Gesture Synthesizer - Generating gestures based on input text

Hongxiang Zhang*
Qizhou Fang*
Wenru Feng†
University of California, Davis
Davis, California, USA

ABSTRACT

The gesture is one of the important factors in communication. However, not everyone is comfortable or capable of expressing themselves through gestures. Our project aims to create a simple gesture synthesizer that generates gestures based on input text, allowing individuals to enhance their communication skills and express themselves more effectively.

To tackle this issue, we design a deep-learning-based gesture synthesizer. We design our workflow into two parts. 1. Input text process: We utilize the Bidirectional Encoder Representation from Transformers which is the state of art natural language processing model to process the input text, understanding the inner emotions. Specifically, the system will classify the user input into several emotion categories for further processing. 2. Gesture Generation: We used Maya to create a gesture model. And the system will generate the appropriate gesture video based on the input text and display it on the screen.

With this project, we provided a simple and effective solution for individuals who struggle with expressing themselves through gestures. By introducing the state-of-art NLP model, our system can understand the input text and express the gesture precisely.

KEYWORDS

animation, sentiment analysis, natural language processing

ACM Reference Format:
Hongxiang Zhang, Qizhou Fang, and Wenru Feng. 2023. Gesture Synthesizer - Generating gestures based on input text. In *Proceedings of (Course project).* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

As digital technologies advance, creating immersive, expressive, and user-oriented interactions becomes increasingly important. While substantial progress has been made in areas such as speech recognition, facial recognition, and text analysis, a key component of human communication—gestures—has been relatively under-explored. This report details the research and development of an innovative gesture synthesizer, a system designed to enhance digital communication through synthetic gestures, underpinned by a deep understanding of the emotions conveyed in text.

Our gesture synthesizer system operates through a three-step process. The first step involves leveraging a state-of-the-art natural language processing (NLP) model, the Bidirectional Encoder Representation from Transformers (BERT). The model was trained to process and classify input text into one of five emotion categories: happy, sad, angry, fearful, and neutral, achieving a commendable classification accuracy of 86 percent. BERT is unique in that it uses a bidirectional translator, allowing it to understand the context of a word in terms of its entire environment, as opposed to the traditional one-way perspective.

The second step consisted of generating the corresponding gestures and some facial expressions, realized with custom models in Maya. This process results in a video of the appropriate gesture being displayed on the user interface, effectively mapping the recognized emotion to a visual non-verbal expression.

The third step made a user-friendly Graphical User Interface (GUI) created using the Tkinter library, providing a simple point of interaction for the user. It can validate the input through the first step, mapping the input to a corresponding gesture video, which are made through maya from the second step, then managing the concurrent video playback using threading, and displaying the gesture video to the user.

This project introduces a gesture synthesizer system aimed at combining textual sentiment analysis with computer-generated gestures. In other words, the Graphical User Interface (GUI) can provide videos with different emotions by checking the user's input, whether it is a word or a sentence.

2 RELATED WORKS

2.1 Sentiment analysis

Sentiment analysis aim to understanding and extracting sentiments, attitudes, opinions, and emotions expressed in text. It involves analyzing and determining the subjective information present in textual data and classifying them into emotion categories. Techniques proposed for sentiment analysis can be categorized into two groups: (1) traditional models and (2) deep learning models. While most recent studies utilize machine learning methods to train models on labeled datasets to automatically learn patterns and features that indicate sentiment.

When evaluating the performance of a single method on a specific dataset within a particular domain, the outcomes indicate a relatively elevated overall accuracy [1, 2, 4, 6] for Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN). Qian [11] demonstrated the efficient behavior of Long Short-Term

*
Hongxiang contributed to the implementation of sentiment analysis model, report formatting, and report writing on abstract, discussion, and sentiment analysis in related work, and methodology.
Qizhou contributed to animation generation, report writing on the introduction, and animation generation.
Wenru contributed to software framework implementation, report writing on software implementation, and conclusion.

