Automatic recognition of the user’s emotional state in one-to-one chat

Francesco Iemma, Lorenzo Massagli, Niko Salamini, Olgerti Xhanej  
 [email@email.com](mailto:email@email.com), [l.massagli@studenti.unipi.it](mailto:l.massagli@studenti.unipi.it), [email@email.com](mailto:email@email.com), [o.xhanej@studenti.unipi.it](mailto:o.xhanej@studenti.unipi.it%20)

ABSTRACT

Emotion Detection can be performed through data that can be obtained from a Chat Application, such as text and audio messages. In this document we present a possible approach to gather and classify heterogeneous data that comes from one-to-one chats. Firstly, there is an introduction to give a proper context to the problem and the usefulness of Emotion Recognition of Users, the application field of such technology goes from healthcare to marketing. In the second paragraph there is an in-depth review of the building blocks that made-up our system, in particular we underline how the Android app is developed and which are the design choices made, then we also explain the overall system that is composed not only by the Android application but also by an Flask server, where the textual and audio classifier are placed, and by a Firebase project. Then there will be a description of the experimental results that have been carried out for testing the system on the task, those tests are done in order to prove the effectiveness of the system and to study its behavior and its power consumption. Finally, in the conclusion, there is a general sum-up of the experimental results, with some possible improvements and takeaways.

1 Introduction

During last years the emotional state and the emotional health have become an important argument of discussion in the society, in fact more and more specialists think that the psychological health is important as well as the physical one. Thus is very important to find new ways and new technologies to detect psychological and emotional issues in the people, this can help to diagnosticate mental issues and to improve quality of life of human beings.

Emotion Detection is a technology that can be exploited also by companies to understand people perception of a particular product or of a particular advertisement. This is useful in order to let the company aware how the people react and which are the things that must be changed in the product or in the advertisement.

Such tasks are commonly performed on images and text data. For instance, performing emotion analysis from Tweets taken from Twitter is a huge research topic.

Indeed smartphones are very popular nowadays as well as chat applications that are wide spreading at unforeseen velocity:mobile apps such as WhatsApp counts approximately 2 billion of monthly active users all over the world. Those applications involve different sources of data that can be sent: two of the most common communication methodologies are classic text messages and audio messages. Those could be used for making emotion detection more feasible since there is available a very large dataset and probably involves a more heterogeneous set of users.

2 Architecture

The developed solution for performing Emotion Detection comprehend different modules: a Chat Application for gathering data to be classified, a Firebase project for storing users, messages and labels, and a Flask application that exposes a Rest-Api for performing the actual classification.

Graphical user interface

Description automatically generated with medium confidence

Figure 1: System Architecture

2.1 Chat Application

In order to make possible the gathering of data for doing the emotion detection, a chat application has been developed. The aim was to take inspiration from some popular apps available such as WhatsApp or Telegram for creating the chat interface. Some functionalities such as loading of messages through scrolls has been added in order to reduce the number of messages to a fixed size and also to reduce the actual reads in the Realtime Database of Firebase.

<<chat interface image>>

**Figure 1:** Chat Interface

Users have the possibility to send two different sources of data, as said before: text messages and audio messages. During the first phase of development a popup could appear, asking to the User to perform some manual labeling for a subset of messages in the chat. This was done to gather a sufficiently large dataset to obtain a ground truth for evaluating the models that have been adopted.  
In the right-corner of the chat interface there is an emoji which represents the ongoing perceived emotion obtained from the last messages.

2.1 Firebase

The data that is utilized for performing operations in the chat are collected through Firebase service: chat messages and users are stored in the Firebase Realtime Database and firebase is also used in order to authenticate users. Then audio messages and labels are stored by exploiting the Firebase Storage.

2.1 Models for Emotion Detection

Emotions can be modeled in many different ways, but the adopted approach consist of modeling emotions from a fixed set of labels: joy, sadness, neutral, fear, anger. The neutrality has been added, as indicated in [5], for not creating biases with uncertain labels. For the emotion detecting we used two type of models, the SVM (Support Vector Machine) model for the text emotion detection and the MLP (Multi-layer Perceptron) model for the speech emotion detection. We have created a flask application to implement two REST APIs, one for the text and one for the speech, to make emotion prediction. The APIs are:

* predict\_text\_emotion – POST
* predict\_voice\_emotion – POST

The flask application has been deployed on Heroku which is a platform as a service (PaaS) that enables developers to build, run and operate applications entirely in the cloud.

3 Experimental results (or another name for the section)

The application has been tested and analyzed on two specific parts: model analysis and power consuption analysis.

