Reyta Akdeniz

Advanced Data Analysis

Final Project Report

FINAL PROJECT:

CLASSIFICATION OF TWEET DATA BASED ON STANCE USING GPT API CALLS

The main objective of this project is to develop a classification model for text-based data, using GPT via API calls. The project utilizes the twitter data that is scraped, by the Politus, based on the condition of including any of a series of pre-selected brand names in the tweet text. Although the specific goal within the case of this project is to classify tweets according to their stance on boycott, the logic and the obtained strategies can be applied to other cases of classification problems. At the end of the project, a logistic regression analysis also going to be conducted to utilize the classified data in an inferential statistical model.

Before the start of classification stage, the data required some preprocess of cleaning. It firstly cleaned from the duplicates, as the same tweets found to be exist in the data more than once. Also, the tweets before the date of October 7th also been deleted from the data since the topic has been a widely discussed issue after this date of recurring of on-going Palestine-Israel war and the reactions towards Israeli occupation has risen massively in Turkey, along with the whole world. The fact, however, that makes the question of this project not the participation to boycott but a stance towards it is that an equal, if not more, amount of reaction towards the boycotters and protestor has been observed throughout the last year.

OpenAI company provides a range of different ways and models of operationalizing the GPT within a code environment for especially larger tasks that requires making, large number of repetitive API calls. Each of these approaches varies in not only their quality performance and applicability to different cases of tasks, but also in their computational, time and financial cost. Thus, the finding the best approach with the most efficient result has been a long task of exploration itself.

The first step and challenge of interacting with GPT by sending prompts via API calls is what is called prompt engineering. It has a certain way of processing and understanding given instructions. For one of the exploration at this stage of the project was that the model does not perform well with tasks that logically involves two different subtasks within itself. At the beginning, the prompt was designed to ask the model classify tweet in one of the four categories; pro-boycott, anti-boycott, neuter, and irrelevant. However, the relevance of the tweet to the issue of boycotting, inherently another matter of question than the stance held, which caused to low quality of performance. The tasks were separated in two different phases, while also another phase of assessing the account type by which the tweet is written added to the process. Thus, the classification, after clarification of the need for these extra steps, consisted of three stages, which lead to creation of the following pipeline repeated for three times; labeling by hand, labeling with gpt, comparing results for evaluation of the accuracy of predictions, filtering unwanted class of tweets out based on labels. The filtering step, of course, for the first two phases and it is not applied at the last phase.

A 200 tweets long data extracted for the testing phase and labeled manually, firstly in terms of the account type, to eliminate whether the tweet belongs to a personal account, or account of an organization, news agency, etc. Then the same data has been labeled by GPT and the responses are recorded another column in the same data. Then, the approaching the case with a regular AI classification model and considering the GPT labels as predictions and the by_hand labels as the actual data, the two columns compared and the performance of the model evaluated based on the confusion matrix and classification report. The specific hypermetric focused on for this phase of classification is decided as precision since the main goal is to filter the non-personal accounts out, and it in fact prioritizes the exclusion of false positives and can afford having some false negatives. Then the data filtered from the non-personal accounts. In the second phase the same process applied for determining the relevance of the tweet to the issue of boycott. The evaluation metric for relevance classification is also precision since the point is again to clean the irrelevant tweets from the data. At the third phase, where the actual stance of the tweet has been assessed, there were three categories, differently from the previous two, namely the anti_boycott, pro_boycott and neuter stances. Since it is not filtering case anymore and all categories are equally significant at this point, focused hypermetric for the stance classification is considered to be f1 score.

At each phase, the results has been observed to understand at what situations the model fails to assess the tweet correctly and to finetune the prompt accordingly. Although this practice contributed to the results, where the precision scores for the first two phases were 0.99 and 0.97 and the f1 macro score for the stance predictions were found to be 0.75, a concern of organically overfitting the prompt to the testing data has been occurred. For validity and providing a kind of a crosscheck, testing stage decided to be repeated with an extra of 200 tweets added to the testing data. Also, learning from the first try of the testing practice that the relevance classification causes a larger portion of data to be filtered out, since the number of irrelevant tweets are comparatively much more than the non-personal account tweets in the data, order of the first two phases is switched by the second try of testing, which created a considerable efficiency in terms of both time and cost.