Course project, June, 2023, Davis, CA, USA

Table 1: Count of each emotions in dataset

Emotions	Number of data
Joy	8283
Sadness	7564
Anger	4693
Fear	4332
Neutral	2254

Table 2: Model training details

Epoches	Time(s)	Train loss	Train accuracy	Val loss	Val accuracy
1	2003	1.0819	0.5603	0.6559	0.7684
2	2029	0.4170	0.8558	0.4853	0.8288
3	1976	0.2469	0.9162	0.4240	0.8514
4	2030	0.1601	0.9461	0.4268	0.8587
5	2030	0.1115	0.9631	0.4306	0.8632

Memory (LSTM) when applied across various text levels. The introduction of the latest large-scale pre-trained model, BERT[5], has made a groundbreaking impact, surpassing previous machine learning models in most Natural Language Processing (NLP) domains.

3 METHODOLOGY

3.1 Sentiment Analysis

In initiating the workflow of our study, we designed a sentiment classifier aimed at categorizing the input text into five distinct emotional states: joy, sadness, anger, fear, and neutrality. In an effort to improve the classifier's performance, we amalgamated four datasets, composed of dialogues, messages, and short narratives. The model employed for this purpose was the advanced BERT pre-training model, which has demonstrated superior outcomes in the field of semantic analysis. Upon evaluation, our model was capable of accurately discerning the emotional context within the input text, achieving a classification accuracy rate of 86%.

3.1.1 Dataset and data preprocessing. The constructed dataset, utilized for this research, was sourced from four distinct data repositories: DailyDialog, ISEAR, Emotions Dataset for NLP, and Emotion-Stimulus[7, 9, 10, 12]. This facilitated the creation of a large dataset, comprising five labels: joy, sadness, anger, fear, and neutrality. Table1 illustrate the composition of each data labels. The complete dataset consisted of 27,226 text entries, primarily short messages and dialogue utterances. The data was partitioned into training, testing, and validation sets, with an allocation ratio of 70%, 20%, and 10%, respectively.

Considering the BERT model's constraints in processing texts longer than 512 characters and our computational resources limitations, we set a maximum length of 350 characters for the texts. Any text exceeding this threshold was truncated. Moreover, to circumvent the risk of the model learning input patterns, we added shuffling of the dataset after each training epoch.

3.1.2 Bert model and training. Driven by the recent advancements in Natural Language Processing (NLP) and machine learning frameworks, namely Bidirectional Encoder Representations from Transformers (BERT) and ktrain[5, 8], we incorporated these into our research methodology. Our model was trained over the span of five epochs, utilizing Google's Colab as the platform for approximately six hours. This procedure culminated in an impressive 86% accuracy on the validation set. The architectural foundation of BERT is predicated on the transformer model architecture, known for its multi-head self-attention mechanisms, position-wise fully connected feed-forward networks, and normalization layers. Unique to the BERT model is the utilization of bidirectional transformers, enhancing its ability to perceive the context of a word by taking into account its entire surrounding context, as opposed to a unidirectional perspective. In order to calibrate the output to the corresponding labels, we introduced an additional normalization layer to the pre-trained model architecture. This adaptation was pivotal in delivering optimal performance for our text classification task.

Inspired by the recent advance in nature language processing and machine learning framework[5, 8], We applied ktrain and Bidirectional Encoder Representations (BERT) to build our model. We trained our model 5 epoches in Colab for around 6 hours. Achieving validation set accuracy up to 86

The architecture of BERT is based on the transformer model architecture [13] with multi-head self-attention mechanisms, position-wise fully connected feed-forward networks, and normalization layers. Where transformer in BERT applied a bidirectional transformer, which allows the model to understand the context of a word based on all of its surroundings. Following this pretrained model, we add one more fully connected layer with sigmoid activation function for multi-classicafication to mapping the output to labels.

We trained our model on a 12G memory GPU with Learning rate 6e-6 and 5 epoches for around 6 hours, as illustrate in the table2 we record the training set loss, training set accuracy, validation loss and validation accuracy. It's clear to see that both training loss and validation loss have a trend of decreasing. Besides, look into the training accuracy and validation accuracy, with the increase of training accuracy, the increase rate of validation accuracy is slowing down, comparing the epoch 4 and 5 validation accuracy only improve 0.5%, thus we stopped training to avoid overfitting.

