



Emotion Prediction Application Using Voice Recognition Built with Neural Networks

Final Report

for

ENGR 498 Global Design Project II

213911362 Eren Ali Aslangiray

215141728 Mehmet Enis İşgören

214050652 Meryem Şahin

215231692 Sümeyye Sena Eminmollaoğlu

Barış Arslan

College of Engineering

İstanbul Şehir University

17 May 2019

Abstract

Acoustically expressed emotions can make the communication more efficient between humans and computers. We think that understanding emotions of humans is the next evolutionary step of interaction of one and other. In this project, our team used state-of-the-art models and techniques to analyze human voice signals to extract the features that computers need for understanding the emotions, as we humans do. This development allows all of the current available personal assistants to evolve and perform significantly better than their current version. The final aim of this project is letting computers to build empathy with humans to achieve the highest interaction quality. The deep details and full implementation of codes of this project can be found at:

https://github.com/Erenaliaslangiray/Emotion_Recognition_Application

Table of Contents

Abstract.....	2
Table of Contents	3
Introduction	4
Literature Review	5
Presentation of Model.....	7
Pipeline of the Project	8
1 st Model	8
2 nd Model	9
Text to Emotion	10
Sound to Emotion.....	10
Final Voting	13
Pre-Guided User Data.....	13
User State Graph	14
3 rd Model	14
Data Collected	15
Presentation of UI (Django Web Page)	15
Evaluation	17
Conclusion	18
References.....	19
Used Libraries	19
Used Resources.....	20

Introduction

As an individual, we all want someone or something that understands us and see what we are going through or how we feel. From the start of the civilization we humans come together to overcome the problems we have and socialize to make every individual's life better. From that perspective, our aim arises. The mood of a person may get affected from various things such as weather, time of the day or even the day of the week, likewise life. It keeps on going. We are willing to give a better daily life for individuals to have a good impact on their mood. Our design aims to predict users' emotions and their response to certain events with the help of previously recorded reactions of users.

Voice is the most common communication tool used to interact with one another and also, emotions are essential for natural communication between humans and have recently received growing interest in the research community. Today, voice and emotion recognition applications are available for limited areas, but these areas are increasing steadily. In these applications, the machine detects the sound and gives appropriate answer to it. While communicating with the phone, people prefer the sound rather than using keyboard, because it is easy and fast. We designed an application which will improve the currently used assistants such as Siri or Alexa, and with the help of this we will have an assistant which will develop empathy with the user and this system will be able to make mutual conversations. The application is going to ask some general information about users (Pre-Guided User Data aka. PGUD) to learn what they like or dislike and their behaviors in certain situations at the beginning. Then it will get users' voice and understand the meaning of the words and sentences in general. Not just the meaning, but also emotions will be recognized by the application from the voice. So that it will evaluate mood of the users at that time and return with some recommendations such as a movie, outside activity, food, product advertisement etc. Then Assistant will ask if the users are satisfied from that recommendation and according to the answer, it will update its decision tree (User-State Graph aka. USG) and adapt to upcoming user preferences.

In a nutshell, our goal is to improve people's daily lives by learning their moods and thoughts at any time of the day. So that we can make suggestions to make them feel better and to make them evaluate their free times. By this motivation, we foresee that our application will

make people feel less lonely and feel more being understood. More importantly we intend to break the barrier of the idea of talking to the “cold machine”.

Literature Review

There are many studies in the area of speech recognition and there are basic approaches that used in this problem. As discussed Gevaert, Tsenov and Mladenov in their paper, general structure of a speech recognition consists five steps which are; speech, signal processing, feature extraction, speech classification and output. In addition, there are commonly used techniques to achieve this problem. Dynamic Time Warping (DTW) compares words with reference words, Hidden Markov Modelling (HMM) splits the speech into small entities and it compares with the best-suited model. Another technique is Neural Networks which we will also be using in our project are similar to HMM, but Neural Networks use connection strengths instead of probabilities for state transitions (Gevaert, Tsenov & Mladenov, 2010, p.2). The article is mainly focusing on Neural Networks so it is very useful for our project.

After the process of getting voice from the user and converting it to the text, with Natural Language Processing the meaning will be extracted from the text. There are 2 methods to do this which are Natural Language Processing for Speech Synthesis and Natural Language Processing for Speech Recognition. NLP for Speech Synthesis is based on text to speech conversion and it uses the sentence segmentation which deals with punctuation marks with a simple decision tree (Reshamwala, Mishra & Pawar, 2013, p.113). On the other hand, NLP for Speech Recognition is based on the grammar of a language (Trilla, 2009, p.3). It is needed to use Natural Language Processing in our project in terms of understanding user's speech and return with some valuable suggestions.

Decision trees are a decision support tool for regression or classification models. According to Jordan, Ghahramani and Saul in their research, we have to know probabilistic decision trees and Hidden Markov models to understand Hidden Markov decision trees. In probabilistic decision tree, decisions are modeled probabilistically and recursion spreading upward and downward in the tree. In Hidden Markov model, the key calculation fit in it, recursion extending forward or backward in the chain. Hidden Markov decision trees are

probabilistic decision trees (upward and downward) with Hidden Markov model (forward or backward).

The most important packages that we used are Keras, Librosa, Fasttext and MongoDB. Keras is an open source deep learning library written in Python. It is capable of running on TensorFlow or Theano. In the Deep Learning applications, Keras helps us both in realizing the prototypes of our models and in the learning process. We use this in fine-tuning layers, hyper-parameters, creating the model, downloading the data sets and making the entries in the neural network. With Keras, models can be easily deployed to other frameworks and it turn model to products easily.

