Behavior and Content Simulation for Social Media Marketing

Team 69

15 December 2023

Abstract

This report presents our comprehensive analysis and solutions for the challenges posed by Adobe Digital Experience Business in the tasks of Behavior and Content Simulation. Our goal was to predict user engagement on social media content (Task-1) and generate compelling tweet text (Task-2) for enhanced marketing strategies.

1 Task-1: Behavior Simulation

1.1 Approach

In the context of media content processing, our approach encompasses a thorough analysis of diverse media types, encompassing both images and videos, to facilitate predictive modeling. For image processing, we employ the Vision Transformer (VIT) technique for feature extraction, specifically utilizing the VIT-base-patch-16 model. Additionally, to ensure uniformity and comparability, we implement standardization procedures by adjusting the image resolution to a standardized 256x256 dimensions through the application of random cropping. For media containing temporal dimensions, processing closely mirrors the steps applied to images in our predictive modeling pipeline. This involves extraction of 32 frames and applying PCA (Process Similarity to Images) to resize the feature to 256*256, contributing to the richness of our dataset for subsequent analysis and predictive modeling.

$$Fv_{(32,256,256)} = EF(V_{(\mathrm{t},256,256)}, 32)$$
$$Fv'_{(256,256)} = PSI(Fv_{(32,256,256)})$$

where Fv represents Extracted frames and EF represents a function to extract frames. V represents media with temporal dimension, Fv' represents interpolated feature vector of size 256*256 PSI represents Process similarly to images. If multiple images with varying resolutions exist, the model prioritizes the image with the lowest resolution for uniform processing. In instances of different media files, feature vectors from all images are stacked, introducing a third dimension for comprehensive representation.Textual features are derived through tokenization, utilizing the pretrained BERT ('bert-base-uncased') tokenizer.

$$t = Tokenizer_{\mathbf{BERT}}(Text)$$
$$Ft = BERT_{\mathrm{Model}}(t)$$

where t represents the tokenized text and Ft represents the textual features extracted by BERT model. BERT pretrained model ('bert-base-uncased') was used to extract the features after tokenizing the text. Figure 1 represents the complete architecture of feature extraction.

1.1.1 Model Architecture:

All the frames of visual_features are passed through the CNN and added up to give a feature vector of size 256*256. The textual feature and visual features are passed through Cross-Attention so at to get the final feature of size 256*256 having feature relating the text and media. The output is then multiplied element-wise by the heatmap value h of the data item.

$$Fv_{(256,256)} = \sum_{i=1}^{n} CNN(f_i) \tag{1}$$

where f_{i} represents each frame and $CNN(f_{\mathrm{i}})$ denotes the output of convolution of each frame, n is the number of frames. The final feature is passed through a regression model having 3 linear layers followed by ReLU.

$$F_f = CrossAttention(Ft, Fv)$$
$$O = F_f * h$$
$$Output = ReLU(Linear_3 ReLU(Linear_2(ReLU(Linear_1(O)))))$$

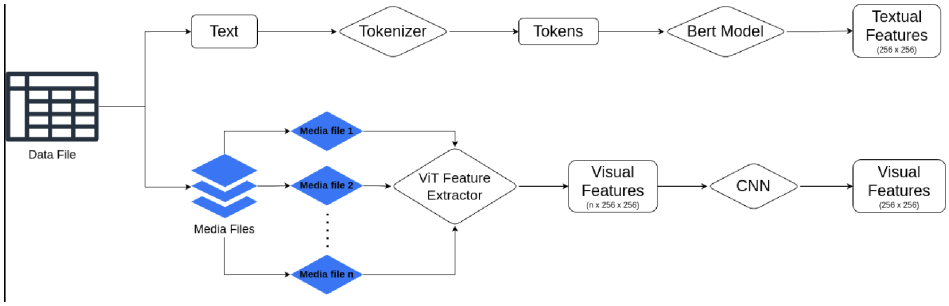

Figure 1: Feature Extraction

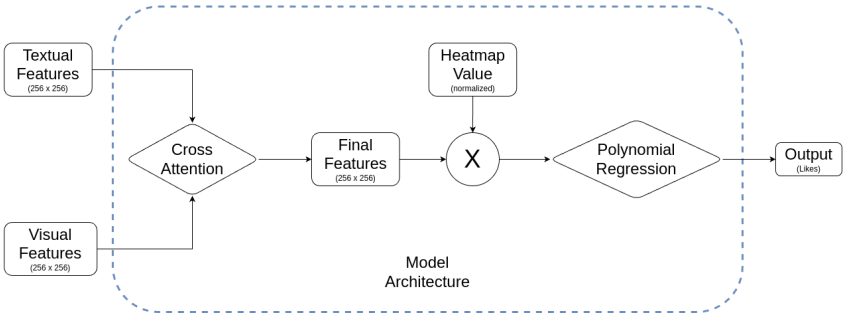

Figure 2: Model Architecture

1.2 Results

The model demonstrated commendable performance with an achieved RMSE of 3.2 on logarithmic scale on validation dataset.

