İstanbul
Bilgi Üniversitesi

CMPE 492 : SENIOR DESIGN PROJECT

CHATBOT WITH FACE EMOTION DETECTION USING DEEP LEARNING

Ekin Silahlıoğlu (116200078)

Kadir Akgül (115200081)

Selin Yeşilselve (115200041)

Computer Engineering

Course Teacher: Pınar Hacıbeyoğlu

Advisor : Tuğba Yıldız

Faculty of Engineering and Natural Sciences

Istanbul Bilgi University

2020

Contents

Figures of Content

Tables of Content

Abstract

In this paper, we proposed a retrieval-based close-domain chatbot model by using BERT vectorization on Long Short Term Memory (LSTM) model. Our process starts with emotion status received from users' snapshot of their faces in real time. Then we use this emotion status as a starting point for our proposed chatbot model. We created two datasets to compare on different Deep Learning algorithms and vectorization models. We came to the conclusion that our proposed model achieved the most efficient accuracy as 90.45%. The architecture integrated to web using Flask.

1. Introduction

Artificial intelligence applications (AI), that are encountered in every field of our day make our lives more efficient and will increase in line with the needs of people. For all that AI has included numerous branches such as Machine Learning, Deep Learning, Image Processing, Natural Language Processing (NLP), Voice Recognition, etc.

The human face is the focus of visual attention from the very beginning. We can draw many conclusions about himself from the facial expression of man. Emotion detection is one of them. In order to understand and communicate efficiently, we should have a speech depending on the emotional state of the other person. Considering this, we have designed a chatbot that communicates by considering the emotional state of the person that we trying to communicate. Ian Goodfellow states that human accuracy on FER-2013 is 65% ± 5% [1]. If we consider the images in the FER-2013 dataset shown in Figure 1, it will be a really hard process to classify the images manually according to 7 (“angry”, “disgust”, “fear”, “happy”, “sad”, “surprise”, “neutral”) emotional states.

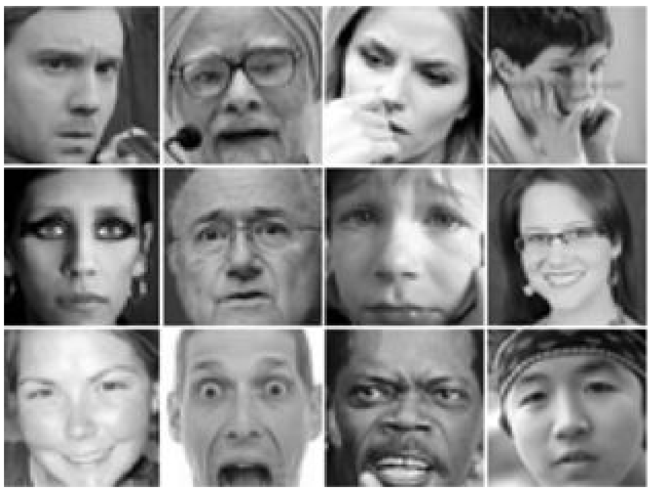


Figure 1 : Samples of the FER-2013 emotion dataset

Chatbots became drastically spread with the evolution of Artificial Intelligence and Natural Language Processing. Over the course, its history the interaction between chatbot and humans gets a more complex way. It was started with ELIZA and most of the participants who talked with ELIZA believed that they were talking with an actual person. ELIZA expects the user input for keywords and it transformed the sentence according to the rules related with that keyword [2] . After that, there are many chatbots available to us such as Rose, Siri, Alexa, Mitsuku, etc.

Chatbots are designed according to their purpose. The first one is task-related chatbots. These generally have close domain areas such as helping customers for their needs, simple chatting, or areas that do not require broad approaches. Moreover, task-related chatbots divide into two groups: Rule-Based and Retrieval Based. Rule-Based chatbot programs are mainly based on rules written by programmers but in Retrieval Based, programmers can use Deep Learning

Algorithms to achieve their goals. The challenging part is non-task related open domain chatbots. These chatbots aim to talk about any subject with the human in a smooth, continuous way like a human being. Open-domain chatbots are hard to build. Because they need an immense amount of data to understand the grammar, context, polysemy, structure of any given sentence so that they can act answer like a human being. The current obstacle is the lack of data and related to this, mistakes in the grammar structure of sentences.

The process of our aim, which starts by taking a real-time image from the user, continues with emotion detection in the received image. Related to the detection, an emotion tag occurs. It continues with the conversation set up with chatbot, which was created with the model mapped to the received emotion tag. The chatbot has created as a retrieval-based close-domain chatbot. Our chatbot has limits, it can talk to a person on some subjects with a limited amount of answers.

2. Related Works

In the area of chatbots, there are numerous examples of generative-based, retrieval-based, and ruled-based chatbots. In this section, some of the works were explained to give a better perspective and understanding of the area.

Conversational Assistant based on Sentiment Analysis article is one of the generative-based chatbot example. In this paper, they aim to create relevant responses to people by understanding whether that person is happy, neutral, or sad. They also took recent tweets from users to get a better insight into their emotional status and letting users see their changes in emotion over time. They used chat samples to train it and the Seq2Seq algorithm for the model. The model works as chatbot takes input from the user and uses the Seq2Seq model to comprehend the sentence and takes emotion status then it can generate a related answer. Also, the conversation is started by the chatbot. Moreover, they added audio files to help the user for changing their mood better. At last, they created an application to launch their model [3]. In A Multi-Model and AI-Based CollegeBot article, they chose an intent-based approach. Their purpose is to create a chatbot that auto-responds to student queries about college basic information. Their data consists of 48 intents with more entities such as names, schedules, addresses, etc. They designed their chatbot in DialogFlow. The chatbot provides voice and text as an input and also yields text and voice as an output. Also, their chatbot has three roles: Principal, Staff, and Student. Each one of them has its purpose for college students. For their future work, they aim to make it multilingual [4].

