Describing the submitted Code for the first Data Challenge of HERMES: “Challenge the Newspaper Data!”

Golnaz Sarkar Farshi

1. The cell **"Convert the Pickle to JSON"** is redundant because the code in the following cell (which cleans the files) also converts pickled pandas dataframes to JSON. Additionally, this code only extracts newspaper articles from the *Kölnische Zeitung*, whereas the code in the next cell considers all newspapers.

2. In the second and third cells, **"Cleaning the Files"** and **"Starting the Cleaning”,** the code first converts the pickled dataframes provided by HERMES into JSON files. Then, it calls the OpenAI API and uses ChatGPT for text cleaning, addressing issues caused by poor scan quality and OCR errors. Here is the prompt that was used to perform this task:

"You are presented with some text in German from in between 1916 and 1945. It is written in the old German of that time. Please go through the text and adapt it to the modern german Language, removing spelling mistakes and grammar mistakes where appropriate. Try to recover the text as faithfully as possible, while minimizing the number of changes. Replace any character from old German language with their equivalents of today, e.g. "ſ" -> "s".Only return the corrected text, do not add any prose or explanations on what you did."

After this cleaning step, the cleaned text, along with its embedding (calculated by the "nomic-ai/nomic-embed-text-v1" model), is saved as additional key-value pairs in the JSON file. This is while the code later uses a different model, namely Google Vertex AI, for embedding. What does this embedding step mean, if later a different embedding model is going to be used?!

3. In the next step, **“Summarize the newspaper articles”**, the code uses Google's Gemini to generate dense abstracts for each newspaper article, although these abstracts are not stored anywhere and not used later. Gemini is employed to generate summaries and topic-related metadata (such as entities and sentiment) for the articles. Furthermore, under the heading **"Summarize the newspaper articles",** the team mentions that they are focusing on three newspapers for cost and time efficiency. However, based on the code, it seems they are only examining *Süddeutsche Zeitung*. It also appears that the code in this section does not produce any results that would be used by other cells, so the entire cell could be omitted.

4. Under the headings **"Embedd and classify the sentiment"** and **"Clustering"**, the code clusters the texts based on their content, which is in turn represented by the semantic embedding of the full texts. None of the code in the previous cell is used here, except the one that does text cleaning. Here, the Google Vertex AI embedding model text-embedding-004 has been used for embedding.

The code under the heading **"Clustering"** filters the dataset to include only articles from the years 1914 to 1919 and retains only those with a neutral sentiment. It then assigns the filtered texts to 10 different clusters. The output of this cell is a JSON file containing the fields "title”, "content”, "embedding”, and "sentiment." The clustering is performed using the KMeans method. It is unclear to me why the summaries were generated in the previous step if they were not going to be incorporated into the embeddings. The purpose of the clustering is also not clear. Furthermore, newspaper articles with a neutral sentiment, which should be the output of this cell, are not used in the following steps at all. So, this cell has no function for the further analysis either.

5. Under **"Summarization of the Reflection with Google Gemini”,** the previous steps are repeated, but this time the code filters out newspaper articles from 1914 to 1919 with a positive sentiment, rather than neutral. The Gemini API is then called again with the prompt: "How did German Newspapers reflect positive sentiments about technological advancements and what factors influenced these perceptions? Newspaper articles:". The script loops over each of the 10 clusters from KMeans, gathers the texts of the articles in the cluster, and creates a combined text that includes all the titles and content of the articles, along with a statement about their positive reflection on technology within that cluster.

I don't understand the purpose of the clustering, as all the clusters contain newspaper articles with a positive sentiment. Furthermore, the clustering does not seem to play a role in the subsequent steps.

6. Under **"Now letting the llm reflect how it changed from 1914 to 1944"**, the code prompts Gemini again with the following: "Du bekommst zwei text inputs der erste ist ovn 1914-1919 und der zweite von 1938-1944. How did German Newspapers reflect postive sentiments about technological advancements and what factors influenced these perceptions? Newspaper articles:". The team has not provided the code that summarizes the sentiments extracted from the newspapers between 1938 and 1944, but it is likely that they used the code from the previous cell to do so. Gemini receives two files here: one containing the summary of the content of newspaper articles with positive sentiments from 1914-1919 and the other from 1938-1945. These files contain plain text, summarizing the overall positive sentiments from the newspapers in each period. The summaries have been generated by Gemini using the code from the previous cell.

The final text result in the Jupyter notebook is again the output generated by Gemini. The first part discusses whether and how newspapers from each era reported on technological advancements with a positive sentiment. There is also a section beginning with "Negative Reflection”, but the team has not provided any code that filters out texts with negative sentiments. The results of these generated texts—regarding both negative and positive sentiments about technological advancement—are controversial. It is unclear whether the sentiments in the newspaper articles regarding technological advancement are ultimately negative or positive. It is also unclear whether the negativity of newspaper reports on technological advancements is really reflected in the newspaper articles themselves or just a moral message generated by Gemini.

**Reflections:**

Would Gemini generate the same results if not provided with the newspaper data? How can we know that these results are based on the newspaper data we provided and not on Gemini's training data?

In summary, the code contains multiple cells that serve no function. It heavily relies on paid LLMs such as the Google Vertex AI embedding model, Google's Gemini model, and OpenAI's GPT models. It is not possible to go through the code and determine whether the submitted results are based on the provided dataset or simply generated by Gemini, because we do not have access to the paid models and APIs that the team has been using.