Librosa is a package for music and audio analysis in python. It is the reference of implementations for some more commonly used methods, helpers and shortcuts. The Librosa implements the various common functions used in the field of music and sound information retrieval.

FastText is a library for text classification and word representation which are fundamental to Natural Language Processing (NLP). It allows us to train supervised and unsupervised representations which is very useful for emotion recognition from text. FastText is able to train continuous bag of words (CBOW) and Skip-gram models. It uses a hash table for either words or character n-grams. FastText also offers Text Classification using machine learning and to do this, it needs the labeled data. In our application when we convert voice to text, we build a second model which extracts the emotion from text by Text Classification and contribute the other model to decide the final emotion state of the user.

MongoDB is basically a database which stores the data in flexible, JSON-like documents that is to say document fields can change from document to document which is what most type of the database services are not being able to do. Python provides a library called pymongo to drive MongoDB operations. We store the users' data in a pymongo database which allows us to make insert and update operations which adds the users' data into database in the first place and get those data whenever user speaks to the application and it offers updating database easily.

The two terms that is we developed and named are Pre-Guided-User-Data and User State Graph.

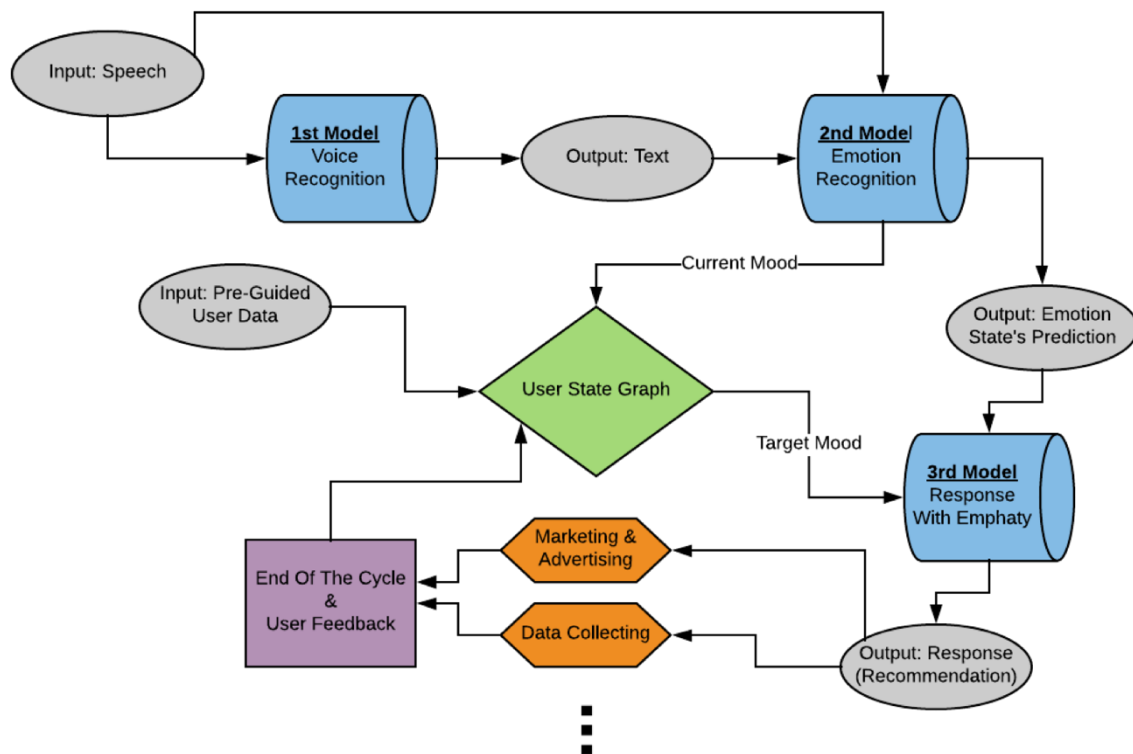
PGUD: Pre-Guided User Data contains basic information about each user. At the very beginning of our application, we want to know the user from different aspects so that we can build a basic User State Graph (USG) and we want to be able to know the actions of the user at certain moods or situations. For getting Pre-Guided User Data what we did is to make a quick test which contains 3 parts; Basic Information, Personal Information and Situational Information. Each of these parts gets different information and forms a knowledge tree. So, at the very beginning of our application we are able to know each user to use this information to guide them in each state of their moods.

USG: Every user has a graph which defines their situational moves to change their moods. As mentioned above, all users have a basic user state graph derived from Pre-Guided User data. This graph contains 4 basic moods which are angry, happy, sad and neutral. For every mood there are transition probabilities to an ideal mood that is to say some recommendations which user may prefer to do in that current situation. In this graph we do not offer user to take him/her from that current situation but we offer to make them pleased with our recommendations. According to the feedbacks that users give to recommendations the graph changes its transition probabilities.

Presentation of the Model

This project contains Machine Learning, Deep Learning, Natural Language Processing (Aka.NLP), Graph Theory, Decision Trees and Markov Models, Web Development and Signal Feature Extraction Techniques. All the coding done in Python 3.6. At deep learning models we mainly used both TensorFlow and Keras libraries. For NLP and word vector representations we used FastText. For feature engineering and feature extracting we used Librosa and FFMPEG. We did generate and use pseudo data from maps and weather. Moreover, we did combine all the algorithms in a single web application which is implemented with Django Library.

