudice

A screen shot of a graph

Description automatically generated

Same method and visualize the first chapter of novel. We cannot find the exact meaning on what is talking about in this chapter. We cannot understand the story based on the words like feelings, fortune, wife, views, etc.

Failing reason: The words in the novel are high similarity for embedding vectors, for instance in the same literature categories. So, currently is difficult to separate them.

* 1. **Language Model Finetuned for entity and relation extraction** 
     1. Gpt-2 finding (entity-entity-relation)

We started with using gpt-2 model to fine tune the input with minizine the loss function. Entity relation extraction training:

A screenshot of a computer

Description automatically generated

Finding: The outputs (uber, drop, drops), (adobe, vector space) are not in the entities-relations format, which means the model could not learn how to extract keywords.

* + 1. Llama3 finetuning (entity – entity- relation)

Entity relation extraction training

A screen shot of a computer

Description automatically generated

Finding: The outputs (I, proposed endeavor, positioned) are accurate, and is meaningful for visualization

* 1. **Llama3-unsloth**

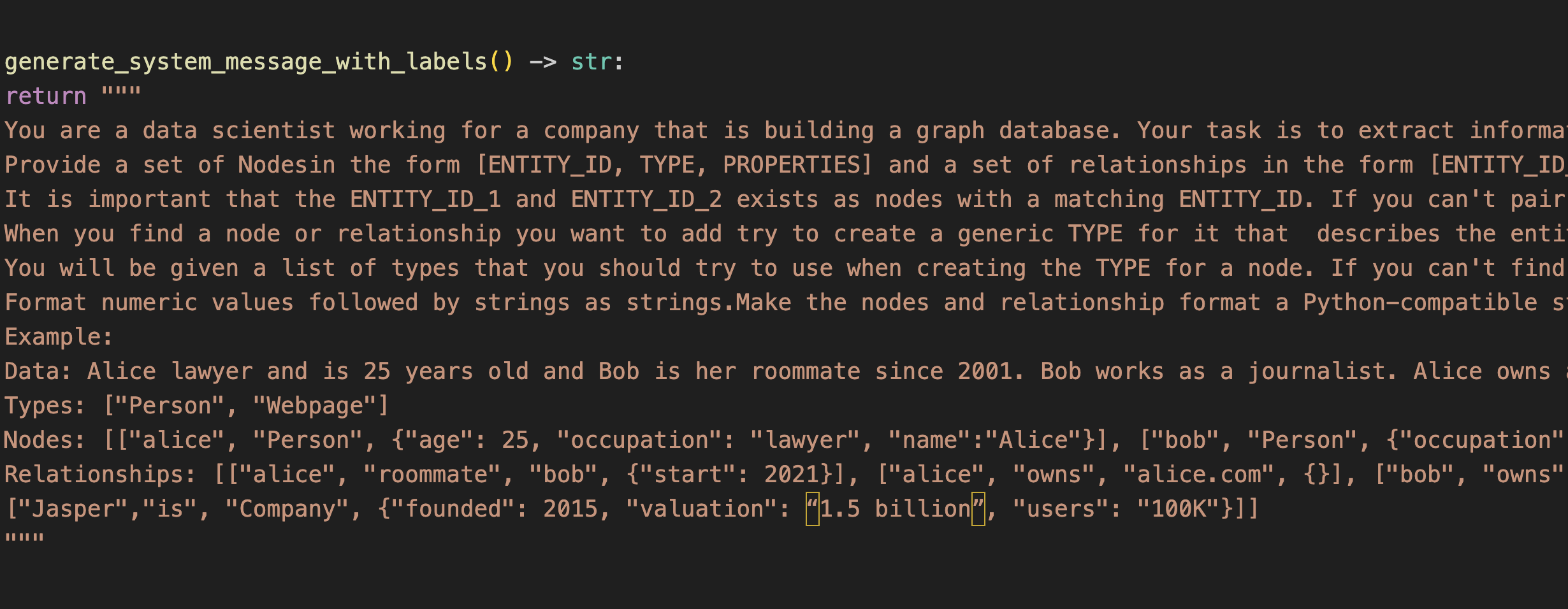
Workflow:

A diagram of a workflow

Description automatically generated

* 1. The workflow started with chunk the documents into 6000 max tokens and 1000 max new tokens.
  2. Then the workflow went into unstructured data extractor to extract entities and relations.
  3. Entities, Relations data cleaning
  4. Graph

Prompts for node extraction:



Results:

A screen shot of a computer screen

Description automatically generated

A diagram of different languages

Description automatically generated

Analysis: From this graph, we can understand, the article’s main topic is about company SHEIN, the extracted entities include markets：Canada，Mexico，U.S., Poland; partners: Amazon, Zara, Temu; company backgrounds: founders and cofounders.

