Behaviour Simulation Challenge

A Report Submitted in Fulfillment of the Requirements for the Mid Prep Submission

by

Team 79

Inter IIT Tech Meet 12.0

December 2023

Chapter 1

Approach

Our team has bifurcated into two subteams to efficiently tackle the tasks outlined in the mid-prep problem statement (PS). One subteam is focused on behaviour simulation, while the other is dedicated to content generation. We will elucidate our methodologies for both tasks in the subsequent sections.

1.1 Behaviour Simulation

The project commenced with an extensive literature review, delving into the Hugging Face transformer library and acquiring a comprehensive understanding of fine-tuning open-source Large Language Models (LLMs) accessible online. After assimilating sufficient knowledge, our initial strategy involved a rudimentary solution amalgamating Natural Language Processing (NLP) and regression. This methodology encompasses stopwords elimination using the Python NLTK library, Stemming words, and subsequent employment of Word2Vec, and concluded with neural network training to predict likes based solely on tweet content. Regrettably, this approach yielded unsatisfactory results, prompting us to transition to an alternative method detailed in the reference papers provided within the problem statement. Later on, we switched to using the BERT model for regression and finally arrived at fine-tuning GIT-LLM.

1.2 Content Generation

The project is initiated with an extensive literature review to grasp the nuances essential for addressing the problem statement. Understanding the dynamics of generating tweet text from metadata—company details, usernames, media URLs, and timestamps—formed the basis of our exploration.

In pursuit of a model accommodating both video and audio inputs, XInstructBLIP was initially utilized (it uses different architectures to process both audio and video). But being a very recent contribution to the LAVIS library, it lacked documentation to fine-tune custom datasets, and our attempts at adding a dataset builder for it failed. Later we found MiniGPT and MiniGPTv2. We did not have the hardware requirements to fine-tune the latter, while the former was taking more time than we had in our hands. We tried to slash the dataset (it is not equally distributed among all the companies and has 220 brand names). Then we found InstructBLIP, built upon BLIP2, but it threw errors while fine-tuning.

For MiniGPT4 and InstructBLIP, our strategy involved transcribing audio from videos for potential input of video into the model in the form of prompts. However, we faced challenges incorporating conditional data into the model beyond using prompts—a method primarily employed for conditional generation. Due to resource constraints, we used the pre-trained Llava 13B model and tried it with detailed prompts including the date and the company name and it gave much better results. Some of the results are included in the result document. If we can achieve better results we'll include it in the presentation.

1.3 Dataset Preparation and Preprocessing

The dataset consists of timestamped tweets containing content, usernames, media URLs, and inferred company details. The primary focus for predicting engagement is the textual content of the tweets. The Dataset was prepared by combining the content, username, date, inferred company and captions of the images which were gathered by using the BLIP2 model on the URLs of the images. Finally, we finetuned the GIT-LLM model following finetuning GIT-LLM.

1.4 Tokenization using HuggingFace Models

After preprocessing, the cleaned tweet content was tokenized using advanced NLP models such as BERT and DistilBERT from HuggingFace. Tokenization breaks down the text into smaller units (tokens), effectively representing the text in a format that machine learning models can comprehend. This step generates a tokenized representation of the tweet content, laying the groundwork for subsequent modeling.

1.5 Dataset Creation and Model Training

The tokenized tweet content and the target variable (number of likes) form the training dataset. A Trainer, configured with a learning rate of 1e-5, facilitates the model training process. The choice of learning rate and training epoch aims to balance model convergence and computational efficiency.

1.6 Evaluation and Performance Metrics

Once the model is trained, it is evaluated using a validation dataset. The evaluation metric employed to gauge the model's performance is the root mean squared error (RMSE), which measures the average squared difference between predicted and actual likes for the tweets in the validation dataset. After scaling the likes, an RMSE of 0.7 was obtained, indicating a low deviation between predicted and observed likes.

1.7 Challenges Faced

A significant challenge encountered throughout this methodology implementation pertains to resource limitations, including constraints in memory resources such as RAM, CPU, and GPU. These limitations directly impact the time required for model training, posing obstacles in scaling up the analysis and hindering overall efficiency.

1.8 Results

Task 1

The predictive model for Task 1 is aimed at estimating user engagement (likes) based on tweet content. The RMSE score achieved was 9244.387, indicating the degree of deviation between predicted and observed likes. After scaling the likes, an RMSE of 0.7 is obtained, indicating a low deviation between predicted and observed likes.

1.8.1 Task 2

The LLama Model was successfully able to generate good prompts using the company data and the date.

1.9 Inferences and Conclusions

1.9.1 Behavior Simulation:

In the context of predicting user engagement (likes) from tweet content, the assumption lies in the premise that the textual content, alongside metadata such as company details, usernames, media URLs, and timestamps, significantly influences user interactions.

1.9.2 Content Generation:

Regarding tweet content synthesis from metadata, the inference revolves around the assumption that incorporating video inputs, alongside textual elements, contributes to a more comprehensive understanding of tweet generation. Additionally, there's an assumption that leveraging prompt-based conditional generation remains a primary method for incorporating conditional data within the model.