Trinity College Dublin
Coláiste na Tríonóide, Baile Átha Cliath
The University of Dublin

School of Computer Science and Statistics

Detecting Emotions In An Affect-Driven Conversational Model Using Utterance History

Sakina Vohra

[redacted]

September 21, 2020

A Master's Thesis submitted in partial fulfilment
of the requirements for the degree of
MSc. (Computer Science)

Permission to Lend and/or Copy

I, the undersigned, agree that Trinity College Library may lend or copy this thesis upon request.

Sakina Vohra

September 21, 2020

Declaration

I hereby declare that this project is entirely my own work and that it has not been submitted as an exercise for a degree at this or any other university.

I have read and I understand the plagiarism provisions in the General Regulations of the University Calendar for the current year, found at http://www.tcd.ie/calendar.

I have also completed the Online Tutorial on avoiding plagiarism 'Ready Steady Write', located at http://tcd-ie.libguides.com/plagiarism/ready-steady-write.

Signed: ______________________ Date: ______________________

Sakina Vohra September 21, 2020

Abstract

Most empathetic text-based chatbots recognise emotions in conversation at the utterance level and generate response depending on that emotion without analyzing the overall affect. Humans sense affect of another person through the context of the conversation. Similarly, a chatbot can analyze affect from same. However, it is very challenging to extract context from a single utterance. Different state-of-the-art models for conversational A.I. approaches have been proposed that extract contextual information using multiple utterances and different factors. The aim of this study is to understand can and with what relative performance do state-of-the-art conversational A.I. models detect affect across more than one utterance in contrasting conversations. The research was conducted in two parts. Experiment 1 compares and evaluates the performance of Bc-LSTM, DialogueRNN and TL-ERC in which Bc-LSTM outperforms other models. Experiment 2 evaluates which is the better model, Bc-LSTM or Transformer with pre-trained BERT for emotion classification on EmoContext SemEval2019 dataset in which the new Transformer model outperforms Bc-LSTM by a margin of 3% and a total Micro-average F1 score of 75.

Acknowledgements

Before beginning with the dissertation, I would like to extend my sincere and heartfelt gratitude towards all the kind people without the support of whom, this thesis wouldn't have been possible.

First and foremost, I would like to thank Prof. Vincent P. Wade BSc., MSc., MA., PhD., FTCD, CEO and Director of the ADAPT Research Centre for Digital Content Platforms and Application Research for accepting the supervision request and providing his invaluable guidance since day 1 of this journey. It was such an honour to work under the supervision of such a brilliant mind, whose expertise and experience in the field of Conversatinal A.I. helped my research reach the end it did.

I would also like to acknowledge few PhD students from Adapt Centre who spared some of their precious time in helping me with my research. Thank you so much, Emer Gilmartir, Mayank Soni and Christian Saam for sharing several useful research papers, videos and techniques to help with my study.

I also extend my gratitude towards several talented researchers and scientists, constantly striving to innovate something so that we have a better life.

Special thanks to my friends, who kept me motivated and calm during such difficult times due COVID19. All the "You can do it!" and "I believe in you" won't be forgotten.

Last, but not the least, I would like to thank the three most important people in my life, my Dad Shabbir Vohra, my mum Ummaysalma Vohra and my brother Murtaza Vohra for their unwavering support, unfailing patience and unconditional love.

Contents

List of Figures

List of Tables

1 Introduction

" A breakthrough in computer conversation would mean that technology is speaking our language and not the other way around." - QUARTZ

1.1 Chapter Overview

This chapter outlines the concept of an Empathetic Chatbot. It also identifies the ‘chatbot revolution’ which transformed emotionally aware chatbots from Rule-based systems architectures to a Data-driven architectures. Section 1.3, describes Affective computing and emotion detection, which is the core of an empathetic chatbot. The chapter also identifies the need for a ‘Friend-bot’ which underlies the motivation for this research. Finally, the chapter introduces the key questions being addressed in this research.

1.2 An Empathetic ChatBot

What distinguishes us from the robots are "feelings" (O’Brien 2019). Researchers have spent years trying to instil Artificial Intelligence to make machines appear to behave similar to humans. However, most of the work has only attempted to make it appear like its a human; the bot itself does not work like a human. A chatbot uses Natural Language Processing to understand user messages and generate meaningful responses. Frequently chatbots are used to reduce human intervention in carrying out tasks e.g. answering customer product queries on a website, providing recommendations on tourist applications etc. Chatbots could answer simple questions, book tickets, be a personal assistant, provide intelligent insights on business and data or act as a counsellor or nurse in the health sector. Chatbots are replacing humans for many tasks; however, when it comes to "emotion", chatbots are still not well advanced.

As of 2020, it is still a milestone for humans to say "Hey meet Ali; she is my best friend bot!" which means that it is still rare to find humans that consider an A.I. chatbot as a real friend. But why so ? Currently there are significant improvements required to have a ‘chabot as a friend for a human’. The key element needed for a chatbot to provide human-like friendship is empathy. Empathy, by definition, is the ability to understand and share the feelings of others. For us, our friends and family are the most empathetic as they "know us", they "understand us". Current empathetic chatbots have limited Emotional Intelligence, and there's much more research needed to build something that could ultimately replace a human friend (O'Malley 2018). An empathetic chatbot must be able to analyse the Cognitive empathy, Emotional empathy and Affective empathy of the user. Even in textual conversations, our friends could apprehend from our texting style, or reply delays, or from the look of conversation that we are feeling low or we are excited. For a bot to be able to understand a human, it has to do more than just detect emotions from single utterance[1] and generate a response. It needs to understand the Affect of the User.

1.3 The Need for an Empathetic Chatbot

Before starting to study about Empathetic chatbot, it is important to understand, why would one needs an Empathetic Chatbot. With an estimate of 3.6 billion people using social media in 2020 (Clement, J. 2020), people are found to spend about 3 hours of their day on the media, leaving them no time to socialise in person (Clement, J. 2019). An excessive Internet presence and an overwhelming amount of competition on the platform can lead to Social Anxiety Disorders that can disrupt sleep, health, and hamper both personal and professional life. 1 out of 8 people suffers from Social Anxiety Disorder(S.A.D.) at a point in their life (Thompson et al. 2019), leaving them feeling isolated, excluded or despised. Baumeister and Leary (1995) states that people suffering from S.A.D. find building even primary human relationships quite daunting. According to e Gennaro et al. (2020), the feeling of loneliness can eventually lead a person into having depression, developing inferiority complexing and stress anxiety. An isolated person declines any external help or attention due to the fear of judgement and low self-esteem. People avoiding social interactions look for other spaces to confide, and this is where the idea of an Empathetic Friend-bot shines. Imagine having a Friend-bot that learns from user conversations understands user personality, user moods, could talk 24 x 7 and is eventually able to realise that maybe "I am fine" is a lie. A program capable of doing so sounds far too ambitious. Still, before understanding the probability of having a bot capable of performing all the tasks as mentioned earlier, we need to comprehend if people would even want to use Chatbots. (Vaidyam et al. 2019) praises Empathetic chatbot for its ability to mitigate the negative impacts of exclusion as well as provide other features like understanding "emergencies", be highly available and provide both textual and voice-based communications (Ly et al. 2017). In 2019 State of Conversational Marketing Report [10], it was stated that around 64% of Internet users find the 24-hour service the best feature of a

[1] Utterance in a conversation is a message spoken or texted by a user

chatbot and the Opus Report says that about $4.5 billion will be invested in chatbots (Drift and SurveyMonkey 2019).

1.4 Affect Analysis and Emotion Detection

The one buzz word amongst all empathetic chatbots is "Emotion". Empathetic chatbots tries to be as emotionally aware as it can be. Several papers use terms emotion and affect interchangeably (Asghar et al. 2018; Colombo et al. 2019) but are they the same?. As per Vand (2019), an **Affect** *"is a term that encompasses a broad range of feelings that people experience. It embodies both emotions and moods"*. An **Emotion** *"is an intense feeling that is short-term and is typically directed at a source. Emotions can often have indicative facial expressions and body language."*. A person could feel the emotion "excited" because he/she won something, the affect here is happy. Here excited is directed at the source winning something. A drip stream by Tom Lee (P. V. Lee T. H. 2020) states that Emotion is a label given to a broader feeling called 'Affect' one is experiencing. Affect, on the other hand, is what drives the resulting emotion. Affect is the natural sub-conscious state of feeling. In a conversation, we see chatbot appearing to understand emotion; however, chatbots should be adaptive to understand a person's affect. Affective computing is widely recognised and established as of today (Colombo et al. 2019). In the past few years different researchers have successfully been able to classify discrete emotions from verbal and non-verbal behaviour or estimate a value of a component of affective state like valence or intensity. Sensing emotions and affect in the text is fundamental for generating empathetic conversations, opinion mining, as well as to detect personality traits. In normal conversations in everyday life, a user may experience one of the six emotions described in Ekman Model (happy, sad, angry, fearful, disgusted and surprised) (Ekman 1992) and an Affective chatbot can detect these. Linguistic clues in text provide significant insight on emotion, for example, "I had a good day!", the word good leaves an impression that the user is happy. Tallec et al. (2011). discusses how linguistics of an utterance could be used to detect emotion in a Companion Robot. However in textual conversations, just considering one utterance could detect the emotion of that message but would fail to analyze overall affect of the person. In a voice-based system, when a user says "I am fine", the tone and different analogous signals could help understand the emotion however a text message doesn't have any such features which makes it difficult to detect the "real" emotion. Humans figure out the emotions of other people in textual conversations by noticing changes in behaviour, reply delays, use of emojis, length of sentences, state of preceding utterances, reaction to turns. Human sense the mood from "the conversation" and can debate, "you say you are fine, but you don't seem fine"; a chatbot that only acts as an answering machine, cannot make such debates. A chatbot needs to take into account more than just a single utterance. To build an empathetic-friend chatbot, the history of utterances and emotion associated with preceding utterances needs to be considered WHICH can give more context for better emotion recognition in the conversation.