In AliMeChat [5], authors state that open-domain chatbots model which are consist of Information Retrieval (IR) and generation model techniques, implementations on Question-Answer (QA) conversations occur some problems. IR models are not able to handle with long-tail questions and answers generated by generations models may be incoherent and pointless. They proposed a hybrid model to eliminate these problems. The process starts, for a question, they use an IR model to withdraw a group of QA pairs and use them as candidate answers to reranked by using Seq2Seq model. Assuming that, there exists a higher score between top candidates than the current threshold the answer will be produced by a generation-based model, diversely it will be accepted as an answer. To evaluate the efficiency of their hybrid model they make a comparison with already created, available to the public chatbot [6]. Two business analysts who were tasked with making the comparison were asked to choose an answer for each test question. The proposed model outperformed the other chatbot, with better results in 37.64% of the 878 questions and 18.84% worse results.

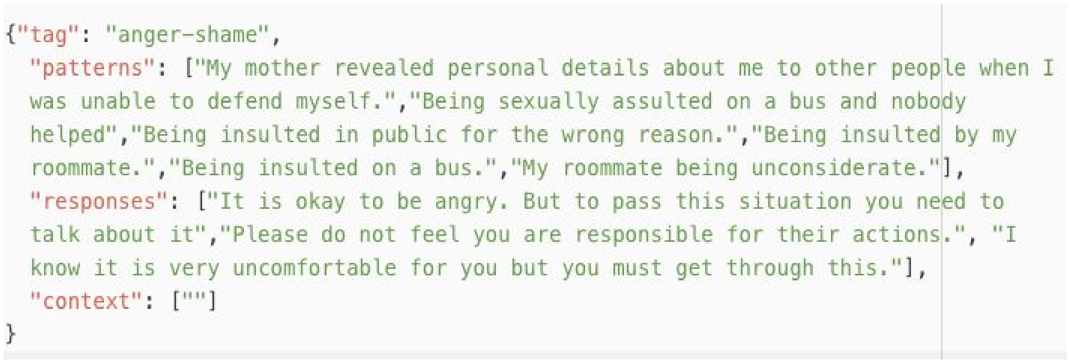
In another article as A Neural Conversational Model [7], the approach distinguishes the model proposed as predicting the next sequence depending on the sequence or sequences given in a conversation is that it can be trained end-to-end by Recurrent Neural Networks. For the measurement part, two separate data sets, closed-domain dataset and open-domain dataset were used. The experiments that have done on a closed-domain which comprise of conversations about customers face computer related issues, mentioned as IT helpdesk troubleshooting chat service. The proposed model achieved perplexity of 8, although an n-gram model achieved 18. In the experiments of open-domain movie transcript dataset that trained by two-layered LSTM, has 17 the perplexity on the validation set. However, the smoothed 5-gram model achieves 28. Addition to these, they evaluated by using human judges consist of four people which make an even comparison between the model proposed and CleverBot (CB) on 200 chosen questions. Agreement ratio is taken as reference in this human evaluation part. And the proposed model has selected versus CB in 97 questions.

3. Methodology

3.1. Dataset

We designed our dataset for our retrieval-based chatbot. It consists of tags, patterns, and responses. Tags represent the context as this chatbot specifies its content according to the conversation. Furthermore, patterns represent the possible sentences that the user may say to our chatbot. We gave multiple sentences for each tag so our chatbot can recognize that a particular sentence refers to that tag. We applied two datasets to our proposed model. The first one (Dataset-1) consists of 8 tags which are greeting, thanks, goodbye, no-answer, recommend, sad-mood, happy-mood and angry-mood, 83 patterns, and 58 responses. Because of its lack of tag range, the conversation can not extend its context and give appropriate responses. Thus the current dataset (Dataset-2) consists of 15 different tags which are greeting, thanks, goodbye, no-answer, recommend, sad-mood, happy-mood, angry-mood, sad-shame, guilt, angry-disgust, sad-lost, celebrate, angry-shame and love, 220 patterns and 95 responses.

Users do not strict with the possible patterns, they can give any sentence they desire. Finally, the responses part represents the potential answers our chatbot can give. As we created a retrieval-based close-domain chatbot these responses are fixed and it can not give any generative answers.



```
{"tag": "anger-shame",
  "patterns": ["My mother revealed personal details about me to other people when I
  was unable to defend myself.","Being sexually assulted on a bus and nobody
  helped","Being insulted in public for the wrong reason.","Being insulted by my
  roommate.","Being insulted on a bus.","My roommate being unconsiderate."],
  "responses": ["It is okay to be angry. But to pass this situation you need to
  talk about it","Please do not feel you are responsible for their actions.", "I
  know it is very uncomfortable for you but you must get through this."],
  "context": [""]
}
```

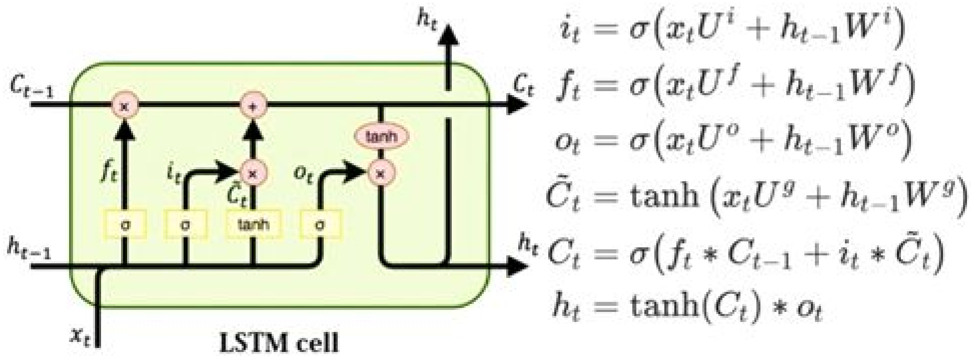
Figure 2 : Example structure of Dataset-2

3.2. Models

In this section, the models that we used for training chatbot; Long Short Term Memory and Gated Recurrent Unit, the vectorization models for our patterns; BERT, ULMFiT and Word2Vec and Convolution Neural Network for emotion detection will be discussed.