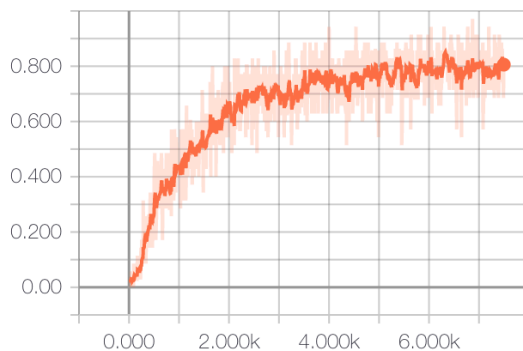
Pipeline of the project:



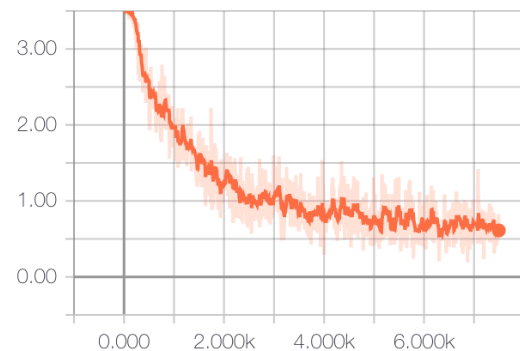
1st Model

First model is a basic speech recognition algorithm. Main goal is converting the human speech to text. In first period of our project we managed to implement speech recognition with over 82% accuracy with test dataset. It has total of 36 words in its library. These are its performance matrices at 0.8 smoothing value:

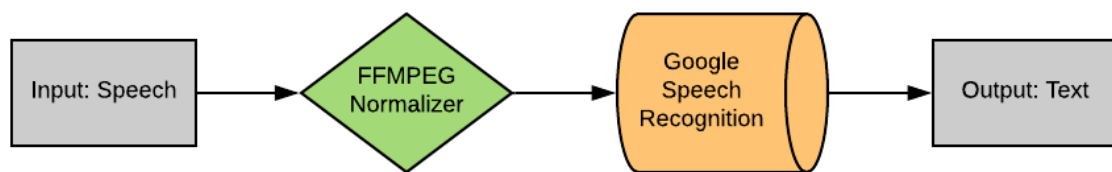
accuracy/accuracy



loss/loss_1

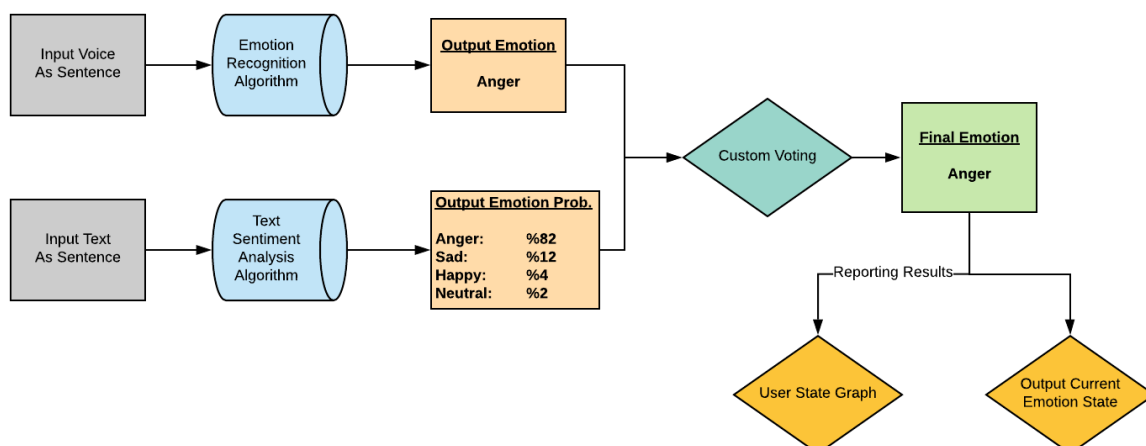


It was a successful model but unfortunately, we are not able to scale it big enough to understand whole English language due to lack of computation power and lack of data. Instead of it we used open sourced Google's Speech Recognition library. But before using our main voice to any of the algorithms that we are going to use, we first normalized the power of the sound. So that we eliminate soft speaker – loud speaker confusion and made our input data normalized for all of the prediction models to work healthy with it. The pipeline of the first model is shown below:



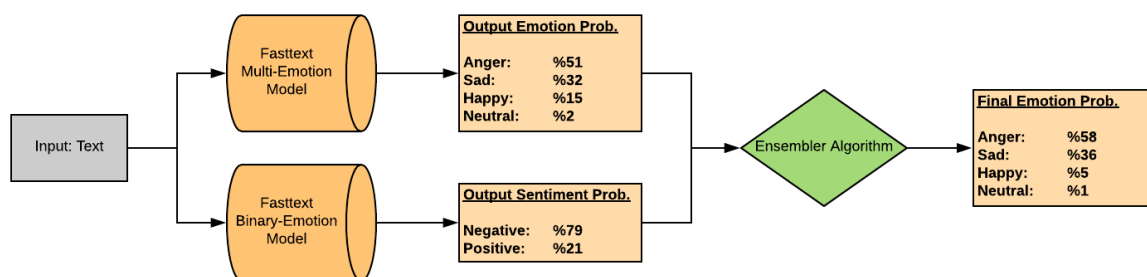
2nd Model

Our second model is the heart of our project and its backbone has ensemble multi-model structure. It has a sophisticated structure that took most of our time to design and implement. At first, we wanted to use 7 main emotions as psychology science defined for basic emotions of a human but lack of training data and for increased performance we decided to drop 3 of those emotions which are disgust, surprise and fear. We kept neutral, anger, happiness and sadness. Pipeline drawing of the 2nd model is like this:



- 2nd Model – Text to Emotion

We planned to extract emotion from sound and also the meaning as coming in as text. As shown in the figure above we have 2 more sub-structures in our second model. Which are Text Sentiment Analysis (Understanding 4 emotions from the context of what being inputted) and Emotion Recognition (Understanding 4 emotions from the sound of what being inputted). We used Facebook's FastText library to build text to emotion (T2E) model that works over 90% accuracy for given text input. Also, for the T2E algorithm it has 2 machine learning model inside of it. First of these models is doing multi-label classification such as 4 emotions that we defined before. Second of these is doing binary classification such as "Positive Emotion" and "Negative Emotion". They are all sensitive models that returns confidence levels of each output so that we combine these two models into one and return it for the voting part. In the voting part we are getting the sentiment result and buffing or nerfing the emotion results with respect to that. We also have penalty ratio defined as 1 at initial to increase or decrease the effeteness of our combiner result. The pipeline of the model is shown below:

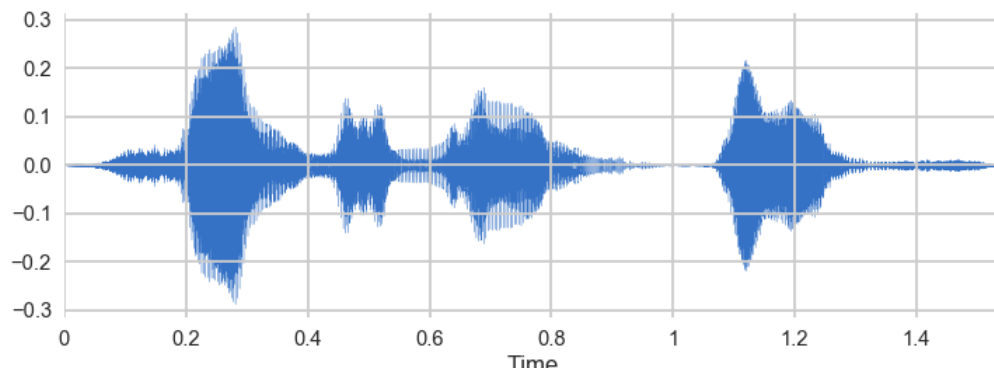


- 2nd Model – Sound to Emotion

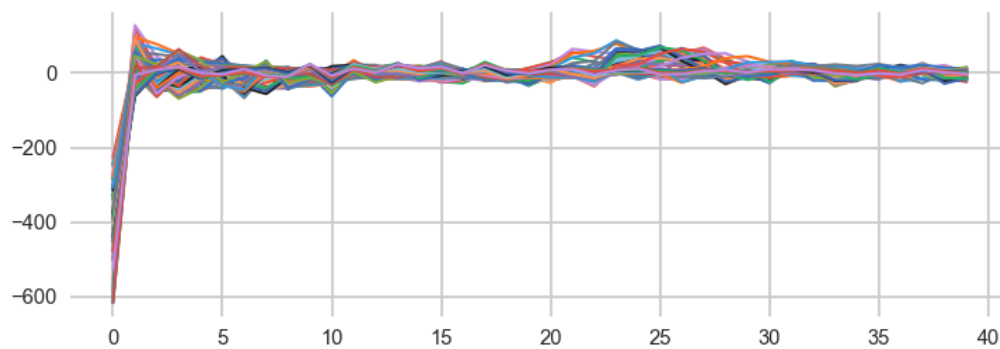
For sound to emotion (S2E) model, again we used multiple models that works and supports each other with voting. Model trained with multiple information extracted from the sound input such as Mel-frequency cepstral coefficients (MFCCs), amplitude spectrogram to dB, sampled Wavelet and zero-crossing rate. The 4 model that will work together will all be implemented with Keras and all of them are deep learning models. At the end we used voting with confidence levels considered to get one common result from all these 4 models.

The features are visualized as following (All the plots are made with same voice file which is anger labeled):

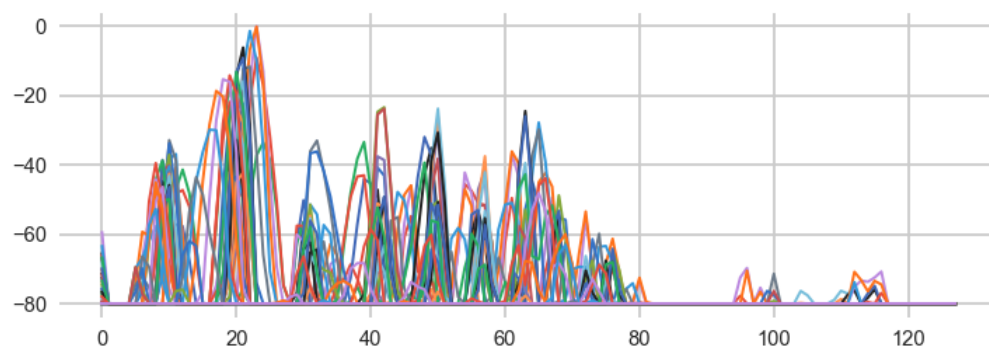
- Original wave:



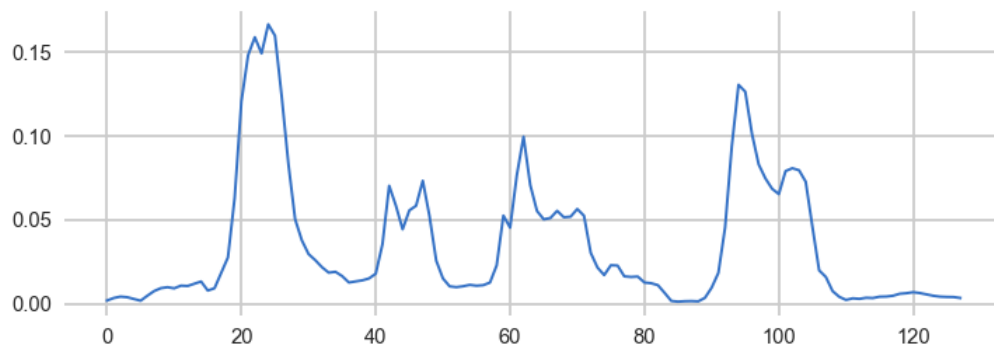
- MFCCs:



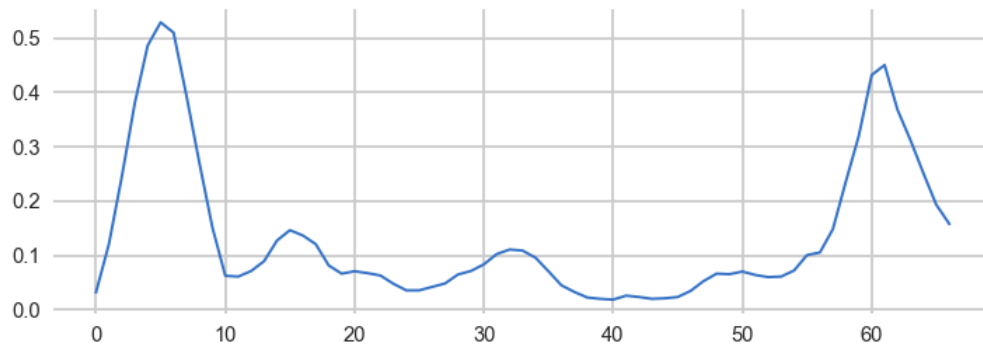
- Amplitude spectrogram to dB:



- Sampled wavelet:



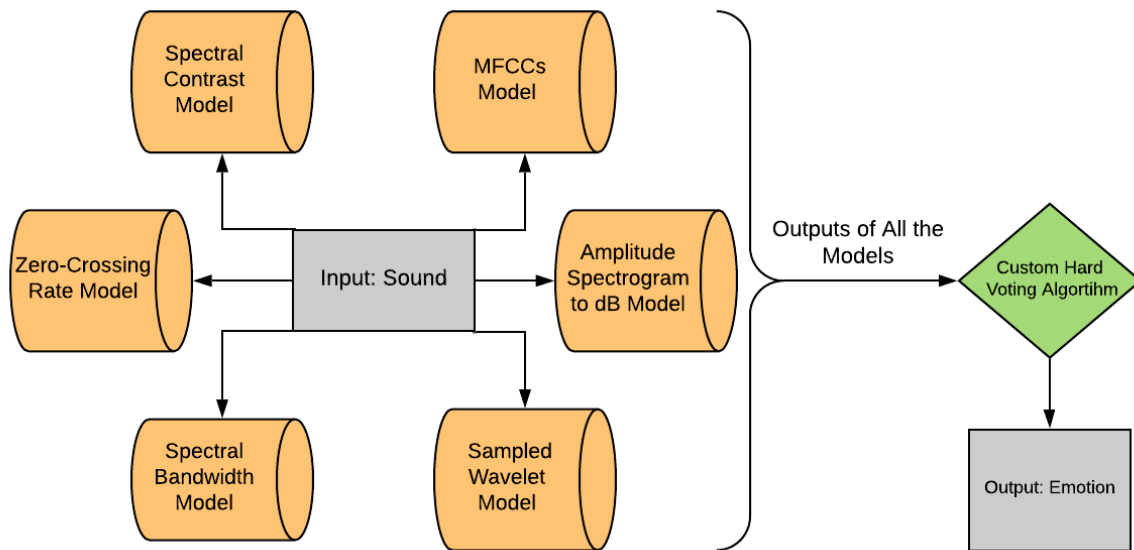
- Zero-crossing rate:



Also, the validation accuracies of the deep learning models are like following:

- MFCCs: %95
- Amplitude spectrogram to dB: %87
- Sampled wavelet: %81
- Zero-crossing rate: %81

The pipeline structure of ensemble models of S2E:



- 2nd Model – Final Voting

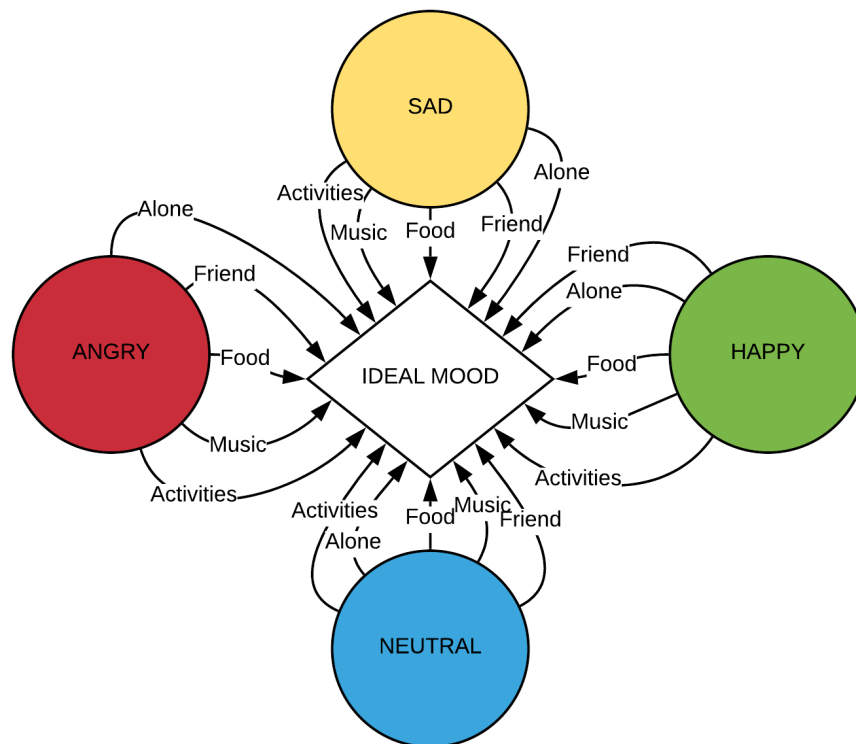
For the final voting part, we implemented our own voting algorithm that will work sensitively to merge these 2 results from T2E and S2E with respect to their confidence level. After this step we will have the emotion predicted.