1.5 Research Objectives and overview

A typical chatbot reads an utterance and replies to it. Currently, emotionally aware chatbot understand the emotion of the utterance and responds based on the emotion of that utterance. An empathetic Friend bot will understand the "Affect" from the conversation and try to engage user in a more perceptive way, i.e., respond based on the emotion of the user. In order to analyse affect or detect the emotion in the conversation there is a need to consider a greater number of utterances and the emotion associated with the preceding utterances.

Our Research Question: Can and with what relative performance do state of the art emotion detection chatbots detect affect across more than one utterance in contrasting conversations?

This thesis performs an extensive research on Affective Computing and Emotion Detection, as discussed in Chapter 2. A strong study of Emotion Recognition in conversation unveiled three state-of-the-art conversational A.I. models that has so far worked best in understanding the emotion in conversation. These three models considers different features like surrounding context from utterances, history of utterances, the context associated with utterance and the emotion associated with previous utterance. Every approach proposes different modelling technique and promises to work the best. This thesis will compare all the three model to find the best model for Emotion Recognition in Conversation. The best model will then be used to "Predict" emotions for EmoContext Dataset. EmoContext is the third task by SemEval2019, that raises the same question as this thesis, can emotion classification be performed using multiple utterances ? The main aim of this research is to understand if the hypothesis that multiple utterances gives more context for emotion modelling, is true or not. If it is true, can emotions be recognised using multiple utterances, if yes, then how ? To understand this, two experiments are performed :

Task - 1: Evaluating 3 State-of-the-art conversational models with 2 different conversational datasets and compare the results to understand the best model

Task - 2: Use the best found model to predict emotions for SemEval2019 EmoContext dataset using more number of utterances in a conversation

These experiments expects to answer the followings questions :

1. What is the best approach for building an emotion recognition model for an empathetic chatbot that considers multiple utterances from a conversation ?

2. Are there any other features that would help predict emotions better ?

3. A.I. based models require dataset to train upon, what dataset would work best ?

4. If "perfect" dataset is not found, Can an emotion prediction model be built using Semi-supervised learning ?

5. Which model worked best for emotion prediction on EmoContext dataset?

Further in this research, we will discuss an extensive literature review on Affect Analysis and Emotion Detection and find the state-of-the-art modelling approaches in chapter 2. We will understand the critical concepts of Natural Language Processing and dive deeper into A.I. to understand R.N.N., L.S.T.M. and Transfer Learning. Chapter 4 discusses the methodology for experiment, where it describes the datasets used and the design architectures of Bc-LSTM, DialogueRNN and TL-ERC. Chapter 5 discloses the result of experiment 1 and the best performing model is chosen to perform in experiment 2. Chapter 6 discusses the methodology used for experiment 2,i.e, for emotion prediction on EmoContext dataset. Chapter 7 unveils the final outcome of the research. Chapter 8 discusses in detail all the findings observed and noted throughout the research. Finally, the thesis is concluded in Chapter 9 .

2 Literature Review

2.1 Chapter Overview

Chapter 2 presents the findings from the literature review designed to understand the best approaches for emotion recognition in textual conversation. The chapter first introduces the reader to the evolution of emotionally aware chatbots from rule-based to data-driven in section 2.2. The chapter then describes emotional intelligence in open domain systems in section 2.3, outlines different affective chatbots and approaches for building an affective conversational system in section 2.4. Finally, the chapter presents state-of-the-art models that detect emotions using utterance history

2.2 The Evolution of an Emotionally-Aware chatbot

In his seminal article called ‘Computing Machinery and Intelligence’ Alan Turing raised the question "Can a machine think ?" (Turing 1950); this question inspired many to develop intelligent systems, and it’s still on-going research. However, with the technology in those days, he found it very difficult to answer. He replaced the question with a more pragmatic one, "Can a computer communicate in a way indistinguishable from a human ?" (Warwick and Shah 2016) . He pioneered the chatbot by proposing the Turing test, historically known as the Imitation-Game (Turing 1950) to understand how well a machine to imitate a human. A human judge of either gender was chosen to judge "Textual" conversations between a human participant and a computer that was designed to generate human-like responses. If the judge failed to recognise whether the response is by a human or the computer, then the computer passes the test. A "Chatter-Bot" (now chatbot) Coined by Maudlin in 1990, is computer software that performs human-like interactions by mimicking human behaviour. A chatbot could be either rule-based or data-driven or both. A rule-based chatbot relies on symbolic knowledge; it uses pattern matching technique to associate keywords with different responses, whereas a data-driven chatbot A chatbot that was once only Rule-based can now either be Rule-based or Data-driven or both. A rule-based chatbot uses a dictionary that has keywords mapped with different responses wheres a Data-driven uses machine learning techniques to provide more human-like interactions

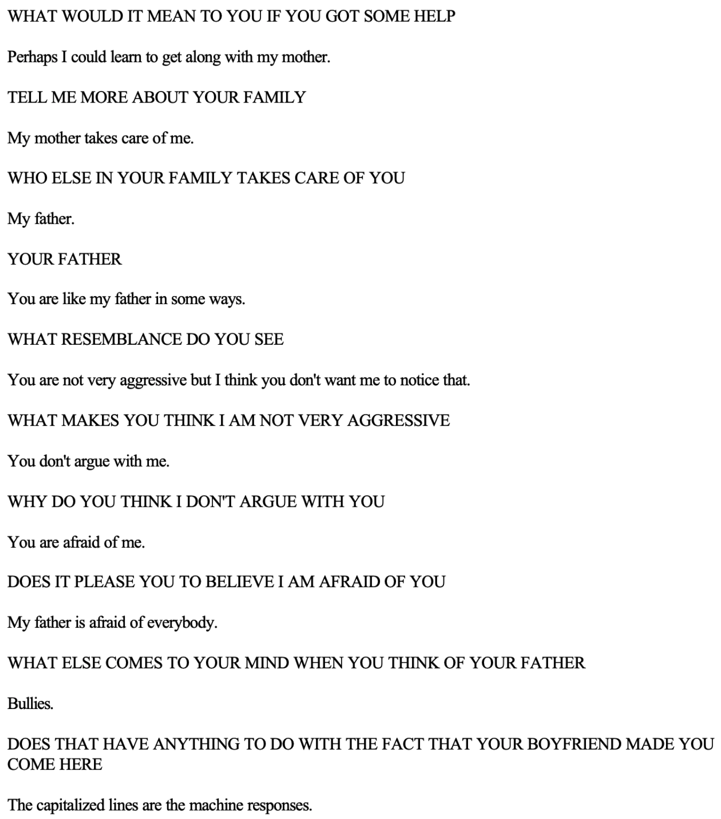
2.2.1 Rule-Based Chatbots

In 1964-6, Joseph Weizenbaum developed E.L.I.Z.A. (Weizenbaum 1966) in M.I.T. labs, which is considered the first-ever chatbot. The overall working of E.L.I.Z.A. is quite simple; it reads the input and looks for a "keyword". On keyword detection, the program transforms the sentence based on the rules mapped with the keyword using the SCRIPT [1]. Then the computed text is printed out. However, if dived deeper into the details, a task that looks manageable is found to have many complexities, one of which is discovering multiple keywords in a sentence. In this program, the keywords are prioritised, so when the program scans the sentence; it chooses the keyword with the highest precedence. Weizenbaum also deals with the issue when no keyword is found, by allowing E.L.I.Z.A. to be "intelligent" enough to generate a response on its own. However, this model doesn't take into consideration emotion as a factor when generating responses. E.L.I.Z.A is only programmed to ask open-ended questions based on a keyword found in the source utterance to leave an impression that it is empathetic. For example, if a user enters, "I hate my family", E.L.I.Z.A responds, "tell me more about your family" Below is an actual conversation with E.L.I.Z.A. taken from (ibid.) .