3.2.1. Long Short Term Memory (LSTM)

Long Short-Term memory networks are a sub division of recurrent neural networks. LSTMs are designed to avoid the long term dependency problem of RNNs. Memory cells and gates allows the cells to remember the previous information for long periods of time. These memory cells can store information, write and read it.



$$i_t = \sigma(x_t U^i + h_{t-1} W^i)$$
$$f_t = \sigma(x_t U^f + h_{t-1} W^f)$$
$$o_t = \sigma(x_t U^o + h_{t-1} W^o)$$
$$\tilde{C}_t = \tanh(x_t U^g + h_{t-1} W^g)$$
$$C_t = \sigma(f_t * C_{t-1} + i_t * \tilde{C}_t)$$
$$h_t = \tanh(C_t) * o_t$$

Figure 3 : LSTM cell and its mathematical formula

The information is controlled with the use of input gates, forget gates and output gates [8] .

LSTM works in an efficient way even if there are widely separated and relevant inputs [9]. LSTM is better at classification, processing and because of its performance it is an ideal model for creating projects that needs continuation and sequentiality.

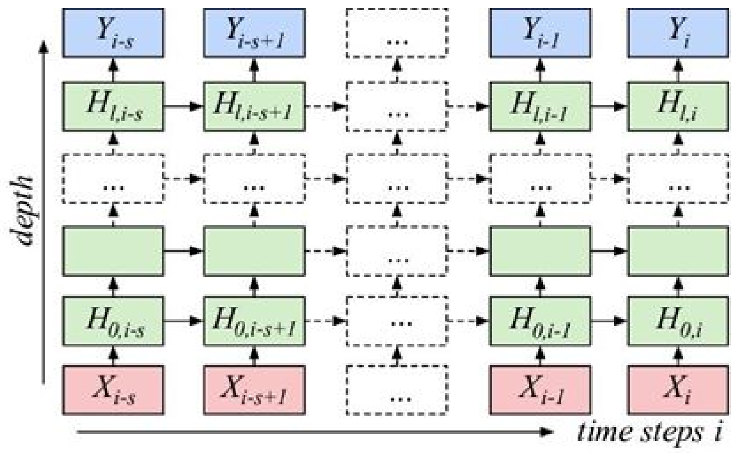


Figure 4 : LSTM work principle in Hidden Layers

3.2.2. Gated Recurrent Units (GRU)

It was first introduced by Cho. et. al in 2014 [10]. Its goal is to elucidate the vanishing gradient problem. GRU is similar to LSTM but they have few differences. In LSTM, it has three gates; input, output and forget gate. However, GRU has two gates reset and update. The update gate helps the model to choose how much of the past information needs to be pass for next step. The reset gate decides how much of the passed information needs to be forgotten. And these two gates are dependent with each other.

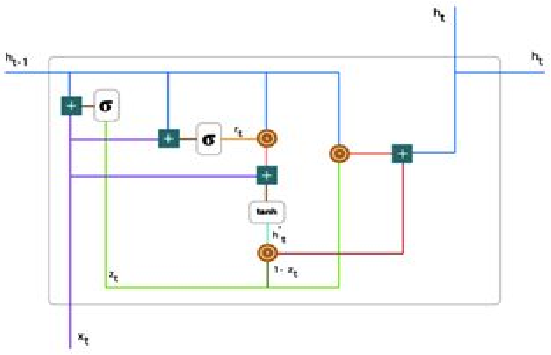


Figure 5 : GRU Architecture

GRUs are able to store and filter the information using their update and reset gates. That eliminates the vanishing gradient problem because it keeps the relevant information and passes it down to the next time steps of the network [11].

The mathematical formula of the encoder GRU hidden state ht as follows [12]:

$$zt = \sigma(Wxzxt + Uhzht-1) \qquad (1)$$

$$rt = \sigma(Wxrxt + Uhrht-1) \qquad (2)$$

$$\tilde{h}t = \tanh(Wxhxt + Urh(rt \otimes ht-1)) \qquad (3)$$

$$ht = (1 - zt) \otimes ht-1 + zt \otimes \tilde{h}t \qquad (4)$$

Gated Recurrent Units use less training parameters therefore they use less memory. Also they execute faster and train faster than LSTM.

$$zt=\sigma(Wxzxt+Uhzht-1) \qquad (5)$$

3.2.3. Convolutional Neural Networks (CNN)

In cases that require us to deal with the image, the model that will enable us to easily overcome these difficulties is Convolutional Neural Network (CNN). CNN uses a Neural Network to solve classification problems. The main goal is to combine features into more features and predict classes better. The process of how CNN works is shown in the figure. More detailed explanation will be given in the forward sections.

