# LSTM-based Semantic Analysis and Facial Expression Generation

1st Dana Aubakirova
*Nazarbayev University*
d.aubakirova@nu.edu.kz

2nd Zhamilya Saparova
*Nazarbayev University*
zhamilya.saparova@nu.edu.kz

3rd Nurdaulet Zhuzbay
*Nazarbayev University*
nurdaulet.zhuzbay@nu.edu.kz

***Abstract*—3D avatars play an important role in conveying information to humans in a natural way. They are widely used in today's online world, in domain of customer service, entertainment, and medical HCI. However, there are certain challenges and problems with the animation of 3D avatars, as it can be complex and time-consuming task. Retrieving emotions from the text can also be a challenging problem in absence of facial expressions or voice modulations. Moreover, there are limited number of research directed to study the field of semantic analysis. The goal of this paper is to optimize the process of 3D avatar animation, by training the LSTM model using the pre-trained GloVe word vectors, and subsequently animating 3D avatars with obtained emotions using the OpenPose for Blender and automated operations in iClone7 real-time animation softwares. In general, it was shown that there is a need to focus on the development of highly accurate emotion detection model while building single pipeline for text-based facial animation, rather than focusing on 3D animation itself. Because the softwares such as iClone7 and Blender come with different add-ons and automated operations that allow for the creation of custom models, assets, and animations alleviating the need for the manual work.**

***Index Terms*—sentiment analysis, 3D avatars, 3D animation, emotion detection, facial expression generation**

## I. Introduction

Facial expression generation on 3D avatars is essential part in various applications, including animated movies [1], teleconferencing, computer games, talking agents, and human-computer interaction [2]. Facial expressions represent different social cues through important facial keypoints, such as eyes, lips, and nose [3]. In this way avatars help to convey important information with corresponding emotions in a natural way [4]. Emotions being basic human trait, have been extensively studied by researchers in the fields of sociology, medicine, psychology etc. Some of the prominent work in categorizing emotions include Ekman's six class categorization: anger, sadness, fear, surprise, disgust and happiness [5]. In this study, we focus on generating five emotions, where three emotions are from Ekman's classification, happiness, sadness, anger, and the other two emotions being excitement and love.

In recent years, traditional facial animation approaches have reached a high level of realism and were tremendously upgraded. However, there are still some challenges and problems that need to be addressed. For instance, active face capture such as web-cameras, motion sensors or markers are expensive and time-consuming to use. Whereas passive techniques, such as facial expression generation based on datasets including captured facial transformations from cameras can be significantly challenged by face occlusion and illumination [6]. Alternatively, research on speech-driven face synthesis has regained attention of the community in recent time. But this approach can also be time-consuming as it requires pre-recorded audio and speech synthesis [7].

Therefore, we address aforementioned problems by considering text-based approach for the emotion detection. Apparently, detecting emotions in text can also be a challenging problem in absence of facial expressions or voice modulations. Even as a human, it is difficult to properly interpret and detect the emotion solely based on the text of the conversation. There are also the difficulty in understanding context, sarcasm, class size imbalance, and rapidly growing Internet slang [8].

However, with the advent of different messaging platforms like WhatsApp, Telegram, and Twitter, it has become essential for machines to understand emotions in textual conversations to respond appropriately [9]. Correspondingly, the applications of emotion detection include and not limited to the social media platforms, the domain of customer service. It has become an essential technology, especially, in the cases when there is a heavy flow of messages and the quick response, or the proper prioritization of messages to respond is needed. For example, responding to an angry message prior to a basic inquiry will significantly increase customer satisfaction.

In this paper, we propose the model based on LSTM that allows for the accurate semantic analysis of the text, which is subsequently used to generate facial expressions on 3D avatars using the advanced 3D animation softwares such as Blender and iClone7. Our work can be divided into two aspects. In the first part, we formulate NLP task as creating a model to detect emotion from text. For that we build an LSTM model that takes as input word sequences. It does not require hand-crafted features and is trained as a single unified model. We trained the model on the dataset consisting of different sentences labeled with corresponding emotions. We preprocessed the dataset such that, each word in the input utterance was represented as the word embeddings using 50-dimensional GloVe word embedding. Based on the observation that GloVe has the best average F1 score among the other models such as Word2Vec, Fast Text, we choose it as our embedding for the Semantic LSTM layer and used a pre-trained word vectors. We evaluate our model on test set, and

some random sentences. In the second part, we focus on using iClone7 and Blender real-time 3D animation softwares to animate the detected emotions on 3D avatars. Using the OpenPose we generate the facial keypoints to represent facial expressions on 3D Avatar in Blender. In addition, we generated facial expressions in iClone7 using the embedded automated operations.

The rest of the paper is organized as follows: Section 2 evaluates the related works. Section 3 describes the dataset, GloVe word embedding and the LSTM model in detail. Our facial expression generation approach, OpenPose with Blender and iClone7, with results are discussed in Section 4. Finally, the conclusion and future work are elaborated in Section 5.

## II. Related Works

There exists a large body of literature for sentiment analysis, and its' translation to the credible facial animation. However, the majority of works are concentrated on using an audio track accompanied with a transcript to provide explicit knowledge about the phoneme content, and subsequently using its' counterpart called visemes for animations. For example, the mapping between phonemes and visemes was implemented in the dynamic visemes model in the recent work JALI [10]. JALI is able to reproduce different speaking styles and emotional states by factoring the facial animation to lip and jaw movements, based on psycholinguistic considerations.

Another promising approach based on the speech audio accompanied by transcript was proposed in [7]. The authors applied deep neural network to learn low-dimensional, latent descriptor that explained the data, instead of using predefined emotion categories. Later, they assigned any number representing semantic meanings such as "sad", "happy" to the descriptors' parameter combinations. The paper compared two baselines: the DI4D system's video-based performance capture and the animation produced by FaceFX. In the experimental study, it was revealed that deep neural network-based model outperformed the second baseline, even though there was an apparent loss in naturalness. However, in this work, there are some difficulties in representing eye motion, which is related to mismatches in the audio correlation.

Some works using the machine-learning approach have focused on reusing captured video frames by concatenating, blending, and warping them [11] [12] [13]. Although these image-based methods can bring realistic results, there is a need to store a large number of frames, which might not be applicable to applications such as VR, games where 3D models must be rendered and animated from different viewpoints.

Among recent works, human perceptions based generative model of facial expressions presented in [14] could be mentioned. To construct their model, firstly, they identified the specific face movements represented as Action Units (AU) [15] to form a valence-arousal space, and secondly, using the cross-correlation they produced an array of facial expressions for six basic and complex emotions.

Finally, Mukashev et al. [4] implemented a model that allows for the generation of humanlike facial expressions based on the semantic analysis of the text. They used Unity as the main software for facial expression generation and ParallelDots API [16] for sentiment analysis. They created two different versions of the facial expressions: the generated one based on predefined emotions, and the live animated version. These two versions were compared and evaluated using a survey, that assessed the avatar's human-likeness, pleasantness, and life-likeness. What distinguishes our method from this effort is that we build the model based on LSTM instead of using ParallelDots API, and used the embedded automatic operations provided in iClone7, and OpenPose for Blender to represent emotions on 3D Avatar.

## III. Semantic Analysis

### A. Dataset and GloVe word embedding

The dataset consists of 188 labeled sentences and labels are presented in the form of integers from [0,4] corresponding to 5 different emotions: happy, sad, loving, excited and bored. The dataset is split into the train set and test set, containing 132 and 56 instances respectively. Each sentence contains up to 10 words. Moreover, in the code in order to train the Long Short Term Memory (LSTM) based Deep Learning model, Global Vector model (GloVe) in Word embedding was used.

GloVe is an unsupervised learning algorithm created to acquire vector representations for words. Training of the model was executed on combined global word to word co-occurrence statistics, and the resulting representations shows linear structures of the word in a vector space [17]. GloVe includes 1 million words and their vectors with dimension (50,).

Training is performed on aggregated global word-word co-occurrence statistics from a corpus, and the resulting representations showcase interesting linear substructures of the word vector space.

### B. Data Preprocessing

- In order to extract information from csv files, the "csv_reader" function was created, which takes as an input path to csv files in string format and outputs 2 variables, x and y, in numpy array.
- Furthermore, to process the labels, one-hot encoders were used, which converts integers from [0,4] corresponding to 5 different emotions to binary numbers. For instance, 1 will be presented like [0,1,0,0,0] and 5 will be [0,0,0,0,1].
- In this stage, GloVe vectors were read. Created function takes as an input path to the glove.txt file and output 3 different variables: words_to_index, which is index of the word, index_to_words, word itself and word_to_vec, which is vector of the word with shape (50,). Furthermore, these variables are passed to create Embedding layer, which is a lookup table that stores fixed dictionary.
- To process sentence, so they can be used in model training, array of strings in the sentence were converted to an array of indices, which corresponds to the word index in the GloVe embedding.