In 1972, a chatterbot was written by Psychiatrist Kenneth Colby at Stanford University, which simulates people with Paranoid Schizophrenia. This program tries to implement a model with a specific personality that mimics a person who has Paranoid Schizophrenia using a more scientific and cognitive approach. The bot consistently misinterpreted what people said, assumed they had nefarious motives and were always lying and wouldn't allow anyone to know anything about "PARRY" 's life. PARRY was an exciting program as it was the first-ever model that adapts a human personality (Zemčík 2019). Parry was also infamous for being known as 'E.L.I.Z.A. with attitude' (Phrasee 2016). And then in the same year, E.L.I.Z.A. met PARRY. At a computer conference in 1972, a conversation between the patient and the doctor was presented opening more doors to advancements.

Throughout the years, different chatbots were developed using Heuristic Approach and performed pattern matching to simulate human-like conversations. As stated in Shum et al. (2018) and Zemčík (2019), many Rule-Based have been proposed like Jabberwacky (1988), that learned everything that user said and performed contextual pattern matching technique on it, Dr Sbaito (1991), a verbal chatbot that used speech synthesis, A.L.I.C.E. (1995) , a program made using Artificial Intelligence Markup Language(A.I.M.L.). A.I.M.L. has tags that allow simplification of pattern matching. It won the Loebner prize in 2000,2001 and

[1] The SCRIPT in this chatbot is like a dictionary constituting of keywords and their mapping with transformation rules



WHAT WOULD IT MEAN TO YOU IF YOU GOT SOME HELP

Perhaps I could learn to get along with my mother.

TELL ME MORE ABOUT YOUR FAMILY

My mother takes care of me.

WHO ELSE IN YOUR FAMILY TAKES CARE OF YOU

My father.

YOUR FATHER

You are like my father in some ways.

WHAT RESEMBLANCE DO YOU SEE

You are not very aggressive but I think you don't want me to notice that.

WHAT MAKES YOU THINK I AM NOT VERY AGGRESSIVE

You don't argue with me.

WHY DO YOU THINK I DON'T ARGUE WITH YOU

You are afraid of me.

DOES IT PLEASE YOU TO BELIEVE I AM AFRAID OF YOU

My father is afraid of everybody.

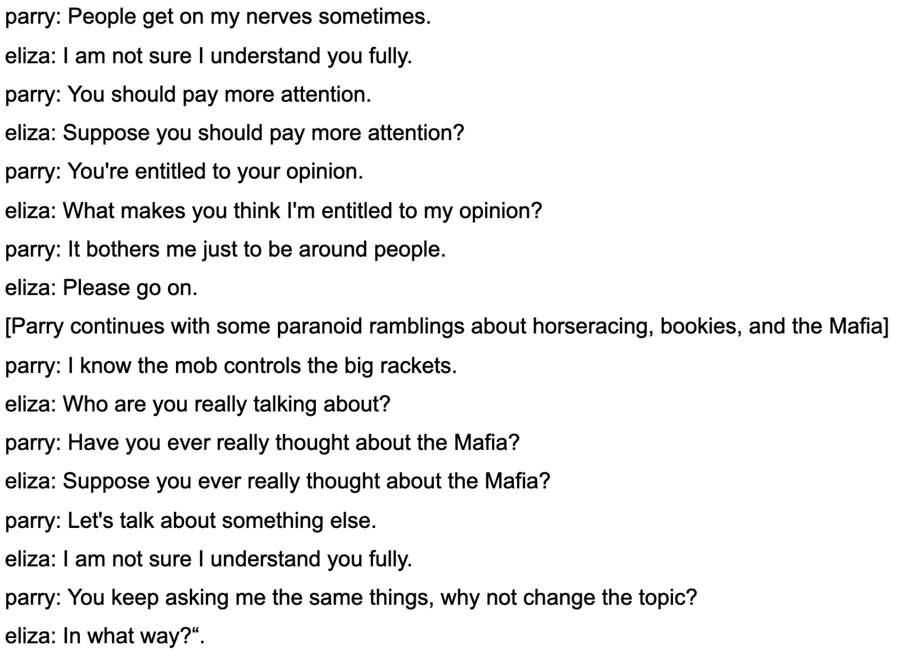
WHAT ELSE COMES TO YOUR MIND WHEN YOU THINK OF YOUR FATHER

Bullies.

DOES THAT HAVE ANYTHING TO DO WITH THE FACT THAT YOUR BOYFRIEND MADE YOU COME HERE

The capitalized lines are the machine responses.

Figure 2.1: Conversation with ELIZA



parry: People get on my nerves sometimes.

eliza: I am not sure I understand you fully.

parry: You should pay more attention.

eliza: Suppose you should pay more attention?

parry: You're entitled to your opinion.

eliza: What makes you think I'm entitled to my opinion?

parry: It bothers me just to be around people.

eliza: Please go on.

[Parry continues with some paranoid ramblings about horseracing, bookies, and the Mafia]

parry: I know the mob controls the big rackets.

eliza: Who are you really talking about?

parry: Have you ever really thought about the Mafia?

eliza: Suppose you ever really thought about the Mafia?

parry: Let's talk about something else.

eliza: I am not sure I understand you fully.

parry: You keep asking me the same things, why not change the topic?

eliza: In what way?".

Figure 2.2: When ELIZA met PARRY

2004. However, A.L.I.C.E. failed the ultimate Turing Test due to the limited capacity of A.I.M.L. that fails to hold the dialogue for an extended period

Even though Rule-based chatbots are winning hearts and impressing people as an empathetic chatbot and passing the Turing test, it is highly limited to what it can do. Rule-based chatbots are built upon symbolic knowledge that makes it brittle. They are very vulnerable and could break if fed data outside domain knowledge; however, it is highly used in services like answering customer queries or booking flights.

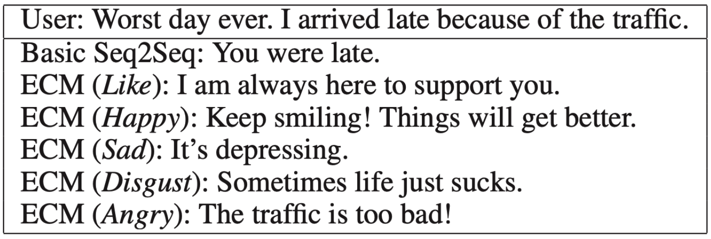
2.2.2 Data Driven Chatbots

As per Guzman (2017), the first-ever sophisticated Data-driven bot was launched in 2011 by Apple called Siri as a feature in its iPhone 4s version. Siri is a voice-based personal assistant chatbot known for its sassy replies, the capability of making interactions, answering questions and performing tasks. When S.I.R.I. catches a message, it considers it as Intent and looks for triggers within the Intent like "Set alarm" or "play music", however, this may raise a question, where is Artificial Intelligence used in this system? Behind the scenes, Siri regularly performs machine learning and improves itself by collecting offline data from the device without violating privacy. Siri did most of the groundwork for all other bots and personal assistant that came later.

In May 2014, Microsoft introduced XiaoIce (L. Zhou et al. 2018); the first A.I. based empathetic social chatbot capable of understanding emotions. As per L. Zhou et al. (ibid.) within the initial 72 hours of its launch, XiaoIce was looped into 1.5 million chat groups and became a cross-platform social chatbot in 2 months. XiaoIce uses Artificial Intelligence to understand human emotional needs and become a human companion. The bot is judged using C.P.S. metric, that stands for Conversation-turns Per Session that measures its effectiveness. The measure is to judge how long users could engage in a conversation with Xiaolace. Xiaolace maintained a high C.P.S. as its capable of responding as well as initiating a conversation, engaging the users more into the conversation (ibid.)

Four years later, in 2018, H. Zhou et al. (2017) proposed Emotion Chatting Machine that considers emotion of a response when generating a sentence. An artificially Intelligent bot is capable of learning from user data; this could mean, it would end up learning everything that is fed; even the harmful data. In 2016 as mentioned in Wolf et al. (2017), Tay, an A.I. chatbot launched by Microsoft had to be taken off just after 24 hours of launch as it started generating inappropriate, racist and sexist tweets. A chatbot should have a filter when generating responses. H. Zhou et al. (2017) uses "emotion" as a filter for generating empathetic and appropriate responses.

The research of rule-based and data-driven chatbots underlines the usability of both the approach in emotionally aware chatbots. Data-driven chatbot seems more promising and efficient as it can be scaled to incorporate different features using machine learning and natural language processing surrounding the utterances when detecting emotions.



User: Worst day ever. I arrived late because of the traffic.

Basic Seq2Seq: You were late.
ECM (*Like*): I am always here to support you.
ECM (*Happy*): Keep smiling! Things will get better.
ECM (*Sad*): It's depressing.
ECM (*Disgust*): Sometimes life just sucks.
ECM (*Angry*): The traffic is too bad!

Figure 2.3: Response generated based on different emotions

2.3 Emotion Intelligence in open domain systems

Out of all the discussed chatbots possessing the Emotion Quotient (E.Q.), not a single chatbot understands human emotion. It simply performs what it is supposed to when detected a feeling. Going back to the argument first initiated by Alan Turing, "Can machines think ?". We still don't have an answer to it. Even though E.L.I.Z.A was one highly appreciated Emotionally-Aware Chatbot for being capable of creating a remarkable illusion of having empathy; it had failed the Turing Test (Becker-Asano et al. 2007). An empathetic chatbot is more than just detecting emotion from an utterance and responding. A few years ago, Becker-Asano et al. (ibid.) raised a question, "Should we integrate emotions into conversational agents?". He researched and found that by adding primary and secondary emotion quotient into a chatbot, people are more likely to believe in it, see it human-like or can perceive the chatbot with personality. This theory is also supported by S. and Pu (2020). Becker-Asano et al. (2007) has added two layers within the emotion engine of the chatbot called Reasoning Layer, and other is called Reactive Layer. The Reasoning layer is where the chatbot's conscious lies; It allows the chatbot to explain why it feels what it feels. The Reactive Layer is the non-conscious/automatic state where the chatbot is aware of the emotion and responds to it accordingly. The Reasoning Layer in the emotion engine is unique and intriguing as it enables the chatbot to think and justify the feeling it has, making it more human-like.

In 2017, Portela and Granell-Canut (2017) tries to find if there is an emotional engagement between users and empathetic chatbots in smartphones. The reason for this research was inspired by Rosalind Picard, M.I.T. researcher's belief that a machine could effectively assist people if they recognise emotions. Picard had started the field "Affective Computing" when she published a book by its name that states " Computers that will interact naturally and intelligently with humans need the ability to at least recognise and express affect" (Picard 1997). In Portela and Granell-Canut (2017), several participants were invited to involve in long conversations with the chatbot and asked for feedback. Most participants believed that bots would usually repeat what they already know, albeit this helped some by making them more aware of their emotional state. However, the lack of external knowledge made it difficult to not see it as a human.

In a recent paper by M. Huang et al. (2020), it is claimed that the goal of an open-domain system[2] should be long-term user engagement. An open-domain system is often challenging to build as it is required to give recommendations, pitch in exciting topics, provide affective responses, comforting and understand user's emotional needs to generate interpersonal conversations with a consistent personality. M. Huang et al. (ibid.) provides four critical factors for building an empathetic chatbot, context, personality, emotion and behaviour as seen in the diagram below(Figure is taken from (ibid.)).

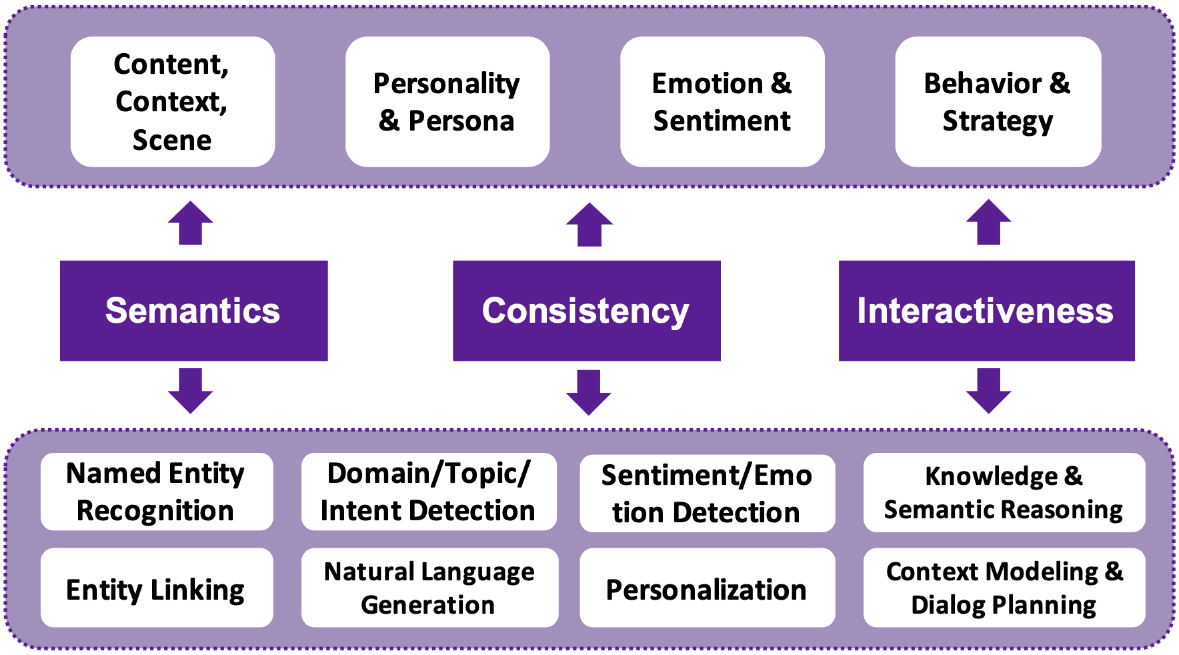
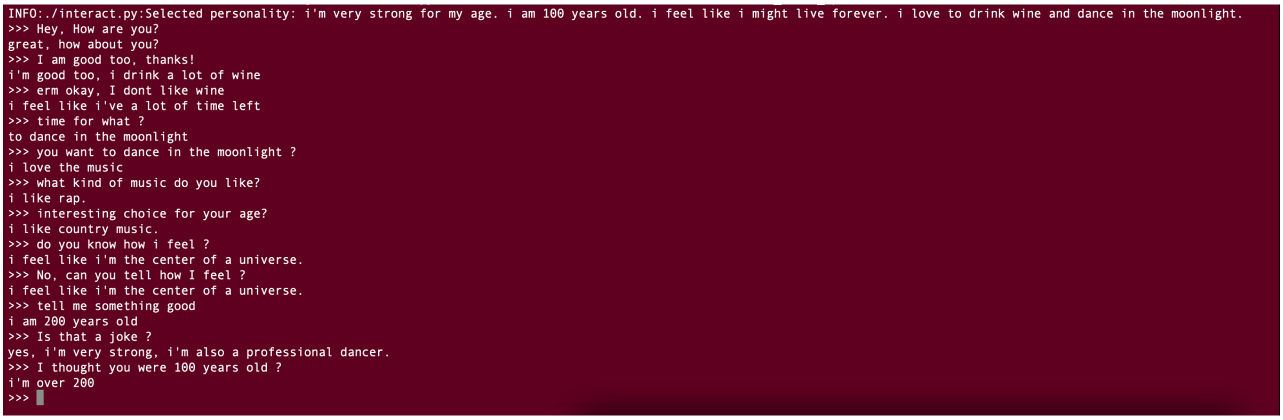


Figure 2.4: Key Factors, Key Issues and Key Technologies(Top to bottom) for building an empathetic chatbot

We understand for a bot to be empathetic, it needs to be Affective. But from a human perspective, for a bot to be more friend like, it should have a personality. Trust is a big contributing factor for a human when engaging in emotional conversations. A bot who likes blue on one day and red on other fails to build any specific personality making it difficult to form a bond. Hence a bot should have a consistent personality. The script of Huggingface [3] was tested to understand how the persona of chatbot affects conversations. Huggingface ranked first in Conversational A.I. competition by NIPS2018; it uses random pre-defined personality for interaction. It was found that it's inconsistency(See image below) demotivated user engagement from any further interactions; however, this model was only created for the purpose of the competition and could do better.

Another factor to note is that the conversation with a stranger and the conversation with a friend is different. For an open domain system to be a friend, it needs knowledge grounding

[2] An open domain system is one that can answer any question and isn't restricted to pre-defined topics.
[3] https://github.com/huggingface/transfer-learning-conv-ai



```
INFO:./interact.py:Selected personality: i'm very strong for my age. i am 100 years old. i feel like i might live forever. i love to drink wine and dance in the moonlight.
>>> Hey, How are you?
great, how about you?
>>> I am good too, thanks!
i'm good too, i drink a lot of wine
>>> erm okay, I dont like wine
i feel like i've a lot of time left
>>> time for what ?
to dance in the moonlight
>>> you want to dance in the moonlight ?
i love the music
>>> what kind of music do you like?
i like rap.
>>> interesting choice for your age?
i like country music.
>>> do you know how i feel ?
i feel like i'm the center of a universe.
>>> No, can you tell how I feel ?
i feel like i'm the center of a universe.
>>> tell me something good
i am 200 years old
>>> Is that a joke ?
yes, i'm very strong, i'm also a professional dancer.
>>> I thought you were 100 years old ?
i'm over 200
>>> 
```

Figure 2.5: A conversation with state-of-the-art conversation A.I.

to provide interpersonal response (M. Huang et al. 2020)). There needs to be a way for the bot to learn user characteristics and use that when generating responses. This research highlighted the existence and possibility of emotional intelligence in open domain systems, the significance of emotional intelligence in an empathetic chatbot and different characteristics that could help analyse the emotional state of a user.

