Project 2: Information Extraction using SpanBERT and GPT-3

Group Members:
Suchit Sahoo(ss6630)
Aarushi Jain(aj3087)

List of Files:
Main.py
googleSearchModule.py
GPT3SetExpansion.py
extractText.py
extractTupleSetExpansion.py
spacy_help_functions.py
spanBertSetExpansion.py
requirements.txt
Config.json

Problem Statement:
Given an input relation - "bill gates: microsoft", generate more relations of the type - "person: employee of" from the extracted webpages using SpanBERT and GPT-3

Setting up the project:

- Extract the proj2.tar.gz file
- Navigate to the proj2 directory
- Run the following command
    - pip3 install -r requirements.txt
- To start using the program run the following command
    - python3 main.py [-spanbert|-gpt3] <google api key> <google engine id> <openai secret key> <r> <t> <q> <k>

Google API Key: AIzaSyAMw1otSb1Tuh6Qe81Eo799bNSO55cqtTI
Google Engine Id: 59dddcc6d7ae4afdd

Design Components:

Main.py File: The entry point to our application is the main method defined in the main.py file. It takes 8 arguments-Google API key, Google Engine Id,ApenAI secret key,relation type, threshold, query and number of tuples requested. It returns the top k results to the user. If the k tuples are not extracted, it modifies the query using extracted tuples . This process is repeated till k tuples are extracted.

googleSearchModule.py File: This module filters out the non-html files and returns the top 10 google search results. It takes 2 arguments-Google API key and Google Engine Id.

extractText.py File: It extracts the text from the web url and trims the web content to 100000 characters. It processes the data to remove non ascii characters.

spacy_help_functions.py: It has helper functions like create_entity pairs which takes a spaCy Sentence object and a list of entities of interest as input and returns list of extracted entity pairs

spanBertSetExpansion.py: It loads the SpanBERT model and predicts the relation type and confidence score based on the given input.

extractTupleSetExpansion.py: It processes the web content and extracts the tuples from it using spanBert/GPT-3 model.

Algorithm Design:

1. **Pre-Processing of the Google Search Results:** For the relevant documents obtained from google search results we extract the contents from the URL. Text is truncated to 10,000 characters. Non ASCII characters,redundant newlines and some whitespace characters are removed from the text .

2. **Creating Entity Pairs:** Spacy model is applied to processed text to split it into sentences. Entity pairs are extracted from these sentences using the create_entity_pair function defined in spacy_help_functions.py. Check if the extracted entity pairs have the desired subject and the object. Process the entity pairs which have the desired subject and object present.

3. **SpanBert Model for Prediction:**The entity pairs are converted into SpanBERT input form. Pre-trained SpanBERT model is loaded using the spaCy library. If the predicted is same as the desired relation, tuple is added to the extracted tuples list along with the confidence score

4. **GPT-3 Model for Prediction:**The entity pairs are converted into GPT-3 input form.Input is fed to the GPT-3 model by making a call to openAI API. If the predicted is the same as the desired relation, tuple is added to the extracted tuples list.

5. **Processing the Results:**
    a. **Top-k tuples are extracted:** Sort the tuples according to confidence score and return the results
    b. **k tuples are not extracted:** From the extracted tuples set,
        i. spanbert: select a tuple with the highest confidence score
        ii. GPT-3: select an arbitrary tuple

Create a query q from tuple y by just concatenating the attribute values together. Fetch the google results for this query and repeat the process till k tuples are extracted. If no such tuple exists, then stop.

Flow Diagram: