Adobe Behaviour Simulation Challenge

Team 15

Abstract

This document contains the approach used by Team 15 to solve the Behaviour Simulation Challenge given by Adobe. The problem statement is based on the fundamental process of communication in marketing, focusing on behavior simulation (Task-1) and content simulation (Task-2) to assist marketers in estimating and enhancing user engagement on social media platforms. The team aims to provide solutions for simulating user behavior and creating content that effectively elicits the desired KPIs, offering valuable insights for optimizing social media strategies.

1 Introduction

Companies often rely heavily on leveraging the power of social media platforms such as Twitter, Instagram, Facebook, etc. to promote their products, boost sales, and build a brand identity. Thus, it is important for companies to optimize their posts to maximize user interaction and ensure efficiency in their campaigns.

User engagement on Twitter is quantified by metrics like user likes, retweets, comments, mentions, follows, clicks on embedded media and links. For this challenge, we consider the number of likes to be our KPI for user engagement.

The given problem statement consists of the following two tasks:

- **Task 1:** Given the content of a tweet, the task is to predict its user engagement, measured by likes.
- **Task 2:** Given the tweet metadata, generate the tweet text.

The team has leveraged the latest advancements in Deep Learning and Generative AI to tackle the above two tasks. The approaches used are described in detail in the proceeding sections of this report.

2 Task 1

2.1 Objective

Given the content of a tweet (text, company, username, media URLs, timestamp), the task is to predict its user engagement, measured by likes.

2.2 Data Processing

- The presence of three modes of visual data present a need for an efficient pipeline which can handle all three forms.
- At the same time, the sheer volume of video/GIF data makes using traditional methods of extracting features and embeddings practically unusable.
- After experimenting with numerous methods, we found out that only using the thumbnail of the video as representative of the entire video gave satisfactory results, allowing us to treat videos and GIFs as images. This not only simplified the pipeline leading to reduced inference time, but also gave efficient and accurate results, within reasonable limits of error.
- To further reduce image processing time and simplify the pipeline, we leveraged the fact that the image and its description are correlated. This allowed us to continue with the approach of converting all media to a language format, which could enable us to process it together with text.
- For posts containing broken links, we used black image as media. This allowed smooth pipeline integration without affecting other parameters.
- The captions for images and video/GIF thumbnails were generated using the vision-language model BLIP-2. The BLIP-2 model was chosen because of its significantly reduced training time and resource requirements, and state-of-the-art performance on Image-Captioning task.
- The remaining fields of time and company were concatenated with the media to create an instruction dataset. This instruction dataset was structured as a prompt for an LLM

2.3 Modelling

- Considering the skewness of the dataset, using LLMs directly for regression was prone to errors.

- To reduce skewness in the number of likes, natural logarithm was applied to the dataset.
- The team then divided the tweets into 5 different buckets, based on log likes and posed the LLM training problem as a classification task.
- Embeddings are extracted from the instruction dataset created in the previous step using DistilBERT. This model allows for faster training and inference, reduced memory overhead, is less prone to overfitting, and gives accurate results on text classification.
- A fully-connected layer followed by a softmax layer is added to DistilBERT, and the instruction data created in the previous step is used to train the entire setup.
- Once a tweet has been assigned a bucket, a separate regressor is trained for each bucket using the embeddings generated by DistilBERT.

2.4 Buckets and Regressors

The bucket sizing was determined by observing the characteristics in the histogram of log likes. The regions identified through the buckets appear to follow distributions which can be modelled by traditional machine learning methods.

All posts are assigned one of five buckets based on the log likes they have:

- Bucket 1: 0 - 2
- Bucket 2: 2 - 3.75
- Bucket 3: 3.75 - 6.25
- Bucket 4: 6.25 - 11
- Bucket 5: > 11 (Outliers)

Multiple regression models were fit on each bucket. The best results were given by the following regressors:

- Regressor 1: XGBoost
- Regressor 2: XGBoost
- Regressor 3: CatBoost
- Regressor 4: CatBoost
- Regressor 5: CatBoost

2.5 Results

- Average Inference Time: <1 s
- Bucket Classification Accuracy: 74.3%
- RMSE on Training Data: 3812.6

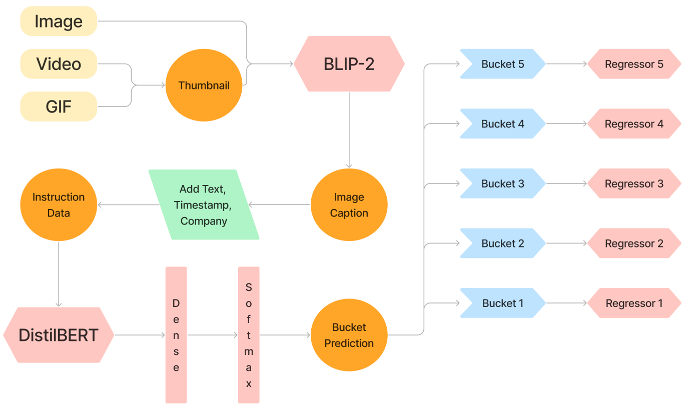

Figure 1: Data Pipeline for Task 1.