The advantages are we can tell the three topics from the graph, and the articles might include data from those entities where stand for a whole paragraph.

Reference：

<https://medium.com/neo4j/harnessing-large-language-models-with-neo4j-306ccbdd2867>

<https://medium.com/neo4j/knowledge-graphs-llms-fine-tuning-vs-retrieval-augmented-generation-30e875d63a35>

<https://medium.com/neo4j/knowledge-graphs-llms-multi-hop-question-answering-322113f53f51>

<https://medium.com/neo4j/knowledge-graphs-llms-real-time-graph-analytics-89b392eaaa95>

<https://medium.com/neo4j/construct-knowledge-graphs-from-unstructured-text-877be33300a2>

* 1. GraphRag
     1. Architecture:

A screenshot of a graph extraction

Description automatically generated

* + 1. Reference(paper): <https://arxiv.org/pdf/2404.16130>
    2. Results and Analysis

The demo datasets include two LLM papers, one finance article, and two novel pdfs.

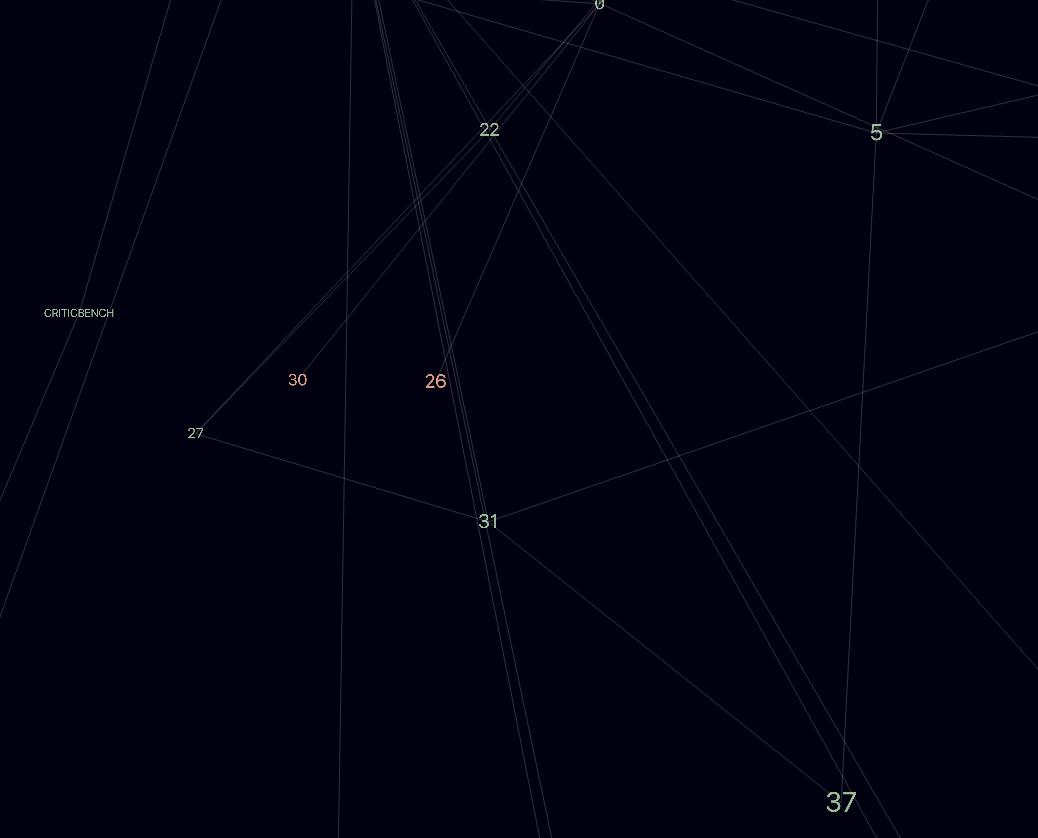
A close up of text

Description automatically generated

A black background with many lines and text

Description automatically generated with medium confidence

ARXIV is the brightest point, we can tell most articles are published on arxiv, and the categories are research papers.



This area is clustered for numbers, but it’s hard to analyze what are those numbers stand for.

* 1. **Kv Cache**

Result:

The purpose for adding kv cache score is that we hope to set node direction and distance in between other nodes. We took the inverse of score as node distance so that stronger connection (higher score) have shorter link, representing closer relations.

Result:

A network of white text

Description automatically generated with medium confidence

In this way, the finance data are separated from the Nvidia background data, which shows larger distance.

1. Future direction

For now, we need to combine the kv score and entities we extracted from language model. The current issue is how to extract the correct kv cache as there are many similar tokens appear in the same file, and they are chunked differently.