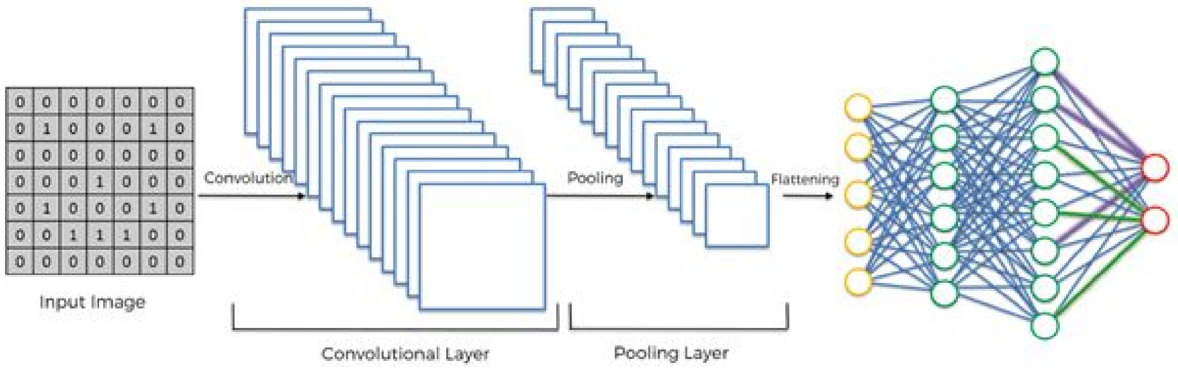


Figure 6 : CNN Architecture

3.2.4. Word Embeddings

Word embedding methods represent words as vectors and it can comprehend the context of a word in a sentence by looking at the semantic, syntactic relation with other words. What word embeddings do is turn each words into a numerical value thus we are increasing the machine's capability of understanding the given sentence.

3.2.4.1 Word2Vec

Word2Vec is created by Tomas Mikolow in 2013 at Google. Word2Vec is a method to construct such an embedding. It can be obtained using two methods (both involving Neural Networks): Skip Gram and Common Bag Of Words (CBOW). It vectorizes each word according to the words near them. It is looking at the meaning of the sentence and has a small vector space. In CBOW model, it takes the context of each word as the input and tries to predict the word corresponding to the context [13] .

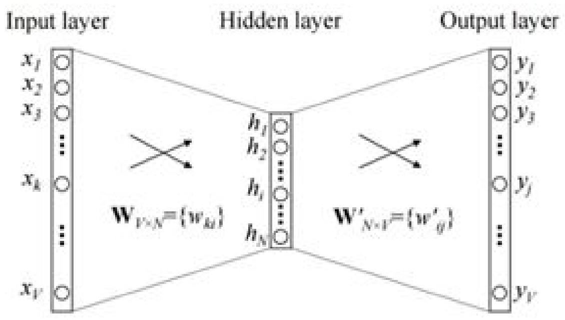


Figure 7 : CBOW Architecture

The input is a one-hot encoded vector, which is if that given word is present is the sentence we give 1 and if not we give 0. The weights between the input layer and the output layer can be represented by a V × N matrix W. Each row of W is the N-dimension vector representation vw of the T associated word of the input layer [14]. And the mathematical formula of CBOW as follows:

$$\mathbf{h} = \mathbf{W}^T\mathbf{x} = \mathbf{W}^T_{(k,\cdot)} := \mathbf{v}^T_{w_I}, \tag{6}$$

$$u_j = {\mathbf{v}'_{w_j}}^T\mathbf{h}, \tag{7}$$

$$p(w_j|w_I) = y_j = \frac{\exp(u_j)}{\sum_{j'=1}^{V}\exp(u_{j'})}, \tag{8}$$

$$p(w_j|w_I) = \frac{\exp\left({\mathbf{v}'_{w_j}}^T\mathbf{v}_{w_I}\right)}{\sum_{j'=1}^{V}\exp\left({\mathbf{v}'_{w_{j'}}}^T\mathbf{v}_{w_I}\right)} \tag{9}$$

And in skip-gram model for each context position, it gets C probability distributions of V probabilities, one for each word.

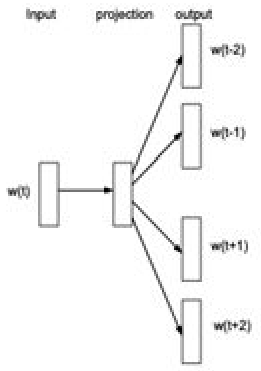


Figure 8 : Skip-gram Architecture

Moreover, given a sequence of training words, the objective of the Skip-gram model is to maximize the average log probability [15].

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{-c\leq j\leq c, j\neq 0}\log p(w_{t+j}|w_t) \tag{10}$$

where c is the size of the training context. Larger c results in more training examples and thus can lead to a higher accuracy, with a more training time. The basic Skip-gram defines p(wt+j|wt) using the softmax function:

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{-c\leq j\leq c, j\neq 0}\log p(w_{t+j}|w_t) \tag{11}$$

According to Mikolov, Skip Gram works well with small amount of data and is found to represent rare words well.

3.2.4.2 BERT

Bidirectional Encoder Representations from Transformers. BERT uses a masked language model inspired by the Cloze task [16].

The model randomly masks some of the tokens and tries to predict that mask word according to the context of the input. The model uses both left-to-right and right-to-left representation to train the model which is called a deep bidirectional transformer. Also, it uses the next sentence prediction task to pre-trains the text-pair representations. BERT shows state-of-the-art performance on sentence-level and token level tasks [17].