3.1.3 Evaluation. As shown the table 3, we measure our result in 4 matrics, precision, recall, F1-score and confusion matrix. Where precision calculate the percentage of correctly predicted positive observations out of the total predicted positives. Recall measure the percentage of correctly predicted positive observations out of the total actual positives. F1-score is the harmonic mean of Precision and Recall and tries to balance the two [14]. And inconfusion matrix each row shows instances in an actual class while each column represents the instances in a predicted class.

As illustrate in the table 4, class 'joy' achieve best performance in all four matrics. Upon closer inspection of the confusion matrix, it is observed that a significant number of misclassified instances assigned to "joy" were wrongly labeled as "Neutral," while the same misclassification occurred for the "Neutral" label. This indicates that the model struggles to distinguish between "joy" and "Neutral." Moreover, both "joy" and "Neutral" emotions may manifest in similar expressions, leading to this overlap. Similar instances

Table 3: Evaluation on each emotion class

Emotion	Precision	Recall	F1-Score
Joy	0.91	0.91	0.91
Sadness	0.87	0.86	0.87
Anger	0.86	0.86	0.86
Fear	0.83	0.82	0.83
Neutral	0.79	0.84	0.81
Weighted Avg	0.86	0.86	0.86

Table 4: Confusion matrix on each emotion class

Emotions	Joy	Sadness	Fear	Anger	Neutral
Joy	1270	25	17	20	70
Sadness	29	1086	44	67	31
Anger	24	42	776	49	12
Fear	26	74	44	795	29
Neutral	44	20	17	23	534

of multi-emotion occurrence are also observed in the "sadness" and "anger" categories. One potential solution to address this challenge is the utilization of fuzzy logic, employing multiple labels to represent a single instance.

3.2 Animation generation

We created 5 video clips for the 5 gestures we are synthesizing. After the input text is classified as one of the five sentiments, we play the corresponding video in the user interface. The animation was done in maya, we used the character model Mery from the mery project. In setting up the poses for these gestures, we referenced example photos of women expressing each of the sentiments. Facial control for our character is assisted by the add-in "MG-Picker". We then utilized the playblast functionality in maya to generate videos of the animation. The movement of the character "Mery " was captured by a camera we freezed at a certain position.

3.3 Software implementation

3.3.1 Creating the GUI. the application starts by setting up a GUI using the Tkinter library. It offers an entry field for the user input, and a button to generate the corresponding gesture video. There is a text hint by the 'ToolTip' class to guide users about valid input options.

3.3.2 Input Validation and Processing. when a user inputs an emotion in the text field and clicks the 'gesture' button, then the application validates the input and checks whether the entered emotion is part of the pre-defined list of five emotions. Then it will show an error message by Tkinter's messagebox if the input is invalid.

3.3.3 Video Mapping and Playback. if the input is valid, it is mapped to a corresponding video by a pre-defined dictionary. The application uses Python's threading model to play the video. The OpenCV library is for video handling.

3.3.4 Concurrent Execution. the application uses a threading event object to handle concurrent video playback. When a video already exists and a new video needs to be played, the application first stops the video, which already existed and played, before starting a new video. The Event object is used to signal the running thread to stop.

3.3.5 Displaying the Gesture Video. the asked video is played in a Tkinter label on the GUI. The video playback happens in a separate thread and updates continuously until the video stops or a new video is asked to play.

4 DISCUSSION

Though our model perform well during test and evaluation, there still some drawbacks and potential works that can be done.

One prospective avenue for further investigation is long sentence emotion classification, since BERT can only deal with sentence no longer than 512 words, which results in misclassification when longer sentences are encountered. One possible solution would be split a long sentence into smaller segments and process them individually, another approach would be utilize more advanced models that designed for longer sentence. E.g. Transformer-XL [3]

Additionally, as previously mentioned, one sentence can express multiple emotions, which means it is not atomically classified. In this case, we can utilize Fuzzy logic, allowing the measurement of emotions in terms of percentage, enabling a more accurate and nuanced representation that aligns closely with human expression.