### C. Model Architecture

Long Short Term Memory (LSTM) based Deep Learning model was chosen to improve performance of semantic analysis. LSTM network is type of recurrent neural networks (RNN) and very efficient in terms of memory and parameter propagation within multiple layers. Below in the Figure 1. is shown the architecture of LSTM. It can be noticed that LSTM works sequentially as RNN forwarding previous hidden state to the next [18]. However, the most important feature of LSTM is that LSTM networks remove the unnecessary data using activation functions, sigmoid and tanh. Sigmoid compresses data between 0 and 1, whereas tanh ranges between -1 and 1.

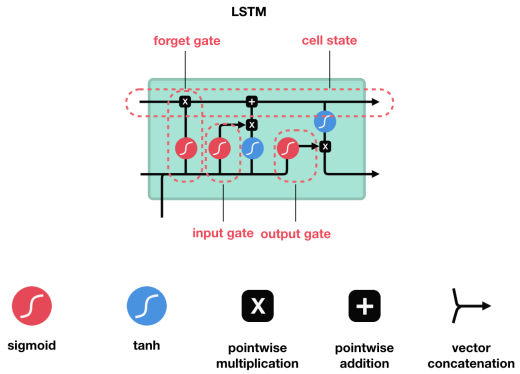



Fig. 1. LSTM architecture [18]

In the Figure 2, architecture of LSTM based Deep Learning model that was used in the project is shown. Each string of the sentence is passed to the GloVe embedding, which outputs vector of the word with dimension 50. Furthermore, each vector is passed to LSTM. Parameters for the LSTM were set as follows: hidden layer dimensionality = 128, number of recurrent layers = 2, dropout = 0.5, batch first = True and other parameters are default. To prevent model overfitting, dropout of Pytorch was used, which randonly removes some hidden layers. The to map the results of LSTM to the proper output dimensions, linear layer was used. At the end, softmax of Pytorch rescales the final result.

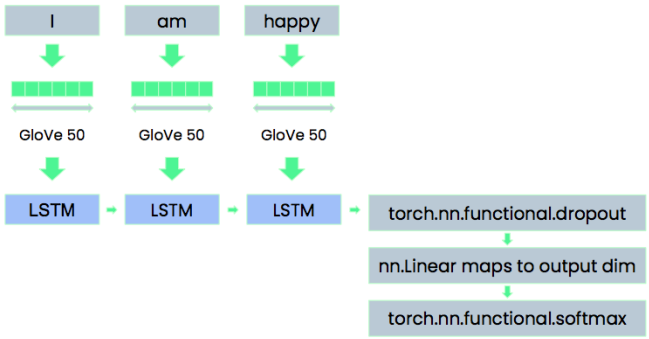



Fig. 2. LSTM based Deep Learning model architecture

### D. Results

Test loss after training 50 epochs equal to 1.095, whereas test accuracy is 0.828. Below are presented the plots of loss and accuracy. Moreover, we passed our sentences and predicted emotions. The results can be seen in the Figure 5.


Fig. 3. Loss and Accuracy

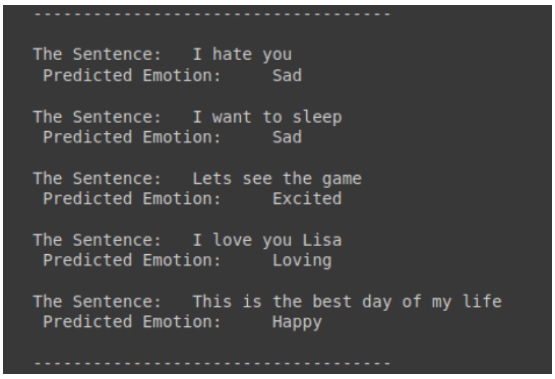



Fig. 4. Prediction

## IV. Facial Expression Generation

The second part of our project was to implement and compare facial expression generation using two different 3-D modelling software. First software is IClone 7 and the second one is called Blender.