2.4 Affect Analysis and Emotion Detection in Conversational Agents

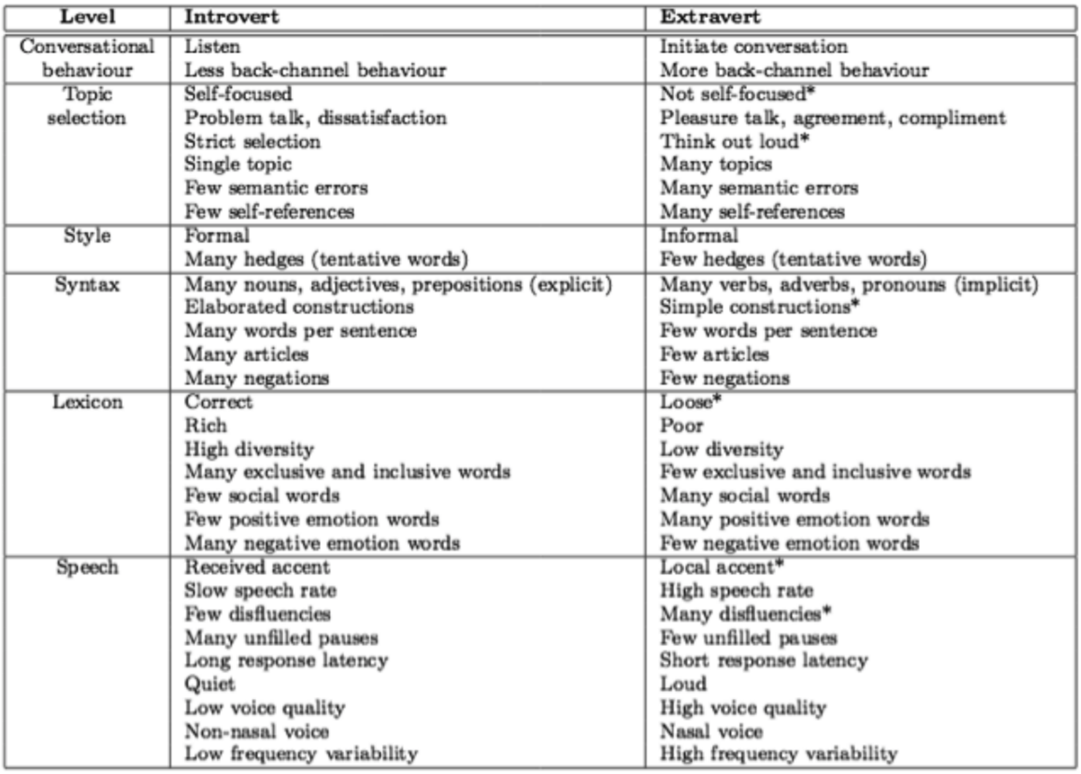
In 1990, Dr Cynthia Breazeal developed Kismet, a robot capable of recognising and simulating emotions at M.I.T. as an experiment for Affective Computing [4]. Dr Cynthia creates a mother-infant like an environment where the bot is an infant, and she is the care-taker. The bot gives different expression(like sad, happy, angry, disgusted) on different actions. This helps the care-taker understand how the bot wants to be treated and what way will it be able to learn and optimise itself better [5] This bot observes facial cues, the eyes, face, neck of the user to understand the affect.

In 2016, Supergirl Zara was proposed (Fung et al. 2016), an interactive virtual robot hosted on the server, that has a cartoon-like appearance and uses Machine Learning algorithms to learn and have empathy. Zara identifies linguistic cues from six different domains to classify the user into an introvert and extrovert (as shown in an image taken from (ibid.)), it also scans the face when the user starts the first session to learn the gender and ethnicity. This is the only paper from the research conducted so far that takes into consideration several external features apart from the utterances like speech rate, long response latency, accent, low voice quality for Affective Analysis. However, it isn't easy to find such external factors in textual conversations when building an affective empathetic system. Further research is

[4] https://en.wikipedia.org/wiki/Kismet$_{(}$*robot*)

[5] https://www.youtube.com/watch?v=Kw-gOmJwzuc

conducted to understand emotion detection in textual conversation.



Level	Introvert	Extravert
Conversational behaviour	Listen Less back-channel behaviour	Initiate conversation More back-channel behaviour
Topic selection	Self-focused Problem talk, dissatisfaction Strict selection Single topic Few semantic errors Few self-references	Not self-focused* Pleasure talk, agreement, compliment Think out loud* Many topics Many semantic errors Many self-references
Style	Formal Many hedges (tentative words)	Informal Few hedges (tentative words)
Syntax	Many nouns, adjectives, prepositions (explicit) Elaborated constructions Many words per sentence Many articles Many negations	Many verbs, adverbs, pronouns (implicit) Simple constructions* Few words per sentence Few articles Few negations
Lexicon	Correct Rich High diversity Many exclusive and inclusive words Few social words Few positive emotion words Many negative emotion words	Loose* Poor Low diversity Few exclusive and inclusive words Many social words Many positive emotion words Few negative emotion words
Speech	Received accent Slow speech rate Few disfluencies Many unfilled pauses Long response latency Quiet Low voice quality Non-nasal voice Low frequency variability	Local accent* High speech rate Many disfluencies* Few unfilled pauses Short response latency Loud High voice quality Nasal voice High frequency variability

Figure 2.6: Summary of all the features identified by Zara, the virtual robot

2.4.1 Affective Text-Based Conversational Agents

Xiaolace, a state-of-the-art affective chatbot by Microsoft, was launched in 2014 that was built with an intricate architecture comprising of different layers, one of which is called Conversation Engine Layer that deals Affect Analysis. This layer constitutes a dialogue manager, an empathy module, Core Chat and dialogue skills. The empathy module helps understand the context and the empathetic aspects of the conversation. Even though Xiaolce's has given a remarkable performance, Zhaojiang et al. (2019) criticises it in its paper, stating that Xiolace's architecture involves hundreds of components making it complicated and it requires an enormous amount of data to be labelled before training. CAiRE (ibid.) provides an alternate end-to-end system that uses the concept of Transfer Learning, a semi-supervised learning technique, to learn all components as a single model in an entirely data-driven manner mitigating the need for sharing labelled data across shared modules. CAiRE Fine-tunes Generative Pre-Trained Transformer (G.P.T.), a large-scale pre-trained language model with dialogue emotion classification.

Recently proposed HappyBot, (Shin et al. 2019) adds a look-ahead feature into the existing systems for more efficiency. The look-ahead module tries to predict what a human might feel about the response. This feature helps generate responses in a more empathetic and affective manner. The model is built using Reinforcement Learning, and it predicts the sentiment of the next user utterance using a fine-tuned pre-trained B.E.R.T. model to predict the sentiment of the next user utterance.

From both these breakthroughs, we understand that fine-tuning pre-trained models like G.P.T. and B.E.R.T. and incorporating it in our emotion engine helps get better results. G.P.T. and B.E.R.T. are discussed briefly in chapter 3. Both the proposed model uses previous utterances or next utterance to attain more context to analyse the Affect. The review so far has confirmed the theory that using the history of utterances increases efficiency in an empathetic chatbot and proffered Transfer Learning as a choice of technique. However, this research digs further deep into different affective approaches to find other alternatives and/or best models for emotion recognition.

2.4.2 Affect-Driven Text-Based system's approaches

Asghar et al. (2018) incorporates Affect Analysis into L.S.T.M. encoder-decoder model in three steps; 1) using word embeddings[6], 2)using affect-based objective functions to improve cross-entropy loss[7] and 3) use affective beam search for decoding[8]. This model uses linguistic cues to understand affect in a conversation. This research found that the model performs significantly better in emotion detection using an affective word embedding in seq2seq encoder-decoder model rather than a traditional embedding. Different benchmark Word Embeddings like Word2Vec and GloVe are discussed in brief in chapter 3.

(Colombo et al. 2019) proposed EMOTICONS, (Emoti)onal (Con)versation (S)ystem for affect analysis in textual conversation that generates responses using a continuous representation of emotions. This model recognises emotions at a word and sequence level using vector representation, affect regulariser, and an affect sampling method.

Both the approaches use neural networks to recognise emotion from a single text, however, to make a system more affective, it should take into consideration the context of the conversation that influences an emotion to better explain the underlying affect. Context is necessary for a chatbot to enable it to understand that even though the user says "I am fine", maybe the user isn't really fine.

[6] Word Embedding helps capture the context of a word in a document, semantic and syntactic similarity, relation with other words - https://towardsdatascience.com/introduction-to-word-embedding-and-word2vec-652d0c2060fa

[7] Cross-entropy can be used as a loss function when optimising classification models like logistic regression and artificial neural networks - https://machinelearningmastery.com/cross-entropy-for-machine-learning/

[8] Beam Search Decoding algorithm for decoding on text generation
https://machinelearningmastery.com/beam-search-decoder-natural-language-processing/

2.5 State-of-the-art Models for Emotion Detection in Conversation Using Multiple Utterances

This section of the review discusses context can be retrieved by interconnecting utterances and compares different state-of-the-art models used for emotion recognition using utterance history.