5 CONCLUSION

Our investigations into sentiment analysis using natural language processing techniques such as BERT and ktrain yield profound findings and promising results. This achievement highlights the effectiveness of using advanced pretrained models such as BERT for sentiment analysis tasks. The model was trained on a GPU with 12GB of memory, limiting the text length to 350 characters to keep it computationally feasible. Furthermore, the data preprocessing method combines four different datasets and contains multiple forms of text data, emphasizing the importance of rich and diverse datasets for training NLP models. On the other hand, evaluation of our sentiment analysis model shows great results in all four metrics used to evaluate its performance (precision, recall, F1-score, and confusion matrix).

Secondly, we used the "Mery" character model from the Mery Project through Maya to animate these emotions, posing in different poses based on reference photos of women expressing them. We employed the "MG-Picker" plugin, which significantly simplifies the facial control process for our characters, allowing for nuanced emotional characterization.

Thirdly, The system operates by Python through a Graphical User Interface (GUI) created using the Tkinter library, and OpenCV library facilitates video processing, managing concurrent execution of videos. In order to generate the corresponding video by user's input.

In summary, when the input is received, the system validates emotions against a list of five predefined categories: happy, sad, fearful, angry, and neutral. Valid input initiates the gesture synthesis process, utilizing a dictionary for video mapping and Python's threading model for video playback.

REFERENCES

[1] Fazeel Abid, Muhammad Alam, Muhammad Yasir, and Chen Li. 2019. Sentiment analysis through recurrent variants latterly on convolutional neural network of Twitter. *Future Generation Computer Systems* 95 (2019), 292–308.

[2] Ahmed Sulaiman M Alharbi and Elise de Doncker. 2019. Twitter sentiment analysis with a deep neural network: An enhanced approach using user behavioral information. *Cognitive Systems Research* 54 (2019), 50–61.

[3] Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc V Le, and Ruslan Salakhutdinov. 2019. Transformer-xl: Attentive language models beyond a fixed-length context. *arXiv preprint arXiv:1901.02860* (2019).

[4] Nhan Cach Dang, María N Moreno-García, and Fernando De la Prieta. 2020. Sentiment analysis based on deep learning: A comparative study. *Electronics* 9, 3 (2020), 483.

[5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

[6] Abdalraouf Hassan and Ausif Mahmood. 2017. Deep learning approach for sentiment analysis of short texts. In *2017 3rd international conference on control, automation and robotics (ICCAR)*. IEEE, 705–710.

[7] Yanran Li, Hui Su, Xiaoyu Shen, Wenjie Li, Ziqiang Cao, and Shuzi Niu. 2017. Dailydialog: A manually labelled multi-turn dialogue dataset. *arXiv preprint arXiv:1710.03957* (2017).

[8] Arun S. Maiya. 2020. ktrain: A Low-Code Library for Augmented Machine Learning. *arXiv preprint arXiv:2004.10703* (2020). arXiv:2004.10703 [cs.LG]

[9] Helen O'Reilly, Delia Pigat, Shimrit Fridenson, Steve Berggren, Shahar Tal, Ofer Golan, Sven Bölte, Simon Baron-Cohen, and Daniel Lundqvist. 2016. The EU-emotion stimulus set: a validation study. *Behavior research methods* 48 (2016), 567–576.

[10] PRAVEEN Praveen. 2020. Emotions dataset for NLP. https://www.kaggle.com/datasets/praveengovi/emotions-dataset-for-nlp?resource=download

[11] Jun Qian, Zhendong Niu, and Chongyang Shi. 2018. Sentiment analysis model on weather related tweets with deep neural network. In *Proceedings of the 2018 10th international conference on machine learning and computing*. 31–35.

[12] KR Scherer and H Wallbott. 1990. International survey on emotion antecedents and reactions (isear).

[13] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems* 30 (2017).

[14] Reda Yacouby and Dustin Axman. 2020. Probabilistic extension of precision, recall, and f1 score for more thorough evaluation of classification models. In *Proceedings of the first workshop on evaluation and comparison of NLP systems*. 79–91.