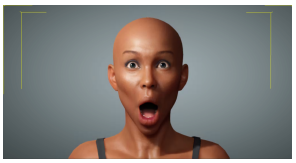
### A. IClone 7 Facial Expression Generation

IClone 7 is a powerful software that is used to create 3D avatars, models and it is mostly used for videogame engine rendering. This approach was straightforward, to use embedded avatars and apply different emotions. IClone already has add-on called ”Facial Puppeteering” and ”Emotions”. By using the first tool you can set different positions for each bone of the face. While for the second option everything is ready and you only have to click on it. In Figure 6 different emotions are shown.

It can be seen from the figure 6 that IClone 7 has very advanced rendering technology that allows for creation of human-like avatars that can exhibit immense number of emotions. However, we wanted to come up with our own facial expression transfer technology. For this purposes we used OpenPose and Blender.

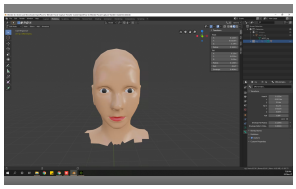
(a) Neutral

(b) Sad

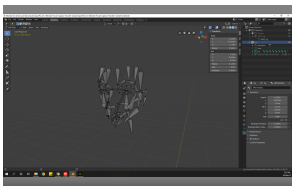


(c) Shocked

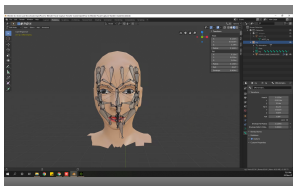
Fig. 5. IClone 7 emotions on avatar



(a) Outer skin



(b) Bones

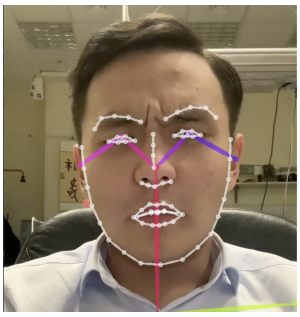


(c) Final Model

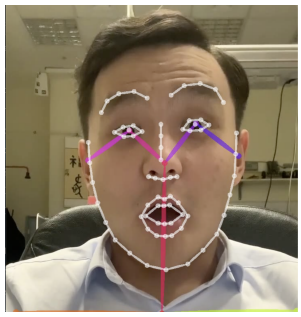
Fig. 7. Blender 3D avatar

### *B. OpenPose and Blender Facial Expression Transfer*

Second approach extracts the keypoints from the pre-recorded video and through the Python script tries to map these keypoints to the face of the 3D avatar. OpenPose is used for extracting the keypoints from the video. Figure 7 shows 2 frames from the video, where Nurdaulet tries to show emotions. These keypoints are extracted as JSON files frame by frame and saved in the same folder as our future blender file.



(a) Angry



(b) Shocked

Fig. 6. OpenPose keypoints

Next step is to prepare our Blender 3D avatar. To make avatars move Blender has specific add-on called ”Rigs”, they are like bones in our face. So, we had to align these bones with the actual skin of our avatar. If these are aligned perfectly there may be errors while running the python script.

Each bone on a face of 3D avatar has its name and these names will be used to map extracted keypoints from the video to the model. To do this, we used various mathematical operations and manipulations on points. The result was not as satisfactory as we anticipated. However, our code and model work, its bones move accordingly with the bones of a person in the video. Incompatibilities are may be due to the fact that OpenPose has limited number of keypoints and our 3D model cannot reach full synchronization with the video.

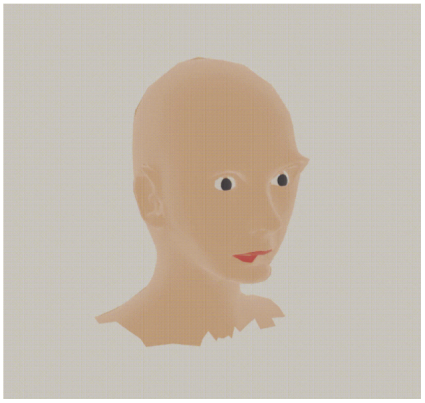


Fig. 8. Final Result

## V. Conclusion and Future Work

We developed the model based on LSTM, PyTorch and GloVe that takes word sequences as an input and feed into LSTM to predict the emotion. We were able to reach relatively accurate results on test set and correct output, when testing the model on random unseen instances. Then, we used OpenPose for representing obtained emotions in Blender. However, the facial expressions in iClone 7 generated by embedded automatic operations significantly outperforms the expressions generated using OpenPose, alleviating the need for comparison. Therefore, for future work we suggest focusing on the improvement of semantic analysis rather than

3D avatar creation and facial expression generation. Because such softwares as IClone 7 have reached excellent level of human-likeness in avatars and can be extensively used for expression generation. In addition, combining the process of facial expression generation with semantic analysis into a single pipeline would be a significant contribution for 3D animation.