3 Task 2

3.1 Objective

Given the tweet metadata (company, username, media URL, timestamp), generate the tweet text.

3.2 Data Processing

The issues faced in this task are similar to the issues discussed in Task 1. However in this task, the absence of caption content implies that the importance of media features increases. Therefore, it is important to establish that representing videos/GIFs as their thumbnails does not lead to a large loss in meaningful data, if we want to tackle the problem in a similar way as done in Task 1. This was established using the following experiment:

- A subset of data was randomly sampled from the training dataset, with all samples having videos and GIFs as their media. Only the first 2 seconds were considered, because studies show that the average user only watches media upto 2 seconds
- We used a pre-trained mPLUG-2 model for generating captions, because of its multimodal capabilities, and used those captions to generate our prompt.
- Since we had not fine-tuned any LLM at that point, we fed that prompt into OpenAI's GPT-3.5 and compared the cosine similarity of the generated text.
- We observed that on an average, the cosine similarities were greater than 0.6.
- This justifies that using thumbnail as representation of video/GIF data is better than using the complete video, because it provides a reasonably accurate representation while saving on a lot of computational power.

Thus, with this result in mind we built the data processing pipeline as follows:

- Since we can treat videos and GIFs as images, we built a similar pipeline to the one described in Task 1 and generated image captions using BLIP-2.

- It was observed that in many cases, a section of the post content was present in the image as text.

- To extract that critical piece of information, we also passed the image through Paddle OCR, which gave the most accurate results with lowest latency. The outputs of the OCR were concatenated with the caption generated using BLIP-2.

3.3 Modelling

- The remaining data such as time and company, were concatenated with the captions obtained in the previous step to form an instruction dataset.

- This instruction dataset was structured as a prompt for an LLM, which is used to generate the post content.

- The LLM used is Llama-2, an open-source LLM developed by Meta AI. Llama-2 offers increased performance in text generation, and comes in various sizes allowing us to achieve better performance using a smaller model, reducing training and inference time. Llama-2 is then fine-tuned on the instruction data.

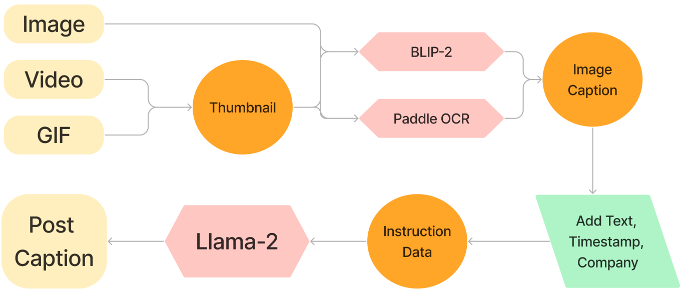

Figure 2: Data Pipeline for Task 2.

3.4 Results

- Average Inference Time: 5.4 s

- BLEU Score: 0.004

References

[1] J. Li, D. Li, S. Savarese, and S. Hoi, “Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models,” 2023.

[2] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, “Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter,” 2020.

[3] A. Khandelwal, A. Agrawal, A. Bhattacharyya, Y. K. Singla, S. Singh, U. Bhattacharya, I. Dasgupta, S. Petrangeli, R. R. Shah, C. Chen, and B. Krishnamurthy, “Large content and behavior models to understand, simulate, and optimize content and behavior,” 2023.

[4] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, D. Bikel, L. Blecher, C. C. Ferrer, M. Chen, G. Cucurull, D. Esiobu, J. Fernandes, J. Fu, W. Fu, B. Fuller, C. Gao, V. Goswami, N. Goyal, A. Hartshorn, S. Hosseini, R. Hou, H. Inan, M. Kardas, V. Kerkez, M. Khabsa, I. Kloumann, A. Korenev, P. S. Koura, M.-A. Lachaux, T. Lavril, J. Lee, D. Liskovich, Y. Lu, Y. Mao, X. Martinet, T. Mihaylov, P. Mishra, I. Molybog, Y. Nie, A. Poulton, J. Reizenstein, R. Rungta, K. Saladi, A. Schelten, R. Silva, E. M. Smith, R. Subramanian, X. E. Tan, B. Tang, R. Taylor, A. Williams, J. X. Kuan, P. Xu, Z. Yan, I. Zarov, Y. Zhang, A. Fan, M. Kambadur, S. Narang, A. Rodriguez, R. Stojnic, S. Edunov, and T. Scialom, “Llama 2: Open foundation and fine-tuned chat models,” 2023.

[5] Y. Du, C. Li, R. Guo, X. Yin, W. Liu, J. Zhou, Y. Bai, Z. Yu, Y. Yang, Q. Dang, and H. Wang, “Pp-ocr: A practical ultra lightweight ocr system,” 2020.

[6] D. Li, J. Li, H. Le, G. Wang, S. Savarese, and S. C. Hoi, “LAVIS: A one-stop library for language-vision intelligence,” in *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, (Toronto, Canada), pp. 31–41, Association for Computational Linguistics, July 2023.

[7] V. Khurana, Y. K. Singla, J. Subramanian, R. R. Shah, C. Chen, Z. Xu, and B. Krishnamurthy, “Behavior optimized image generation,” 2023.