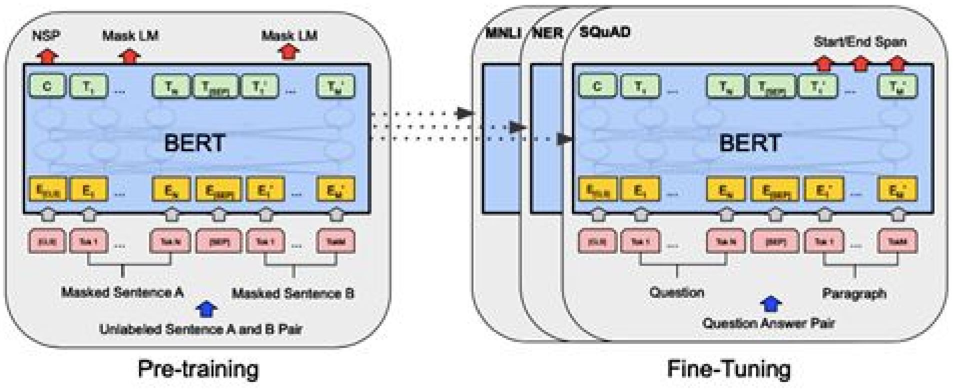


Figure 9 : BERT input representation

3.2.4.3 Universal Language Model Fine-Tuning (ULMFiT)

ULMFit is a pre-trained language model with a large general domain corpus and fine-tunes it on the target task. It can easily work on a task with a diversity of document, size, number, and label type. It uses a single architecture and training process. Also, it does not require custom feature engineering or preprocessing steps. It does not use ant transformers thus uses unidirectional LSTM [18].

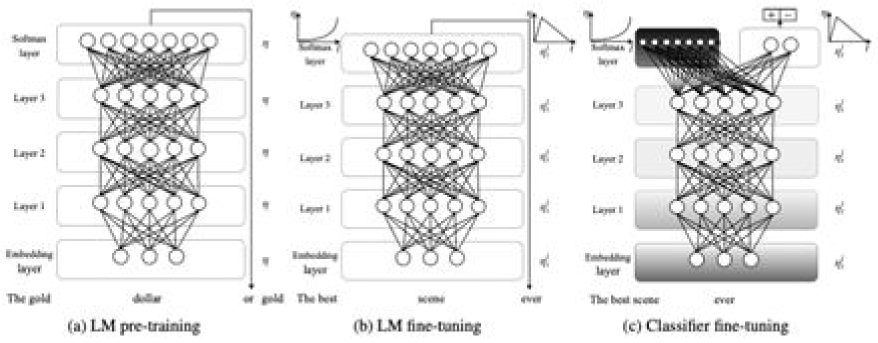


Figure 10 : ULMFiT Architecture

ULMFiT has 3 steps. The first one is the General Domain LM. In this step, they train the language model on the Wikitext-103 corpus. The second one is Target Task LM Fine-Tuning and here it allows us to train a more powerful LM for small datasets. Two fine-tuning are proposed: Discriminative Fine-Tuning and Slanted Triangular Learning Rates which uses different learning rates at each layer.

We used Word2Vec, BERT, and ULMFiT to evaluate our model. We are going to see compared results in the following sections.

3.3. Hyperparameters

To create our chatbot model we decided to use LSTM within Sequential Network. The first layer is an Embedding layer with the parameters of vocabulary size (include BERT vocabulary), dimensions (in the model 64 is chosen), and maximum length (assume that each sentence length should be 20). LSTM has 100 neurons. There are two Dropout layers with a 0.2 dropout rate. And a Dense Layer for the output with the parameters of 15 (represents tag number) and a sigmoid activation function.

Moreover, for compiling the model, Adam optimizer, 10% validation split, sparse categorical cross-entropy loss, and accuracy metrics are used. Ultimately to fit our data, we decided to use 400 epoch with a batch size of 5.

4. Experimental Setup

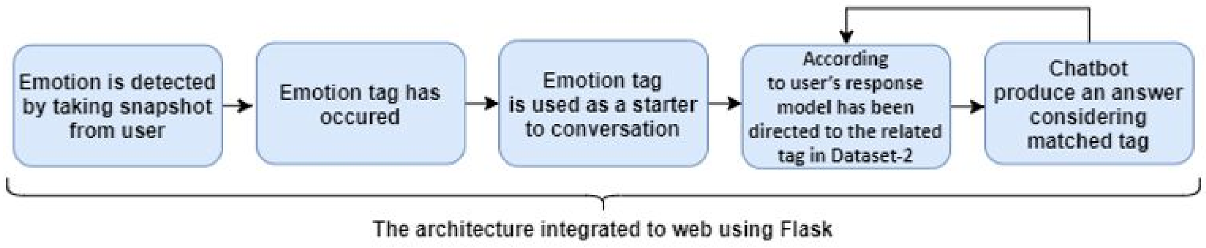

Emotion is detected by taking snapshot from user
Emotion tag has occured
Emotion tag is used as a starter to conversation
According to user's response model has been directed to the related tag in Dataset-2
Chatbot produce an answer considering matched tag
The architecture integrated to web using Flask

Figure 11 : The architecture of proposed chatbot model

Related to the figure our process has started with taking a snapshot from the user and produce an emotion tag which is proposed by a model address to Arriaga et.al.'s works [19]. The model that used for training creating according to Xception [20] architecture. This model is consist of residual modules [21] and depth-wise separable convolutions [22]. Residual modules permit the progression of information from the beginning layers to last layers.

The desired features H(x) are adapted in order to solve the problem F(X) such that:

$$H(x) = F(x) + x \qquad (12)$$

The convolution operation is distinguished as depth-wise convolutions and point-wise convolutions in Depth-wise separable convolution. Depth-wise separable convolutions reduce the conjecture corresponding with the standard convolutions by a factor of :

$$1/N+1/D^2 \qquad (13)$$

The distinction in normal Convolutional Layer and depth-wise convolutional layer is mentioned clearly in Figure.

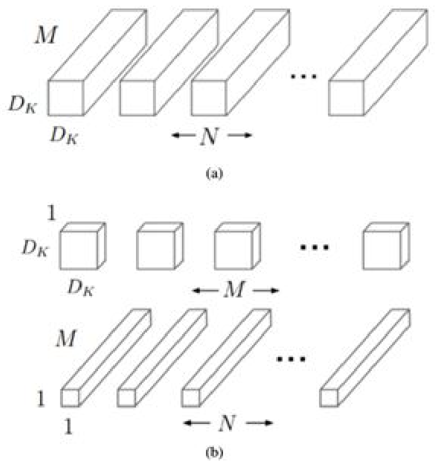


Figure 12 : Difference between (a) standard convolutions and (b) depth-wise separable convolutions [22].