## REFERENCES


[1] K. Aitpayev and J. Gaber, "Creation of 3d human avatar using kinect," *Asian Transactions on Fundamentals of Electronics, Communication & Multimedia*, vol. 1, no. 5, pp. 12–24, 2012.

[2] C. Chen, L. B. Hensel, Y. Duan, R. A. Ince, O. G. Garrod, J. Beskow, R. E. Jack, and P. G. Schyns, "Equipping social robots with culturally-sensitive facial expressions of emotion using data-driven methods," in *2019 14th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2019)*. IEEE, 2019, pp. 1–8.

[3] M. Pantic, "Machine analysis of facial behaviour: Naturalistic and dynamic behaviour," *Philosophical Transactions of the Royal Society B: Biological Sciences*, vol. 364, no. 1535, pp. 3505–3513, 2009.

[4] D. Mukashev, M. Kairgaliyev, U. Alibekov, N. Oralbayeva, and A. Sandygulova, "Facial expression generation of 3d avatar based on semantic analysis," in *2021 30th IEEE International Conference on Robot Human Interactive Communication (RO-MAN)*, 2021, pp. 89–94.

[5] P. Ekman, "Are there basic emotions?" 1992.

[6] H. X. Pham, Y. Wang, and V. Pavlovic, "End-to-end learning for 3d facial animation from raw waveforms of speech," *arXiv preprint arXiv:1710.00920*, 2017.

[7] T. Karras, T. Aila, S. Laine, A. Herva, and J. Lehtinen, "Audio-driven facial animation by joint end-to-end learning of pose and emotion," *ACM Transactions on Graphics (TOG)*, vol. 36, no. 4, pp. 1–12, 2017.

[8] U. Gupta, A. Chatterjee, R. Srikanth, and P. Agrawal, "A sentiment-and-semantics-based approach for emotion detection in textual conversations," *arXiv preprint arXiv:1707.06996*, 2017.

[9] A. S. Miner, A. Milstein, S. Schueller, R. Hegde, C. Mangurian, and E. Linos, "Smartphone-based conversational agents and responses to questions about mental health, interpersonal violence, and physical health," *JAMA internal medicine*, vol. 176, no. 5, pp. 619–625, 2016.

[10] P. Edwards, C. Landreth, E. Fiume, and K. Singh, "Jali: an animator-centric viseme model for expressive lip synchronization," *ACM Transactions on graphics (TOG)*, vol. 35, no. 4, pp. 1–11, 2016.

[11] B. Fan, L. Xie, S. Yang, L. Wang, and F. K. Soong, "A deep bidirectional lstm approach for video-realistic talking head," *Multimedia Tools and Applications*, vol. 75, no. 9, pp. 5287–5309, 2016.

[12] L. Wang and F. K. Soong, "Hmm trajectory-guided sample selection for photo-realistic talking head," *Multimedia Tools and Applications*, vol. 74, no. 22, pp. 9849–9869, 2015.

[13] K. Liu and J. Ostermann, "Realistic facial expression synthesis for an image-based talking head," in *2011 IEEE International Conference on Multimedia and Expo*. IEEE, 2011, pp. 1–6.

[14] M. Liu, Y. Duan, R. A. Ince, C. Chen, O. G. Garrod, P. G. Schyns, and R. E. Jack, "Building a generative space of facial expressions of emotions using psychological data-driven methods," in *Proceedings of the 20th ACM International Conference on Intelligent Virtual Agents*, 2020, pp. 1–3.

[15] P. Ekman, "Facial action coding system," 1977.

[16] "Emotion analysis." [Online]. Available: https://komprehend.io/emotion-analysis

[17] J. Pennington. [Online]. Available: https://nlp.stanford.edu/projects/glove/

[18] M. Phi, "Illustrated guide to lstm's and gru's: A step by step explanation," Jun 2020. [Online]. Available: https://towardsdatascience.com/illustrated-guide-to-lstms-and-grus-a-step-by-step-explanation-44e9eb85bf21