Sponsor Motion Summary Status Report

Scalable and Cost-Effective Deduplication: Leveraging Algorithms and LLMs

Group Members: Rohan Chaudhary, Sarmad Kahut, Valentina Torres

Business problem:
SponsorMotion is a consulting and data company with a specialized AI-powered database of US events. Founded by experienced professionals in the industry, they leverage artificial intelligence to create a comprehensive database of healthcare related events in the United States. Their mission is to make all events easily discoverable for both sponsors and attendees, streamlining the process of connecting the right sponsors with the right events.

The goal is to optimize the data ingestion process of SponsorMotion. Although the current pipeline is functional, it needs improvement to enhance the company's scalability and achieve the goal of making all US events searchable. The specific challenges to address are optimizing the costs of post-scrape filtering and processing of event records, as well as establishing a fully automated data quality control process to identify problematic or duplicate records.

Scope:
The scope of the project involves large language models. Our objectives are to reduce costs measured as $/1,000 captured events, and decrease the number of duplicate records. Additionally, we will explore the possibility of smarter characterization of events and the possibility of giving event recommendations to sponsors based on their interests. Moreover, we will investigate alternative data sources to accomplish similar goals.

Expected approach:
To address the challenges and achieve the project objectives, the expected approach involves leveraging large language models (LLMs) and employing various text similarity detection techniques. These can aid in optimizing the data ingestion process by improving the post-scrape filtering and merging of duplicate records.

Initial analysis:
During the initial data analysis, our team received a dataset of US healthcare events with 21 columns and about 48k rows. However, the focus was primarily on the "start date of the event," "name," "state," and "summary" columns, as they were deemed most helpful for the analysis. The remaining columns were not considered, as the main problem to tackle was duplication, which could be addressed using the summary and name columns. To begin with, we eliminated 2767 rows that did not have a summary and start date. The start date column required standardization to ensure consistency across all rows. Because of this, we decided to eliminate the timestamp it had, and we added an extra column where we change its format to DD/MM/YYYY. The summary column underwent initial cleanup, involving the removal of stop words, converting all text to lowercase, and eliminating the most frequent words in the descriptions which included conference, health, annual, and medical. This step aimed to decrease the likelihood of false duplicates due to commonly used words in the healthcare industry.

Research:
To have a deeper insight into the business process, we started by understanding the workflow when a user searches a specific division related to a healthcare event. Events can be classified as medical conferences, seminars, trade shows and exhibitions, webinars and online events, marathons or a walk

related to a medical cause, etc. This process involves scraping URLs using LLMs (Language Model Models) and creating a database. The steps can be explained as follows:

1. User performs a specific division search which is sent to Google to obtain URLs.
2. The URLs of these search results are passed to a system called Bubble, which has a database to store them for further processing.
3. The text resulting from the URL scraping is sent to OpenAI for analysis using LLMs. If the URL corresponds to an event, OpenAI extracts various event characteristics such as location, dates, summary, etc.
4. If the analyzed URL is identified as an event, the information is stored in a database, utilizing a service like Pinecone for efficient storage and retrieval.
5. If the analyzed URL does not correspond to an event but is instead a list or other content, it is sent back to the Node for further scraping.

The process continues until all URLs have been processed or until there are no more duplicate events identified.

The process currently uses GPT-3.5 to scrape URLs, as this LLM model gives efficient output by incurring affordable cost, though the company is trying to optimize the cost further as the size of the data increases and with further developments in OpenAI models. This served as the initial point of research as we started by understanding the functioning of LLMs (Language Model Models) by focusing more on the analytical capabilities, prompt evaluation, and data generation part but also keeping in mind the technical engineering aspect of LLM development. Various online articles and websites helped us in getting acquainted with the architecture of LLMs and utilizing LLM models for various applications in Natural Language Processing. Specific research papers and video tutorials helped us to dive further in evaluating the generative capabilities and efficiencies of LLM models and how tools like LangChain can be used to fine-tune and customize these models for user specific purposes.

Since the scope of our project involves the cost structure, we reviewed the pricing information provided by OpenAI. The problem of cost optimization led us to an informative research paper about FrugalGPT. The paper explores and evaluates three distinct strategies to lower the computational cost associated with utilizing LLMs. These strategies include prompt adaptation, LLM approximation, and LLM cascade. To illustrate the LLM cascade, FrugalGPT effectively determines the appropriate combinations of LLMs to utilize for different queries, aiming to enhance accuracy while minimizing expenses. For the deduplication problem, we explored similarity detection algorithms and models such as cosine similarity with TF-IDF vectorization, word2vec, fuzzy matching, and BERT along with calculating cosine distance.