1.3 Enhancements and Iterative Refinements

Our initial model prioritized textual features, and future iterations will strategically integrate visual features using cross-attention mechanisms. We plan to implement a systematic company classification system using APIs from Google, Twitter, and Wikipedia, tokenizing predefined categories' descriptions and incorporating them into our model. Furthermore, to enhance likes prediction, we propose incorporating emoji sentiment analysis using NLTK's SentimentIntensityAnalyzer on the content column, providing a more comprehensive and accurate predictive model.

2 Task-2: Content Simulation

2.1 Approach

The model creation process involves several steps, including image analysis using a pre-trained convolutional neural network (CNN), building a dataset vocabulary, creating a word embedding with the word2vec (Continuous Bag of Words) algorithm, and training a Long Short-Term Memory (LSTM) neural network. Utilizing pre-trained models from torchvision, specifically GoogLeNet trained on ImageNet.

$$M = GoogLeNet(ImageNet)$$

where M represents the pre-trained model, specifically GoogLeNet trained on ImageNet. The image preprocessing phase involves iterating through the images in the dataset, applying the GoogLeNet creators' specifications for preprocessing, and obtaining latent variables (L) representing the processed images. The video thumbnail is incorporated as a feature in the model, and the processed image file paths along with corresponding 1024 latent variables are written to a CSV file (C).

$$C = W(P, L)$$

where C represents csv file containing image file paths P and latent variables and W represents writing operation to csv. Creating a dataset vocabulary includes parsing the caption structure from the 'Flickr 8k.token.txt' file, splitting the dataset into training and validation sets, building a dictionary of words, tokenizing content, removing infrequent words, sorting words by frequency, and writing the vocabulary to a file. Word embedding with Word2Vec (CBOW) involves implementing the algorithm, creating a PyTorch Dataset, designing a PyTorch model to predict the center word from context words, and training the word embedder. The resulting word embedding weights matrix is saved to disk. The LSTM neural network is trained using the latent representation of an image and the beginning of a caption as inputs. The output is the next word in the caption, and the trained LSTM model is employed to generate tweet text based on tweet metadata.

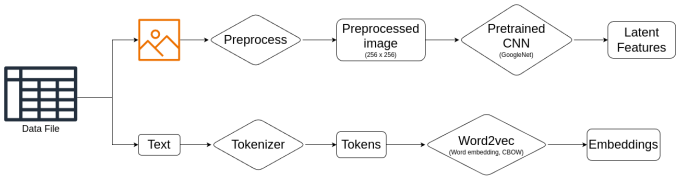

Figure 3: Feature Extraction

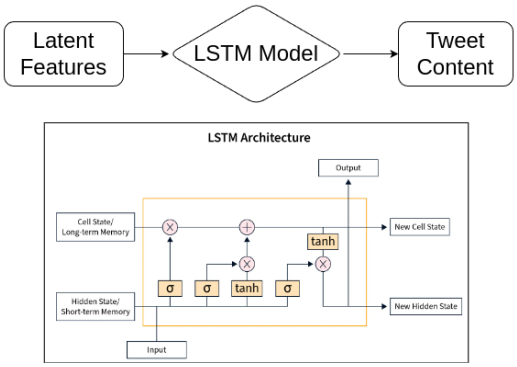

Figure 4: Model Architecture

2.2 Enhancements and Iterative Refinements:

Our initial model focused on utilizing available image features, primarily using single images and thumbnails from video content due to time and resource constraints. To enhance our approach, we plan to incorporate latent representations of videos and consider all images in a Twitter post, creating a unified feature representation. We aim to explore multimodal fusion techniques, including late fusion or attention mechanisms, to effectively blend visual and textual data and capture intricate relationships. Additionally, we propose allowing dynamic caption lengths during training, augmenting the dataset for diversity, exploring transfer learning for word embeddings, implementing real-time data integration for adapting to trends, and extending the model to incorporate emoji generation for more contextually relevant tweet text.

3 System Specifications

64 GB DDR 4 RAM, Intel I9 12th generation, 8 GB Nvidia RTX-3070 GPU.

4 Conclusion

In conclusion, our approach to Behavior Simulation successfully addressed the tasks outlined by Adobe Digital Experience Business. Unfortunately due lack of computational resources and time limitations we were not able to completely train the LSTM model for Content Simulation. However we have explained our approach briefly and were able to implement it.

5 Appendix

6 References

GitHub Repo: Link 1

Google Vision Transformer: Link 2

Image Captioning with PyTorch LSTM: Link 3

BERT 101: Link 4

Heatmap CSV: Link 5

GoogleNet (InceptionV1) with TensorFlow: Link 6

Word2Vec Explained: Link 7