The last outlining of the creation of Neural Network is consist of 4 residual dept-wise separable convolutions. Each layer followed by ReLU activation function and batch normalization operation. After these executions completed in pooling layer, average-pooling and softmax activation function have applied. In the figure, the whole process referred to as mini-Xception is shown.

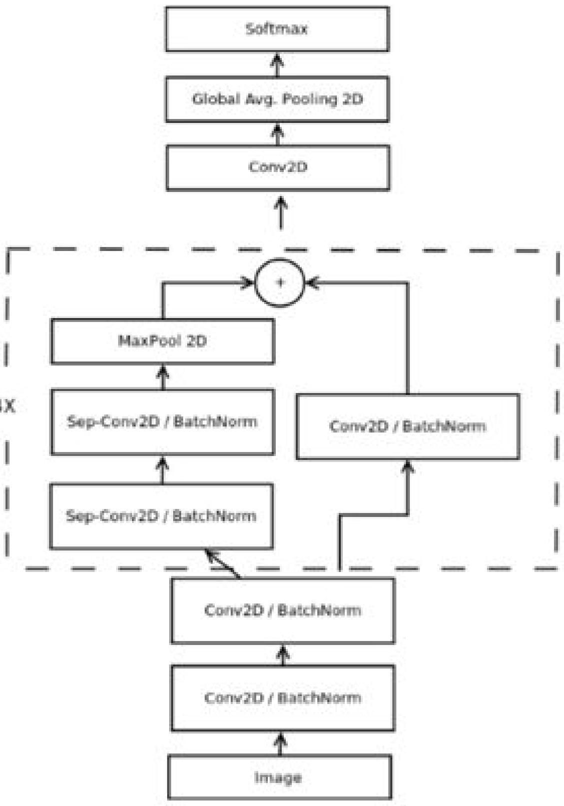


Figure 13 : Proposed Model for Real Time Classification

Result of these implementations, the output will be an emotion tag. Furthermore, we use this output as an input for referencing the tag to the chatbot system for starting the conversation.

To train the chatbot model we proposed, we made preprocessing steps for the dataset which is mentioned in Section 3.1. Firstly, we created 3 different variables which are consist of tags, patterns, and responses to process them easily. To decrease the complexity of patterns that are taken from the dataset, the procedures of cleaning the data are implemented such as removing punctuations and lower the words. LSTM model requires vector formation as input thus patterns are put into the BERT vectorization algorithm to transform the data into the required format. At the same time, to fit our target values into our proposed model, label encoding is implemented on target values which then transform string values into integer values. Moreover, the model is fit by taking patterns and tags as input.

The flow of conversation is started by chatbot with a sentence according to the emotion tag. When a user types a sentence, the proposed model will predict its related tag after the user's input went through preprocessing steps which is mentioned earlier. Finally, the chatbot will give an appropriate response from the responses list that is matched with the related tag.

The result of both architecture implemented on web using Flask. We chose Flask because of its easy implementation and Pythonic behaviour [23]. There are screenshots related to how we implement is on Flask will be shown in Appendix section.

5. Experimental Result & Discussion

The confusion matrix of mini-Xception model which is used in emotion detection, prepared by the authors is displayed in the figure. As we all can see there are some errors, such as the wrong prediction of 'fear' as 'sad' etc. Result of these, the model has achieved 66% accuracy on FER-2013 dataset which contains 35,887 grayscale images. And these images classified as "angry", "disgust", "fear", "happy", "sad", "surprise", "neutral".

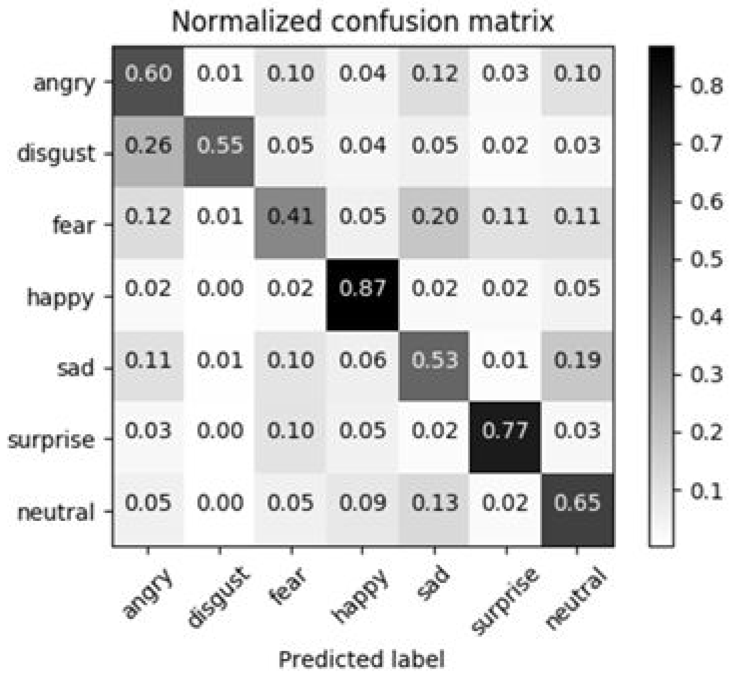


Figure 14 : Normalized Confusion Matrix

In Table 1, it is shown the comparison of two models on two different datasets that we used in chatbot part.