Solution tried:

One of the significant challenges of the deduplication problem is finding a right balance between the false positives and false negatives such that the algorithm recommended can be implemented on a larger scale.

With the objective of measuring the closeness in meaning between summaries and identifying duplicates, we explored various models and techniques to achieve this goal. We initially focused on semantic text similarity, which measures the closeness of summaries. To assess this, we employed the cosine similarity metric with vector representations of the text. We utilized TF-IDF (Term Frequency-Inverse Document Frequency) vectorization to represent each summary as a numerical vector. The cosine similarity between these vectors was then computed and compared to measure the similarity between two summaries. We also explored the use of Word2Vec to capture semantic meaning and context. By averaging or summing the word embeddings of the words in a summary, we represented

the summary as a numerical vector. Cosine similarity was once again used to measure similarity. Finally, we tested BERT, a transformer-based language model to create a matrix based on contextual tokenization of corpus present in each summary.

Initial Results:
After experimenting with these models, we determined that cosine similarity with the TF-IDF vectorizer was flagging more accurate results since other models were either flagging a large number of duplicates or none at all, i.e.. large false positives and false negatives. Therefore, we proceeded with further analysis of the output by changing the threshold for similarity calculation to a range of 0.6-0.9 and comparing results by exporting the output.

Conclusion and further refinement for Precision:
Since the output was still flagging a large number of false duplicates and the computation time was large, we decided to sort the data based on predefined criteria, such as start date and state, to improve organization by ordering the events and facilitating accurate comparisons. This will be used as a bucketing technique to group events based on common attributes, allowing similarity analysis within specific subsets.

Expected Findings:
The most crucial output from this project will be resolving the deduplication problem and making sure that such events are merged into a single record. In addition, we intend to develop a language model trained on the dataset. This model will optimize the cost of using several GPT models by utilizing the least costly models first and then moving on to the more powerful and costly ones as needed. Once these primary business problems are solved, we could potentially explore the option of scaling the model to industries other than healthcare. This would not only enhance revenue opportunities for SponsorMotion, but it would also help to fine-tune future models with the data. Finally, we can also look at streamlining the automation of the taxonomy of events pertaining to a specific specialty. This would make it much easier for clients to use the platform to find events that fit their specific needs.

Challenges:
When working on the deduplication problem, one of the challenges is to get an algorithm that does not correctly interpret the data, which could result in the loss of legitimate records during deduplication. It is essential to have sufficient computational resources to work with models like GPT and automate the entire process. However, as we scale the project, we may run out of computational resources. One of the foreseeable challenges associated with the use of LLMs as an important aspect of the business is that these models are intrinsically unstable and are constantly evolving in terms of their efficiency and costs which makes them not fully reliable to use as the company aims to expand their dataset to include all US events.

Next steps:
We will explore the use of a lexical similarity approach through fuzzy matching using event names to account for potential variations or typographical errors, enhancing the detection of similar summaries despite minor discrepancies. We will explore further optimization and fine-tuning to continually improve the accuracy and efficiency of the text similarity analysis for duplicate event detection. Finally, after having solved the deduplication problem, we will be working on solving the cost optimization challenge.

REFERENCES:

- Fuzzy Matching:
  https://www.youtube.com/watch?v=1jNNde4k9Ng
  https://www.youtube.com/watch?v=y-EjAuWdZdI&t=729s
- Word Embedding:
  https://towardsdatascience.com/text-classification-with-nlp-tf-idf-vs-word2vec-vs-bert-41ff868d1794
  https://www.geeksforgeeks.org/python-word-embedding-using-word2vec/
- Web Scraping:
  https://blog.apify.com/chatgpt-web-scraping/
- Language Models:
  https://towardsdatascience.com/the-easiest-way-to-interact-with-language-models-4da158cfb5c5
- Generic LLM intro: https://en.wikipedia.org/wiki/Large_language_model
- OpenAI
  - Cost structure: https://openai.com/pricing
  - GPT-4 announcement: https://arxiv.org/abs/2303.08774
  - Papers on GPT 3.5 vs. 4: https://arxiv.org/abs/2304.13714 (look beyond the "clinical" lens, at what the models did)
  - API: https://platform.openai.com/docs/introduction (note difference between GPT-3.5 "chatty" format and GPT-4)
- Cost Optimization approach: https://arxiv.org/abs/2305.05176