3.1 Model Analysis

The model analysis has been performed for the text and for the audio parts.

3.1.1 Text analysis

The text analysis has been performed considering four different scenarios that are described in the table 1.

1. Scenario 1: Using WASSA dataset to train the model and to test it.
2. Scenario 2: Using a combination of the WASSA dataset and our labels given with the application to train and test the model.
3. Scenario 3: Using only our labels to train and test the model.
4. Scenario 4: Using the WASSA dataset for the training and our labels for the model.

As we can see in the table 1, the model performs better when we use only a dataset source to train and test the model. This is due to the fact that two different sources can label the same message in two different emotions, adding a bias to the model.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | WASSA TEST AND TRAINING | WASSA + USER TEST AND TRAINING | USER TEST AND TRAINING | WASSA TRAINING AND USER TEST |
| ACCURACY | 0.83 | 0.50 | 0.64 | 0.38 |

Table 1 Model Analysis

3.1.2 Audio Analysis

The audio analysis has been performed analyzing the accuracy of the model on our audios and labels. The results can be seen in the confusion matrix.

Graphical user interface

Description automatically generated

The accuracy that we have obtained is 38%, probably due to the fact that we have tried to simulate emotions and we used a few number of instances for each classes.

3.2 Power Consumption Analysis

To analyze the power consumption of our devices while using FeelChat, the tool battery historian was adopted. An automated common usage of the application was simulated with adb command line tools: the same chat was loaded with 15 seconds, for a total of 20 minutes. The device screen brightness was fixed during the whole experiment, caching of audio messages was disabled (in order to simulate the incoming of audio messages) and devices were connected via WiFi on the Internet. By varying the content of the chat, in terms of chat messages and audio messages, we have analyzed the power consumption of FeelChat, as shown in the following table:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | Oneplus 7T | Oneplus Nord | Mi 10 Lite 5G |
| Experiment #1: 10 text messages | **Device Estimated Power Usage** | **0.03%** | 0.04% | 0.05% |
| **Wifi Data Transferred** | 0.78 MB | 0.38 MB | 0.68 MB |
| Experiment #2: 9 text messages, 1 audio messages | **Device Estimated Power Usage** | 0.13 % | 0.06% | 0.37% |
| **Wifi Data Transferred** | 69.6 MB | 25.52 MB | 67.75 MB |
| Experiment #3: 7 text messages, 3 audio messages | **Device Estimated Power Usage** | 0.34% | 0.22% | 1.10% |
| **Wifi Data Transferred** | 211.1 MB | 106.11 MB | 225.8 MB |

Different considerations can be inferred from the latter results:

1. Loading text messages result in a minimum power consumption on our devices.
2. Downloading data from the Internet (such as audio messages) will result in considerably higher power consumption results

To reduce the power consumption, we checked if an audio file was already downloaded when loading a chat and we loaded a window of chat messages instead of the whole chat.

4 Conclusion

Observing the results of the experiments we can state that the application performs decently in the emotion classification of text, however, further improvements can be done gathering more data coming from testers. This could be useful to fight against the relativity of some kinds of words that could be classified differently by the users.

About the audio classification we must consider that classify the emotion only by MFCC coefficients it's hard, we can see the contribution of this model as a reinforcement of the detection made by text. About this, we could improve the general detection studying more complex policies to merge the classification of the 2 models.

In terms of power consumption, we can see that for different devices we get the same trend of increasing power consumption when the number of audio messages increase. To deal with this problem a possible solution could be to find a good compromise between audio quality and user experience, in this way we can reduce the amount of data to transfer and consequently the power consumed by devices.

REFERENCES

[1] Humaid Alshamsi, Veton Kepuska1, Hazza Alshamsi and Hongying Meng, AUTOMATED FACIAL EXPRESSION AND SPEECH EMOTION RECOGNITION APP DEVELOPMENT ON SMART PHONES USING CLOUD COMPUTING.

Department of Electrical and Computer Engineering Florida Institute of Technology Melbourne, USA

Department of Electrical and Computer Engineering Brunel University, London

[2] Albu, Alexandru & Spinu, Stelian. (2022). EMOTION DETECTION FROM TWEETS USING A BERT AND SVM ENSEMBLE MODEL. UPB Scientific Bulletin, Series C: Electrical Engineering. 84. 62.

Conference Name:ACM Woodstock conference

Conference Short Name:WOODSTOCK’18

Conference Location:El Paso, Texas USA

ISBN:978-1-4503-0000-0/18/06

Year:2018

Date:June

Copyright Year:2018

Copyright Statement:rightsretained

DOI:10.1145/1234567890

RRH: F. Surname et al.

Price:$15.00