Pre-Guided User Data

For PGUD, we prepared set of questions with psychology professors in Şehir University to understand and build the users base case of initial USG. The questions asked user's gender, preferences, weather likes and dislikes, and so on. At each user's first-time usage of our system, we will ask these questions and with respect to the answers, we will build the User State Graph. We did the survey to 350 people to get "average" responses of the crowd. So that our USG will have a base case that even our user wanted to skip answering those first-time questions.

User State Graph

USG is a unique idea that we come up with while thinking of having some kind of decision tree and personality tree and tracker inspired by Markov Models. A small example of it can be as following:



Please know that it is hard to draw its full-sized original graph so we plot a small example of it on above. It is a directed and weighted graph which will track the user's preferences and updates itself at each interaction with the user. The application will look at this graph to decide what it should do. The graph has its base case at initialization and this base case will be built with PGUD that user answered and also PGUD Survey.

3rd Model

3rd Model is a simple, non-complicated model that looks the PGUD and current emotion mood and also some third-party data such as weather, location, etc. to decide what to offer, or say back to user. It is important that this model is incremented and will increase its performance over time with each interaction of users.

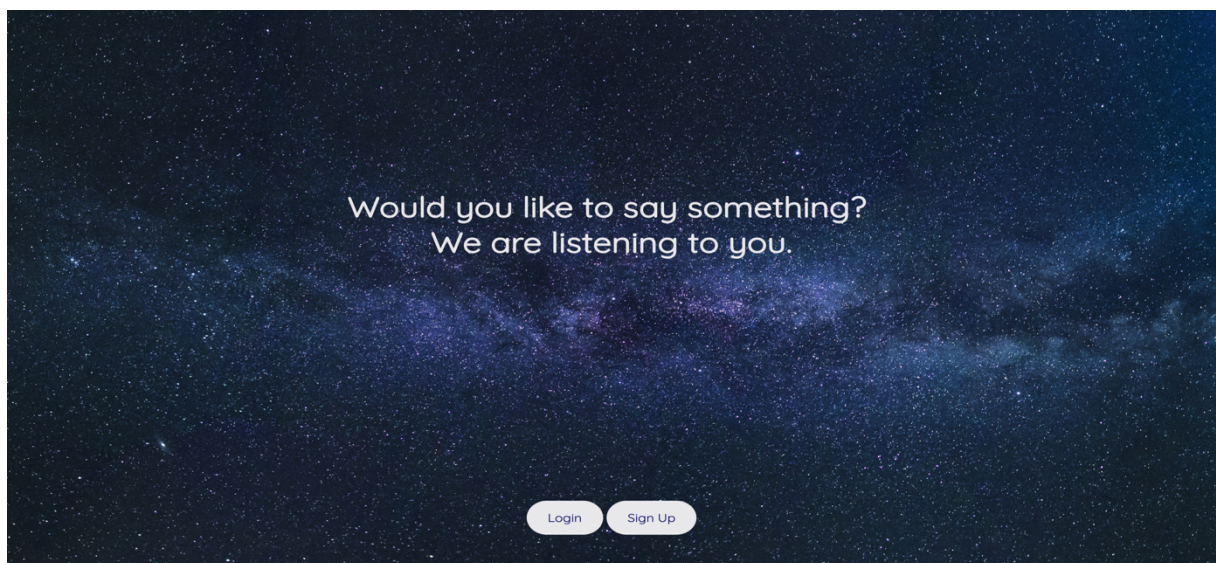
Data Collected

So far, we collected 4 datasets for speech recognition, 18 emotion or sentiment labeled text dataset and 5 emotion labeled voice datasets. Also, we did a survey with PGUD. In the end, we have a huge collection of datasets which is around 50 GB.

Presentation of UI (Django Web Page)

The UI part of project has a web page-based implementation. We designed it as if it is a real piece of product. So, we have working log-in, sign-up system with a working database. Also, a commercial homepage and we named our AI Hermes. At each user's first-time log-in we ask them PGUD questions to set their initial USG. At each user's login system will bring the users variables and graph to front to work with. So here are some screenshots of the web page system:

- Homepage:

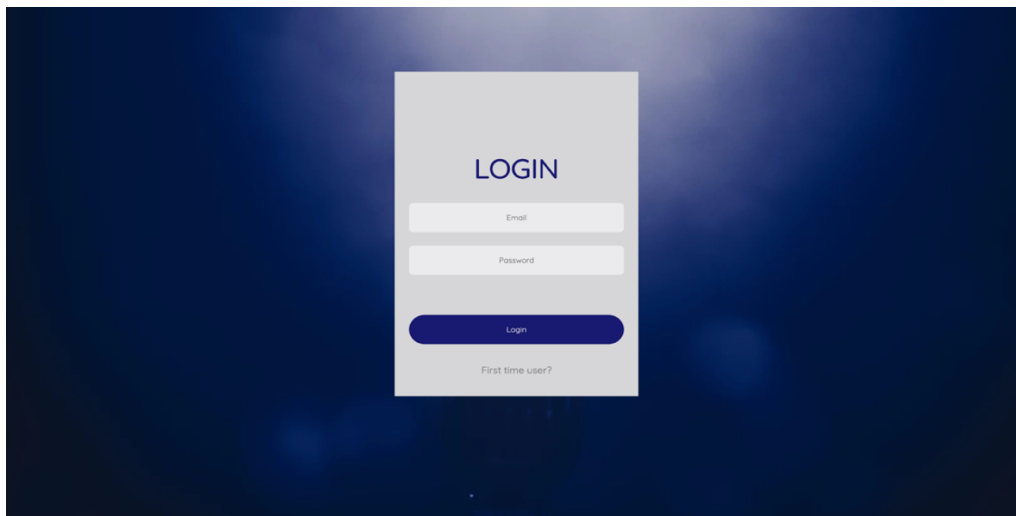


How does it work

Hi I'm Hermes. Basically, I am an AI that understands your feelings throughout your voice and context but I'm intent to do more. At the first time we meet, I have been told that humans ask each other some questions to know them better. So, I will try to ask you some questions to know you better.