	LSTM	GRU
DATASET-1	81.11%	90.76%
DATASET-2	90.45%	78.93%

Table 1 : Model Accuracy Comparison on Old and Current Dataset

In LSTM model, Dataset-2 exceeds the Dataset-1 by getting 90.45% accuracy. Reason of this increase is due to fact that there are more data for our model to learn and give stable results. In contrary, because of the GRU model works more efficiently on small datasets, Dataset-1 gave more accuracy than the current dataset. GRU uses less training parameters and because of that it does not learn all aspects of the data. In addition to this, LSTM model works better on datasets using longer sequences. Considering all of these facts, we decided to train our dataset on LSTM model.

	BERT	Word2Vec
DATASET-2	90.45%	77.50%

Table 2 : Comparisons of Vectorization Models on Current Dataset

Comparisons of vectorization models on Dataset-2 are shown in Table 2. First of all, all these accuracies are reached using the LSTM model except ULMFiT. In Word2Vec has 77.50% accuracy. However, this accuracy is not desirable for us. Because Word2Vec does not comprehend the context of longer sentences, can not handle unknown words, it just uses nearby words to understand it. For instance, it can not differentiate the “I got a job offer today.” and “I got mad.” as two different contexts. And it uses more memory due to creating embedding matrices that include each unique word in the dataset. Furthermore, ULMFiT has the lowest accuracy as 12.72%. Because it uses the unidirectional LSTM model and does not use transformers. On the contrary, BERT uses transformers and bidirectional LSTM model which is much better in word representation rather than unidirectional LSTM. When BERT encounters with an unknown word, it splits into subwords to check them in its vocabulary. So it can understand more complex words even if that word does not exist in its vocabulary. Thus BERT gives the highest accuracy among the three.

6. Conclusion

In this paper, our purpose is to create a chatbot that interacts by acknowledging the emotional status of the person that we trying to communicate. We designed our datasets depending on our purpose which can be easily implemented on models that we compared regarding the related works that we examined. BERT vectorization on the LSTM model achieved the most efficient accuracy as 90.45%. The system is implemented on web using Flask.

The proposed chatbot model can be improved to boost emotional status of users by using recommendation systems. Besides, the proposed chatbot model can be transformed into generative-based by increasing the amount of data and using improved deep learning techniques.

References

[1] Challenges in Representation Learning: A report on three machine learning contests, Ian Goodfellow et al., 2013

[2] A Survey on Conversational Agents/Chatbots Classification and Design Techniques Shafquat Hussain, Omid Ameri Sianaki, and Nedal Ababneh, March 2019

[3] Conversational Assistant based on Sentiment Analysis, Suraj D. M., Varun A. Prasad, Shirsa Mitra, Rohan A. R., Dr. Vimuktha Evangeleen Salis, September 2019

[4] A Multi-Model And Ai-Based Collegebot Management System (Aicms) For Professional Engineering Colleges, K. Arun, A. Sri Nagesh, P. Ganga, July 2019

[5] AliMeChat: A Sequence to Sequence and Rerank based Chatbot Engine, Minghui Qiu, Feng-Lin Li, Siyu Wang, Xing Gao, Yan Chen, Weipeng Zhao, Haiqing Chen, Jun Huang, Wei Chu, July 2017

[6] http://www.tuling123.com/

[7] A Neural Conversational Model, Oriol Vinyals, Quoc V. Le, July 2015

[8] Sak, H., Senior, A., Beaufays, F.: Long short-term memory recurrent neural network architectures for large scale acoustic modeling. In: Fifteenth Annual Conference of the International Speech Communication Association, 2014

[9] S. Hochreiter and J. Schmidhuber. Long short-term memory. Neural Computation, 1997

[10] Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation , Kyunghyun Cho, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, Yoshua Bengio, 3 September 2014

[11] https://towardsdatascience.com/understanding-gru-networks-2ef37df6c9be

[12] A GRU-based Encoder-Decoder Approach with Attention for Online Handwritten Mathematical Expression Recognition, Jianshu Zhang, Jun Du, Lirong Dai, 4 December 2017

[13]https://towardsdatascience.com/introduction-to-word-embedding-and-word2vec-652d0c2060fa

[14] word2vec Parameter Learning Explained, Xin Rong, 5 June 2016

[15] Distributed Representations of Words and Phrases and their Compositionality, Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, Jeffrey Dean, 16 October 2013

[16] Taylor, 1953

[17] BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, Jacob Devlin Ming-Wei Chang Kenton Lee Kristina Toutanova Google AI Language, 24 May 2019

[18] Universal Language Model Fine-tuning for Text Classification, Jeremy Howard, Sebastian Ruder, 23 May 2018

[19] Real-time Convolutional Neural Networks for Emotion and Gender Classification, Octavio Arriaga, Paul G. Plöger, Matias Valdenegro, October 2017