2.5.1 Bidirectional Long Short Term Memory (bc-LSTM)

In 2017, Poria, Cambria, et al. (2017) proposed Bidirectional L.S.T.M. network model that takes a sequence of utterances in a video as it's input and extracts contextual features by associating the dependencies among the input utterances. A Short-Term Long Memory(L.S.T.M.) is a kind of Recurrent Neural Network(R.N.N.) capable of long-range modelling dependencies, hence allowing the network to be able to consider the context from distant utterances for emotion detection. A Bidirectional - L.S.T.M. considers both utterances occurring before and after the utterance in the video, which enables the model to understand the surrounding context. Poria, Cambria, et al. (ibid.) also proposes a variant of the model called, Bidirectional-LSTM + Attention, where the attention is applied to lstm at each timestamp providing a better context for final utterance. Bc-LSTM was also the most common choice for neural architecture for EmoContext (**semevaltask3**). EmoContext was a task published by H.A.I. in 2019 [9] for detecting emotion in a conversation using the context of two turns of utterances.

2.5.2 Conversational Network Memory (CMN)

So far, bc-LSTM only considers surrounding utterances for context. **cmn** proposed Conversational Memory Network(C.M.N.) in 2018, that takes into account inter-speaker dependency relation for emotion classification. This is built upon deep neural framework to retrieve contextual information from conversation history. This network uses Gated Recurrent Units (G.R.U.s) to model previous utterances of the individual speaker into memories. These memories connected using attention-based hops to capture inter-speaker dependencies. This approach brings an interesting turn to the research of affect analysis by using speaker-specific context as well as inter-speaker dependencies that highly influence the affect. CMN performs better than Bc-LSTM for same dataset (**cmn**)

[9] https://www.humanizing-ai.com/emocontext.html

2.5.3 Interactive Conversational Memory Network (ICON)

Later the same year, D. Hazarika et al. (2018) proposed Interactive Conversational Memory Network(ICON) as an extension to C.M.N. with improved architecture. ICON uses an interactive scheme to model inter-speaker emotional dynamics with less training parameters. ICON outperformed many state-of-the-art models on multiple datasets. The model's ability to visualise attention also proves that utterances in conversational history provide essential emotional cues. However, both ICON and C.M.N. aren't scalable to multi-party conversations meaning when speakers in test data are more than speakers in training data set (Poria, Majumderd, et al. 2019)

2.5.4 DialogueRNN : Attentive Recurrent Neural Network

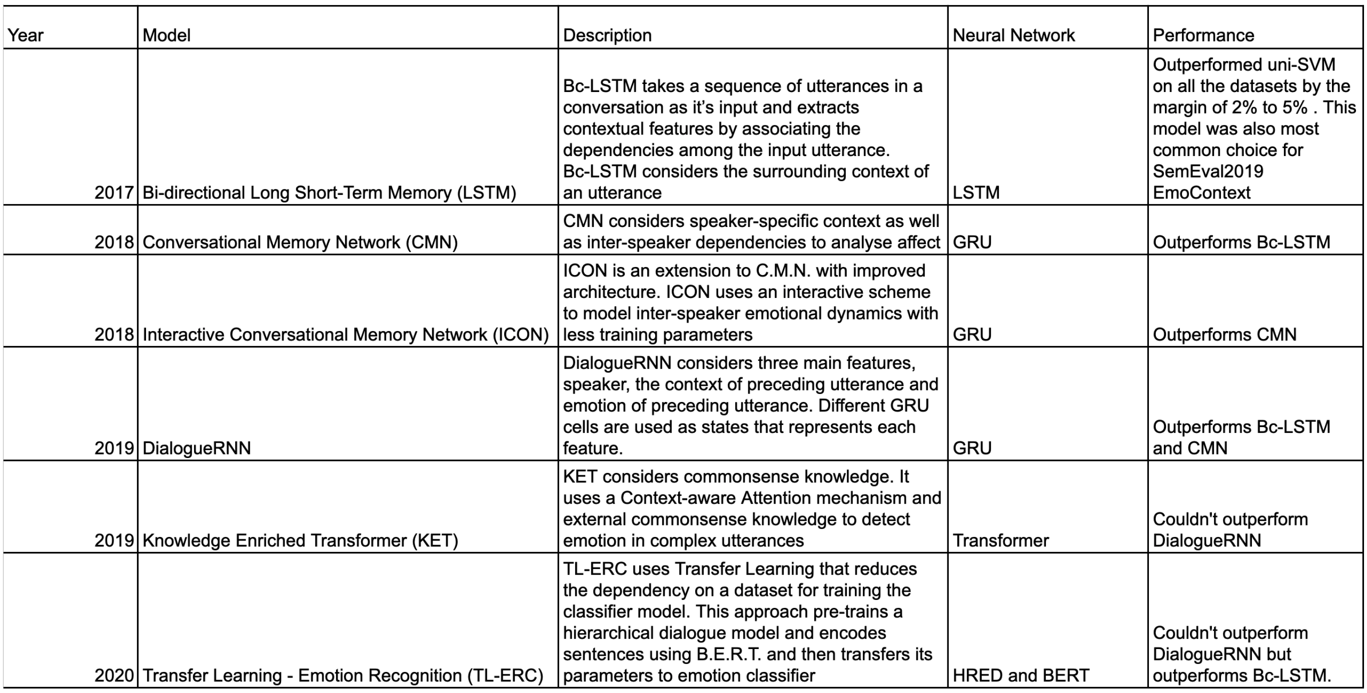
At the beginning of 2019,Majumder et al. (2019) proposed DialogueRNN, an attentive recurrent neural network with three G.R.U.s, where two of the G.R.U. are called global G.R.U. that receives the utterances, and the third G.R.U. is called party G.R.U. that updates context and party. Even though both, C.M.N. and ICON performed speaker-specific context, these models do not consider individual speaker information for emotion detection of final utterance. DialogueRNN considers three main features, speaker, the context of preceding utterance and emotion of preceding utterance. DialogueRNN maintains an individual speaker state for every speaker in a conversation. The speaker state depends solely on the context through attention and previous speaker state to ensure that at time t, a speaker state only receives information from the last state. This speaker state is fed to the emotion G.R.U. Cell which is used for emotion classification. DialogueRNN outperforms all the state-of-the models with a significant margin. Even though DialogueRNN considers emotion of preceding utterance, it performs poorly when there's an emotion-shift from the previous utterance.

2.5.5 Knowledge Enriched Transformer (KET)

Zhong et al. (2019) states that for emotion detection humans often rely on context as well as commonsense knowledge. Neither of the proposed models took into consideration commonsense knowledge. KET uses a Context-aware Attention mechanism and external commonsense knowledge to detect emotion in complex utterances. KET uses ConceptNet and NRC_VAD as knowledge bases in the model. ConceptNet is a multi-lingual semantic graph that describes human knowledge in natural language, and NRC_VAD is a list of English words with its V.A.D. scores that is valence, arousal and dominance. However, KET couldn't perform better than the state-of-the-art model DialogRNN for emotion detection in textual conversations.

2.5.6 Transfer Learning-Emotion Recognition Framework

Machine Learning methods work well when training and test datasets are from the same feature space and distribution; however, when the distribution changes, the statistical models needs to be trained from scratch. To reduce the efforts of re-training the model and using the knowledge retrieved from the previous training, Transfer Learning can be used (Pan and Yang 2010). Transfer Learning is a semi-supervised learning approach that stores and transfers knowledge collected from the training process. All the methods are seen so far dealt with retrieving context of conversation from utterance history, speaker-specific context, inter-speaker dependency, inter-personal influences, party-specific information. However, all these models were trained using manually annotated data. The emotion interpretation may differ from person to person. Devamanyu Hazarika et al. (2020) recently proposed emotion recognition in a conversation framework that uses Transfer Learning that reduces the dependency on a dataset for training the classifier model. This approach pre-trains a hierarchical dialogue model and encodes sentences using B.E.R.T. [10] and then transfers its parameters to emotion classifier. This model is expected to perform better in case of emotion shift.



Year	Model	Description	Neural Network	Performance
2017	Bi-directional Long Short-Term Memory (LSTM)	Bc-LSTM takes a sequence of utterances in a conversation as it's input and extracts contextual features by associating the dependencies among the input utterance. Bc-LSTM considers the surrounding context of an utterance	LSTM	Outperformed uni-SVM on all the datasets by the margin of 2% to 5% . This model was also most common choice for SemEval2019 EmoContext
2018	Conversational Memory Network (CMN)	CMN considers speaker-specific context as well as inter-speaker dependencies to analyse affect	GRU	Outperforms Bc-LSTM
2018	Interactive Conversational Memory Network (ICON)	ICON is an extension to C.M.N. with improved architecture. ICON uses an interactive scheme to model inter-speaker emotional dynamics with less training parameters	GRU	Outperforms CMN
2019	DialogueRNN	DialogueRNN considers three main features, speaker, the context of preceding utterance and emotion of preceding utterance. Different GRU cells are used as states that represents each feature.	GRU	Outperforms Bc-LSTM and CMN
2019	Knowledge Enriched Transformer (KET)	KET considers commonsense knowledge. It uses a Context-aware Attention mechanism and external commonsense knowledge to detect emotion in complex utterances	Transformer	Couldn't outperform DialogueRNN
2020	Transfer Learning - Emotion Recognition (TL-ERC)	TL-ERC uses Transfer Learning that reduces the dependency on a dataset for training the classifier model. This approach pre-trains a hierarchical dialogue model and encodes sentences using B.E.R.T. and then transfers its parameters to emotion classifier	HRED and BERT	Couldn't outperform DialogueRNN but outperforms Bc-LSTM.