I keep small notes about you on our each contact to became your best friend with understanding you better. When you don't like what I proposed, I will try to avoid it next time and opposite applies when you like. So, I am your emotional, empathically active friend AI that cares for you.

- Log-In page:



LOGIN

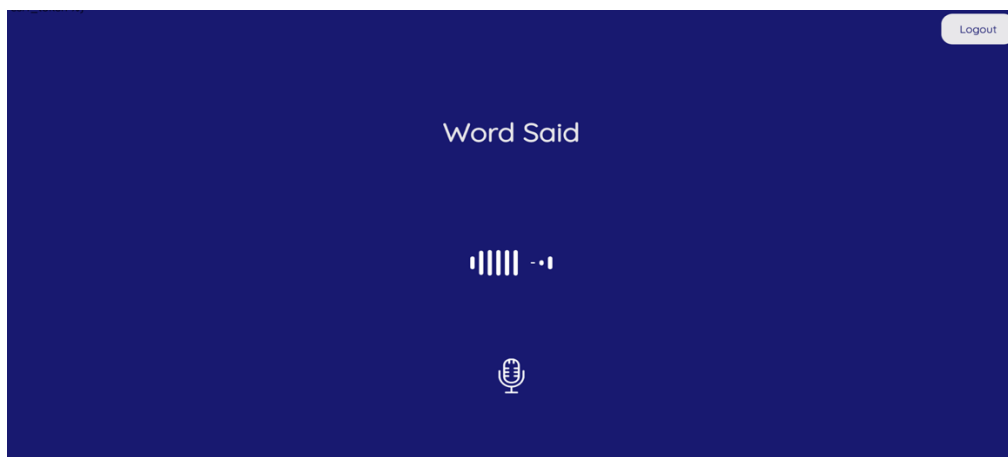
Email

Password

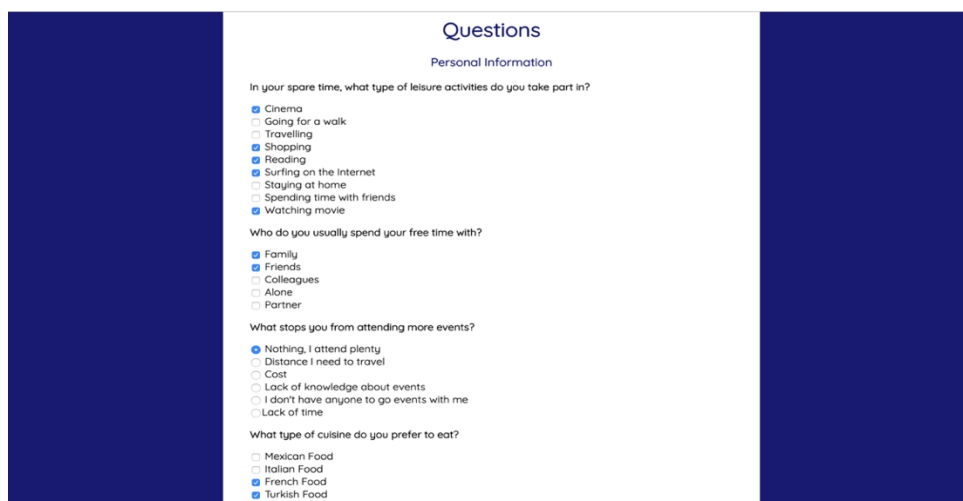
Login

First time user?

- User homepage:



- PGUD Questions:



Questions

Personal Information

In your spare time, what type of leisure activities do you take part in?

- ☒ Cinema
- ☐ Going for a walk
- ☐ Travelling
- ☒ Shopping
- ☒ Reading
- ☒ Surfing on the Internet
- ☐ Staying at home
- ☐ Spending time with friends
- ☒ Watching movie

Who do you usually spend your free time with?

- ☒ Family
- ☒ Friends
- ☐ Colleagues
- ☐ Alone
- ☐ Partner

What stops you from attending more events?

- ☒ Nothing, I attend plenty
- ☐ Distance I need to travel
- ☐ Cost
- ☐ Lack of knowledge about events
- ☐ I don't have anyone to go events with me
- ☐ Lack of time

What type of cuisine do you prefer to eat?