The results with the same prompts prepared in the first testing period found to be performing equally well in the second try for especially the first two phases, with the precision scores of 0.98 for relevance and 0.99 for account type. For stance however, f1 macro score above the level of 0.63 cannot be obtained despite some additional efforts of refining the prompt. The precision scores for the pro and anti stances are 0.72 and 0.88 at this classification, while the neuter class, with 0.41 score, causing the problem to a great extent.

Although, the testing phase is terminated at this point with the above stated scores for this project, some further ideas of developing and increasing the performance of the model is still being considered for future studies on the matter with a belief and aim of better scores. For one, the relatively lower scores at the stance classification still might be resulted from existence of three categories at this stage, although the all of them perceived as the classes of the same assessment question of stance. Deriven from both the success of the model at the first two phases and also the relatively higher precision scores of pro and anti categories in the third phase despite low f1 score, suggests a possibility of that it can provide better results if the task also narrowed down to the two category classification. One approach might be to integration of determination of neuter tweets within the account type classification. Since the order of the first two phases has been switched in the second testing period and the irrelevant tweets has already been eliminated

by the start of account type assessment in this revised pipeline, rearranging prompt of this second phase to filter all neuter tweets which would also inherently include non-personal since non-personal accounts mostly preserve a neutral tone can be still work with high performance for the second phase, leaving the last phase with a relatively easier task of distinguishing pro tweets from the anti ones.

Besides the evaluation of success of the model in terms of correctly labeling the tweets, there are also few different perspectives that are considered and failed or still can be considered, in terms of the structure of code and approach held in the operationalization of API calls to interact with GPT models. This different approaches creates considerable differences in the computational and financial cost of the tasks due varying ways and amounts of tokens processed through tasks. For one, prompt-caching which is in fact perfect for tasks where the prompts with an exact matching prefix, in my case the evaluation instructions, are being send iteratively. It is, however, found to be inapplicable in the case of this project after a series of tries, since it requires cached prompt to be 1024 tokens or in its incremented sequence, while the prompts of the three phases of this project ranges from 100 to 150 tokens. Another approach, is to concatenate not one but 20 or 30 tweets to my static instruction prompt in each loop and process tweets in batches instead of one by one assessment. It is an efficient approach not only for processing the tokens of instructions part of the prompt 20-30 times less, but also makes much less API calls, affecting the financial cost. It, however, also cannot fully be applied in the project despite several good results had with smaller portions of testing data. It created unmatched length of responses, in other words labels, with the number of tweets processed, when it is tried to be used on larger set of data. It, however, an approach that is worth still more effort of debugging for future study. Lastly, there is a specific way of making API calls to GPT called batch API, that works different from the in-code batching of tweets explained above. It requires preparing of a batch file in jsonl format, sending it via API and it returns the classified data back in 24-hours. Although it is a considerable approach for cost efficacy with large datasets, for a case that requires three consecutive classification phases, with performance evaluation in between of the three and processing of all two times, one for testing and one for the final data is found to be not efficient with the limited time had. Also, it requires some more research and further understanding of its application.

After completion of classification task, that is applied to 1200 more tweets that is filtered down to 624 tweets with relevant and personal account type are added to the testing data in hand and the a logistic regression analysis is conducted with a total of 822 tweets. Two separate model was used with dummy dependent variables of being an anti-boycott tweet or not for the model 1 and of being an pro-boycott tweet or not for the model 2. The dependent variables are also included in binary format for reflecting the specified emotion or not. It is found that the emotions of disapproval, surprising and disgust significantly predicts the pro-boycott stance, while the anger and sadness do not have a significant impact. In the second model however, anger observed to significantly predict the anti-boycott stance again with the disapproval sentiment.