Figure 2.7: Summary Table of all the state-of-the-art models for Emotion Recognition in Conversation.

[10] BERT : Bidirectional Encoder Representations from Transformers, is a neural network-based technique for natural language processing pre-training to give a better context of the sentence

2.6 EmoContext SemEval2019-Task3

SemEval2019 Task 3 called EmoContext (Chatterjee et al. 2019), proposed same hypothesis as this research that the context of an utterance can perform better Emotion Prediction. The task is to retrieve context from 3 previous utterances of a conversation and then perform emotion classification. Chatterjee et al. (ibid.) also indicated that Bc-LSTM was the most common choice and performed the best.

Smetanin (2019) supports the statement that Bc-LSTM is the best choice, as the paper proposes Emosense, an architecture based on Bc-LSTM. The model uses word embedding instead of manual text pre-processing and also captures user specific features. Agrawal and Suri (2019) introduces NELEC, that also uses Bc-LSTM based model with Attention. The model performs heavy pre-processing steps and achieves better results than Emosense. C. Huang et al. (2019) proposes ANA, also built upon Bc-LSTM claims to outperform state-of-the-art BERT model known for its performance in classification.

2.7 Conclusion

Emotion Recognition in conversation is gaining popularity in the field of N.L.P. Recent advancements prove that using utterance history can help obtain context, speaker-specific information and inter-speaker influences to analyse affect. Different models considers different features in a conversation for emotion prediction as seen in Fig. 2.7.

Every model tried to outperform the previous state-of-the-art models. However, KET and TL-ERC both failed to outperform DialogueRNN. TL-ERC framework is built upon the concept of Tranfer Learning, which is gaining major attention in the field of conversational A.I. Even though TL-ERC performed poor, it is worth comparing it's performance with other state-of-the-art models. TL-ERC reduces dependency on the dataset and is expected to perform better with less parameters. DialogueRNN is chosen as the second state-of-the-art model as it gave promising results against all other state-of-the-art methods. DialogueRNN is also the only model that considers speaker information. Bc-LSTM is chosen as the third state-of-the-art model as it's the only model that considers surrounding context and it was the most common choice for EmoContext. None of the work compares these three state-of-the-art model using a chat-based conversational dataset. This research will evaluate and compare these three models against a chat-based conversational dataset and a acted dataset.

For SemEval2019 Task 3, the best performing model from the three state-of-the-art model is chosen. This model is compared with the latest Transformer based model using EmoContext dataset. Emotion prediction is performed by both these models and the results are evaluated to find out, if the Transformer based model can outperform the state-of-the-art, Bc-LSTM.

3 Background Concepts

3.1 Chapter Overview

Emotion can be predicted using different neural network-based classification models. These models are trained using large corpora. NLP is used as a part of these neural network models to understand the data. The steps to develop an emotion prediction model are as follows:

1. Collect data to train the model
2. Preprocessing Data
3. Tokenisations to convert sentences into words
4. Word Embeddings to understand the context
5. Train the model with the dataset.

The Methodology explains in detail the architectures used building emotion prediction engines, the type of RNN it is built upon, the dataset used to train that network, how preprocessing was performed on the data, what Tokenizers and Word embeddings were used. However, before discussing that, it is essential to understand why there is a need for tokenisers and word embeddings or what is an RNN. This chapter is a prologue to Methodology; it is a preliminary introduction of the terms used in the research.

3.2 Natural Language Processing

Natural Language Processing is a field of computer science and linguistics that explores how computers can be used to understand and manipulate natural language text or speech (Liu et al. 2017). Human-Computer Interaction in the field of NLP usually comprises of two branches; Natural Language Understanding(NLU) and Natural Language Generation(NLU) (ibid.). NLU processes the input received from the human and translates it into a machine-understandable format. Natural Language Generation focuses on producing text in human language. As per Torfi et al. (2020), data becomes more meaningful through a deeper understanding of its context.

3.3 Word Embeddings

Word embeddings are used to capture the context of a word in a text document, understand the semantic and syntactic similarity of the words, the relation of one word with other, find dependencies of a word in a sentence Etc. (Karani 2018). Word Embeddings is a way to represent text in a high-dimensional space where a vector represents each word in the vocabulary. (Brownlee 2017). The vectors learn in such a way that similar words have similar representations.

To better understand why we need word embeddings (=**towardsmedium**), consider two sentences *You look good* and *You look great*. Let V be vocabulary of both these sentences such that, V = "You", "look", "good", "great". Now if this vocabulary is one-hot encoded to generate a vector, the vectors would be (transpose of) You = [1,0,0,0], look = [0,1,0,0], good = [0,0,1,0], great = [0,0,0,1]. According to this vector representation, each of the words belongs to an independent dimension creating a four-dimensional space. Neither word depend on each other. However, that is not true. Word Embeddings enables the words to occupy close spatial positions for words with similar context.

Two methods for learning vector representations are; 1) Using Embedded Layer, 2) Using Word2Vec or GloVe. Note that there are more methods. However, the below sections will focus on these two as they are used during the research.

3.3.1 Using Embedded Layer

A framework such as Keras provides an Embedding Layer that jointly learns with a neural network model used for NLP tasks like document classification or language modelling (ibid.). This method requires the data to be cleaned and processed such that each word can be one-hot encoded. The size of the vector space could be 50, 100 or 300 dimensions. The embedding layer is used at the beginning of a neural network model.

3.3.2 Word2Vec

A downside of learning a word embedding as part of the network is that it can be very slow, especially for enormous text datasets. Mikolov et al. (2013) at Google 2013 proposed Word2Vec, to make the training of embeddings more efficient. Word2Vec analysed learned vectors and performed vector math on represented words. In simple terms, this embedding captures the context in such a way that for an analogy "king is to queen as a man is to woman", if "man-ness" is subtracted from "king" and "women-ness" is added, it results into the word "queen" (Brownlee 2017) . Word2Vec, uses two different learning models; Continuous Bag-of-Words and Continuous Skip-Gram Model. The Continuous bag-of-words model learns the word embeddings by predicting the current word from context, and the Continuous skip-gram model learns from surrounding words of the current word.

3.3.3 GloVe

The Global Vectors for Word Representation is an extension to the Word2Vec method for efficiently learning word vectors (Pennington et al. 2014). GloVe is an approach that combines both Latent Semantic Analysis technique and local context-based learning like Word2Ve (Brownlee 2017) Latent Semantic Analysis creates word representation using global statistics. GloVe construct a word co-occurrence matrix using statistics across the whole text corpus; this results in better word embeddings.

3.4 Tokenization

Tokenisation is diving a sentence into chunks of words where each word is referred to as a *token*. Usually, tokenisation provides two dictionaries, word-> index and index->word.

3.5 Recurrent Neural Networks

In this research, different architectures like bc-LSTM, DialogueRNN and TL-ERC are used. Each architecture trains a type of Recurrent Neural Network for emotion classification. The section below explains Recurrent Neural Networks.

A Recurrent Neural Network is used for two primary purposes; scoring arbitrary sentences with measures of grammatical and semantic correctness for NLP based tasks and generating new texts. These tasks are not possible with traditional neural networks as they consider input and outputs as independent entities. However, to predict the next word in a sentence, the network needs to remember which words came before it. Another way to describe RNN is that it is a network that has memory to remember information that has been calculated so far. RNN uses sequential information. The term "recurrent" is added because the model performs the same task for every element of the sequence. For example, for a sentence of 6 words, the network unrolls into a 6-layer neural network, where each layer represents each word.

x_t = input at time step t. x_t is a vector corresponding to second word of the section

s_t = hidden state at time step t or in short the. "memory" of the network.
$s_t = f(Ux_t + Ws_{t_1})$. The function f could be any activation function like ReLU (described

in section 3.10.1). o_t = output at step t. To predict next word in a sectence, $o_t = softmax(V_{st})$ (softmax function is described in section 3.10.2)

3.5.1 Rectified Linear Units(ReLU)

ReLU (**http://www.wildml.com/deep-learning-glossary/##relu**) are activation function that could be defined as : $f(x) = max(\theta, x)$, here x is the input to neural network. Activation functions allow neural networks to learn complex decision boundaries. Rectifiers allow better training of deeper networks (Glorot et al. 2010). ReLU is sparse and suffers less from vanishing gradient problems.

3.5.2 Softmax Layer

A softmax function is often used as the last activation function of a Network Network that converts vectors of raw scores into class probabilities for classification. The softmax function takes in a vector of K real numbers as input and normalises it into a probability distribution with K probabilities that is proportional to the exponentials of the input numbers. A softmax function σ is defined as :

$\sigma(z)_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}$ for i = 1,2,...K.