- ☐ Mexican Food
- ☐ Italian Food
- ☒ French Food
- ☒ Turkish Food
- ☐ Chinese Food

- Recommendation Pop-Up:

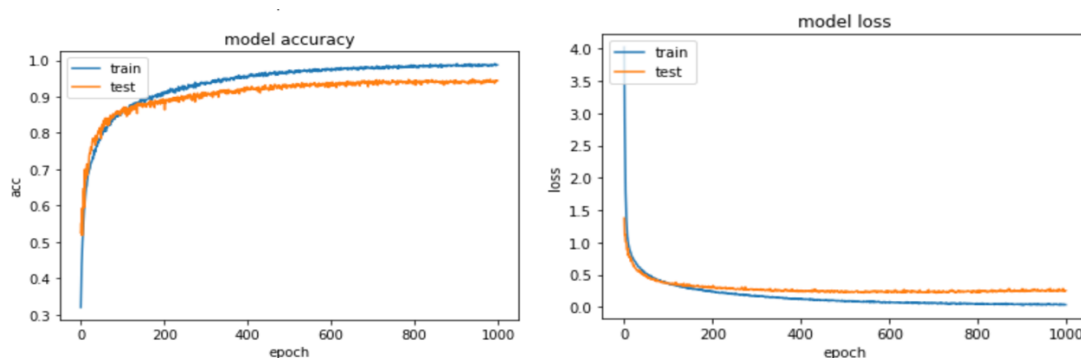


Evaluation

At the evaluation part, discussing our metrics briefly will be more than satisfactory. We had over %75 accuracy of predicting the emotion from user. With that we have confident that this kind of product will serve well.

Around 8 months of work time, we downloaded 50GB of various data and cleaned, studied the data and the terminology of the topic, did feature extraction on them, have built around 8 neural network models, build a website and build a working pipeline. All team members worked on different fields and at different tasks.

One of the highest performed deep learning model is built on keras sequential method and we used our own custom optimizer. The extraction method was MFCCs which is the closest filtering and sampling mechanism of human ear hearing mechanism and the method for mainstream speech recognition algorithm. The learning history graph was like following:



Conclusion

In the conclusion, it is a relief that achieving a thing that is not practically done and released before. We believe that such technologies as emotion recognition will be the next big step of evolution of personal assistants and the human-AI interaction. The results that we achieved in this project can help many scientists to develop even better algorithms and products.

In the end we have a machine that understand emotions of the person who speaks to it, like humans do. With the idea of making current technologies more human-likely this will be a big improvement. When we thought about solving the problem of being understood new problems arise. How realistic will it be? How scalable will it be when we think it about all cultures around the globe? What will be the next step of personal assistant evolutionary road? And many questions that made us think about the science we did in this project. Hope humanity and we scientist will find out all the answers.

References

- Used Resources

Bahl, L. R., Brown, P. F., Souza, P. V., & Mercer, R. (n.d.). Maximum Mutual Information Estimation of Hidden Markov Model Parameters for Speech Recognition.

Dasgupta, P. B. (2017). Detection and Analysis of Human Emotions through Voice and Speech Pattern Processing. *International Journal of Computer Trends and Technology (IJCTT)*, 3.

Gevaert, W., Tsenov, G., & Mladenov, V. (n.d.). Neural networks used for speech recognition. *Journal of Automatic Control*, 1-7.

Ingale, A. B., & Chaudhari, D. (2012). Speech Emotion Recognition. *International Journal of Soft Computing and Engineering (IJSCE)*, 235-238.

Jordan, M., Ghahramani, Z., & Saul, L. (n.d.). Hidden Markov decision trees. Canada: University of Toronto.

Jurafsky, D., & Martin, J. (2018). Hidden Markov Models. In *Speech and Language Processing* (pp. 1-17).

Lakomkin, E., Zamani, M., Weber, C., Magg, S., & Wermter, S. (2018). EmoRL: Continuous Acoustic Emotion Classification using Deep Reinforcement Learning.

Liang, X., Du, X., Wang, G., & Han, Z. (2018). Deep Reinforcement Learning for Traffic Light Control in Vehicular Networks. *IEEE Transactions on Vehicular Technology*, 11.

Mordkovich, A., Veit, K., & Zilber, D. (2011). Detecting Emotion in Human Speech.

Nam, J. (n.d.). Audio Representations. Graduate School of Culture Technology, KAIST, (p. 34).

Niu, Y., Zou, D., Niu, Y., He, Z., & Tan, H. (n.d.). A breakthrough in Speech emotion recognition using Deep Retinal Convolution Neural Networks.

Pitz, M., & Ney, H. (n.d.). Vocal Tract Normalization as Linear Transformation of MFCC. Aachen, Germany: University of Technology.

Reshamwala, A., Mishra, D., & Pawar, P. (2013). Review on natural language processing. *Engineering Science and Technology: An International Journal*, 113-116.

Trilla, A. (2009). Natural Language Processing techniques in Text-To-Speech synthesis and Automatic Speech Recognition. Ramon Llull University.

Yamato, J., Ohya, J., & Ishii, K. (n.d.). Recognizing Human Action in Time-Sequential Images using Hidden Markov Model. NTT Human Interface Laboratories.

- Used Libraries

django: <https://www.djangoproject.com/>

fasttext: <https://fasttext.cc/>

FFmpeg: <https://ffmpeg.org/>

Google Maps: <https://www.google.com/maps>

Google Speech Recognition: <https://cloud.google.com/speech-totext/>

Joblib: <https://joblib.readthedocs.io/en/latest/>

Keras: <https://keras.io/>

LibROSA: <https://librosa.github.io/librosa/>

NumPy: <https://www.numpy.org/>

pandas: <https://pandas.pydata.org/>

PyAudio: <https://pypi.org/project/PyAudio/>

PyMongo: <https://api.mongodb.com/python/current/>

PyWavelets: <https://pywavelets.readthedocs.io/en/latest/>

TensorFlow: <https://www.tensorflow.org/>

Special thanks to Dr.Ali Çakmak, Dr.Ayşe Reyhan Bilge Yıldırım, Dr.Barış Arslan, Dr.Cüneyt Utku, Assoc.Dr.Enes Eryarsoy, Dr.Ertuğrul Çetinsoy, Dr.Sami Anis Abuhamdeh for helping us and contributing to this project with their valuable ideas.