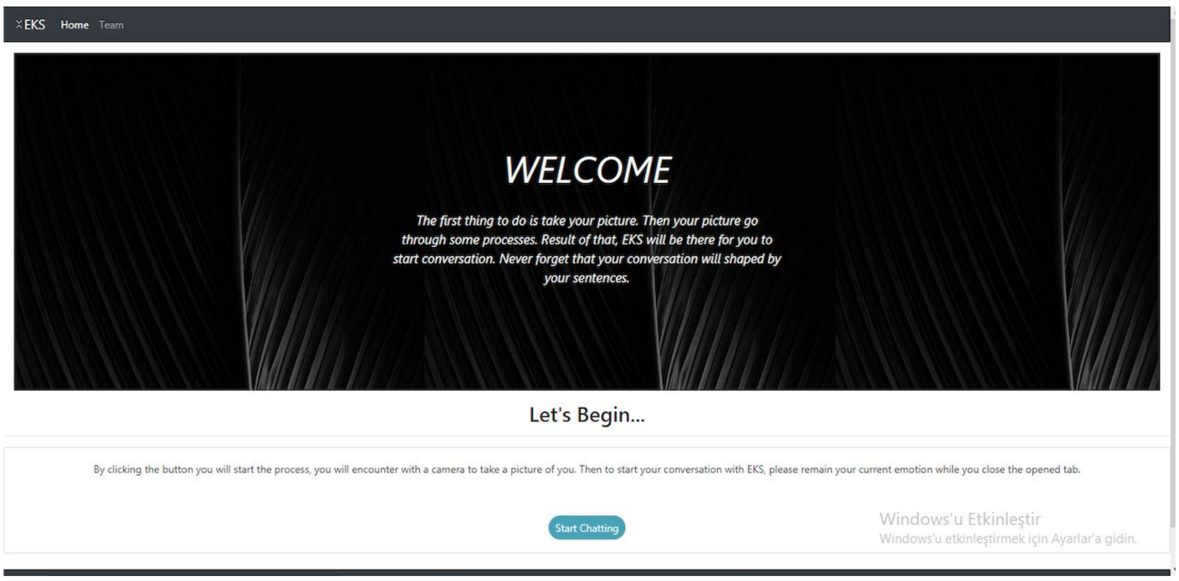
[20] Francois Chollet. Xception: Deep learning with depth-wise separable convolutions. CoRR, abs/1610.02357, 2016

[21] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 770–778, 2016

[22] Andrew G. Howard et al. Mobilenets: Efficient convolutional neural networks for mobile vision applications. CoRR, abs/1704.04861, 2017

[23] https://www.fullstackpython.com/flask.html

Appendix



WELCOME

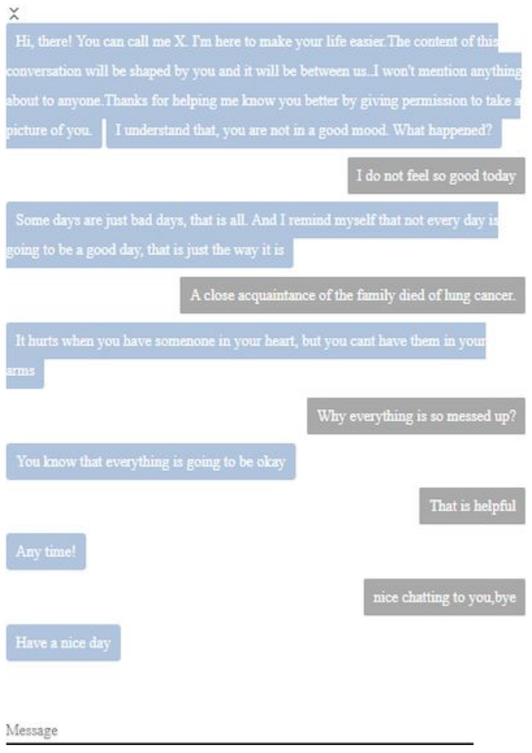
The first thing to do is take your picture. Then your picture go through some processes. Result of that, EKS will be there for you to start conversation. Never forget that your conversation will shaped by your sentences.

Let's Begin...

By clicking the button you will start the process, you will encounter with a camera to take a picture of you. Then to start your conversation with EKS, please remain your current emotion while you close the opened tab.

Windows'u Etkinleştir
Windows'u etkinleştirmek için Ayarlar'a gidin.

1[1]



Hi, there! You can call me X. I'm here to make your life easier.The content of this conversation will be shaped by you and it will be between us..I won't mention anything about to anyone.Thanks for helping me know you better by giving permission to take a picture of you.

I understand that, you are not in a good mood. What happened?

I do not feel so good today

Some days are just bad days, that is all. And I remind myself that not every day is going to be a good day, that is just the way it is

A close acquaintance of the family died of lung cancer.

It hurts when you have someone in your heart, but you cant have them in your arms

Why everything is so messed up?

You know that everything is going to be okay

That is helpful

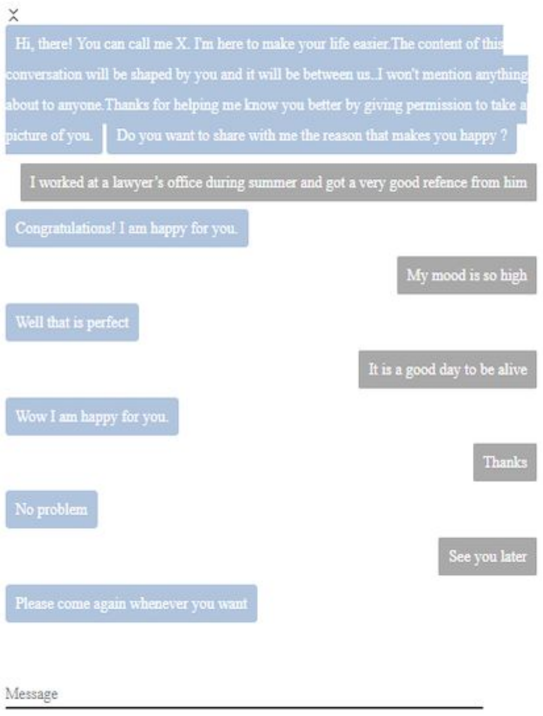
Any time!

nice chatting to you,bye

Have a nice day

Message

2[2]



Hi, there! You can call me X. I'm here to make your life easier.The content of this conversation will be shaped by you and it will be between us..I won't mention anything about to anyone.Thanks for helping me know you better by giving permission to take a picture of you.

Do you want to share with me the reason that makes you happy ?

I worked at a lawyer's office during summer and got a very good refence from him

Congratulations! I am happy for you.

My mood is so high

Well that is perfect

It is a good day to be alive

Wow I am happy for you.

Thanks

No problem

See you later

Please come again whenever you want

Message

3[3]

[1] Main Page for the web site
[2] Conversation for sad mood
[3] Conversation for happy mood