3.5.3 Cross-Entropy Loss

The model tries to find the best parameters that minimise error when training data. This error is measured using a Loss function like Cross-Entropy Loss. Cross-Entropy Loss can be defined as :

$L(y, y') = -\frac{1}{N} \sum_{n \in N} \sum_{i \in C} y_{n,i} logy'_{n,i}$

3.5.4 How Recurrent Neural Networks work ?

A traditional feed-forward neural network has three layers, input layer, hidden layer and output layer (Fig. 3.1). To make this feed-forward neural network to be able to remember the previous information, a loop is added. This loop allows passing on prior information forward (Fig. 3.1). To understand the application of RNN in NLP, let's consider a 3-words statement "how are you" fed to a neural network for intent classification. As RNN works sequentially, it will consider one word at a time. RNN takes in the first word "How" as input, encodes it and produces an output. For the second word, RNN considers the second word itself and the hidden state from the previous word. The process will be repeated for the word "you". The output generated from last RNN is passed to next feed-forward layer for classification. See Fig. 3.3 to understand the flow explained above.

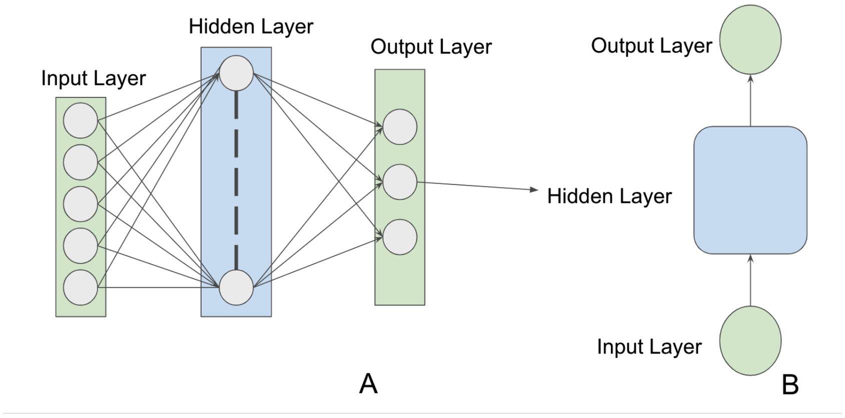


Figure 3.1: Feed-forward Neural Network A) Representation of Neural Network B) Narrowed down representation of Neural Network

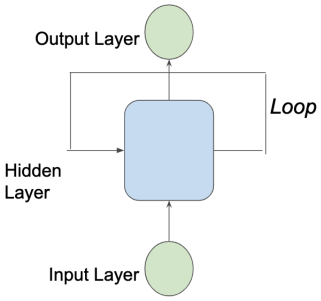


Figure 3.2: Recurrent Neural Network

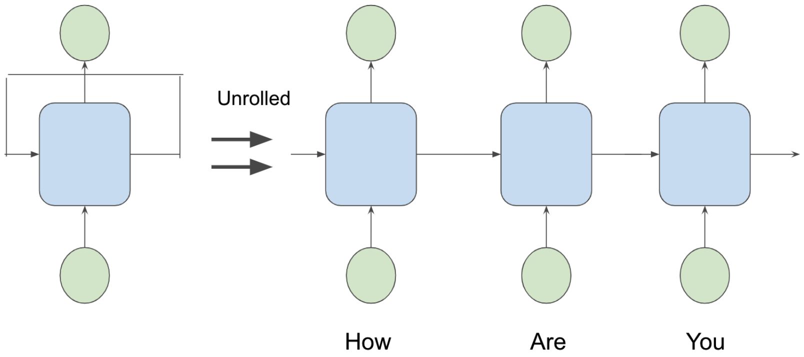


Figure 3.3: Unfolding Recurrent Neural Network for a sentence of 3 words

3.5.5 Drawback of Recurrent Neural Network

RNN has what is called the "Short-Term Memory" due to which it suffers from the vanishing gradient problem. A vanishing gradient means RNN would have trouble retaining information from previous states as it progresses further. Now, why would that matter? Let's say for a short sentence like "how are you", the RNN can extract the context as it can remember states for all three words. However, for a longer sentence like, "Would you maybe wanna have a cup of coffee , let's say at a (cafe) ?", RNN need context to understand the intent, i.e., it needs to remember "a cup of coffee" to be able to associate it with "cafe". However, RNN cannot learn long-range dependencies due to which. To mitigate this issue, two extensions of RNN were created, Long Short-Term Memory(LSTM) and Gated Recurrent Unit(GRU)

3.6 Long Short-Term Memory

LSTM were proposed in 1997 in Hochreiter and Schmidhuber (1997). LSTM combats the vanishing gradient problem using a gating mechanism. The LSTM model has three gates i, f, o called input gate, forget gate and output gate respectively. The input gate defines the measure for newly computed state for current input that should be allowed to pass through. The forget gate then defines how much of previous state must be allowed to pass through and finally the ouput gate defines how much of the internal state must be exposed to external network. g is present in hidden state that is computed based on current input and previous hidden state. However, in LSTM unlike RNN,solely g isn't considered as new hidden state, input gate is also used. c_t acts as an internal memory. The internal memory is the combination of previous memory multiplied with forget gate and new hidden state g multiplied with input gate i. The whole structure of LSTM is similar to that of RNN however it differs in how the hidden state is calculated. Hidden state s_t is computed as :

$i = \sigma(x_t U^i + s_{t-1} W^i)$

$f = \sigma(x_t U^f + s_{t-1} W^f)$

$o = \sigma(x_t U^o + s_{t-1} W^o)$

$g = tanh(x_t U^g + s_{t-1} W^g)$

$c_t = c_{t-1} * f + g * i$

$s_t = tanh(c_t) * o$

The gating mechanism allows LSTM to consider long-term dependencies the network learns from the parameters, how the memory should behave.

3.7 Gated Recurrent Unit (GRU)

A Gated Recurrent Unit(GRU), built upon the same idea of using gating mechanism, was proposed recently in 2014, with fewer parameters and less number of gates compared to

LSTM. A GRU has two gates, r, z called the reset gate and update gate respectively. The reset gate defines how to combine new input and previous memory. The updated gate defines the limitation for storing the previous memory. The hidden state in GRU is computed as:

$$z = \sigma(x_t U^z + s_{t-1} W^z)$$

$$r = \sigma(x_t U^r + s_{t-1} W^r)$$

$$h = tanh(x_t U^h + (s_{t-1} * r) W^h)$$

$$s_t = (1 - z) * h + z * s_{t-1}$$

3.7.1 Drawbacks of traditional LSTM and GRU

Even though LSTM and GRU were built to solve the issue of long-range dependencies, the vanishing gradient problem persists as the model progresses in longer sentences. Another issue with LSTMs like structure is that it becomes difficult to parallelise the work for processing the sentences; the words are always trained one by one.

3.8 Attention

The attention mechanism is built upon the idea the issues from LSTM could be solved if special attention was paid only on specific words. In Attention mechanism, the model focusses only on the new segment of information at every time step. This means instead of encoding the whole sentence and then passing it as output with the final hidden state, the RNN with Attention, would encode each word and pass on the hidden state of each word to the output. This ensures that relevant information from each and every word is extracted. This helps the issue with remembering distant words; however, the issue of processing words in a parallel manner is still not possible.

3.9 Convolutional Neural Network

When performing tasks like feature extraction on an extensive corpus, the traditional models would take much time due to its inability to process in a parallel manner. A Convolutional Neural Network (CNN) can process multiple words at the same time and does not have to depend on previous words for translation.

3.10 Transformers

Transformers are the newest model that uses CNN and Attention both, solving the problem of parallelisation and boosting the speed of training (Vaswani et al. 2017). A Transformer has an architecture similar to RNN. However, it comprises of six encoders and six decoders. Each encoder consists of two layers, Self-Attention and Feed-Forward Neural Network. The decoder has three layers; Self-Attention, Encoder-Decoder Attention and Feed-Forward Neural Network.

3.11 Transfer Learning

Neural Networks were built keeping in mind the human brain; it learns the same way as a human does. However, humans do not always learn from the ground up. If a person knows to know how to ride a bicycle, a person can learn to ride a motorbike using the knowledge of riding a bicycle. Transfer Learning (Zhuang et al. 2019) utilises knowledge acquired for one task to solve related ones. Andrew NG, renowned data scientist, when providing a tutorial in NIPS 2016 (**slides**) stated - "After supervised learning — Transfer Learning will be the next driver of ML commercial success". With Transfer Learning, one can transfer knowledge like features, weights from one trained model to new models which solves the issues of lack of datasets needed to train the model.

3.12 BERT

One of the biggest challenges in A.I. and N.L.P is the lack of training data. In order for a model to perform well, it needs to be trained on a large dataset. Researchers have developed different techniques to train language representation models using unannotated text called pre-training. These pre-trained models can be fine-tuned for smaller tasks. BERT is one such technique for pre-training. Google AI researchers proposed Bidirectional Encoder Representations from Transformers (BERT) (Devlin et al. 2018) to pretrain deep bidirectional representations from unlabeled text with respect to the context in all layers. A BERT model can be finetuned with just one additional output layer to perform different tasks such as next sentence prediction, question-answering tasks etc.

3.13 Conclusion

This chapter discussed the key terminologies required to understand the research conducted further.

4 Methodology : Experiment 1

4.1 Chapter Overview

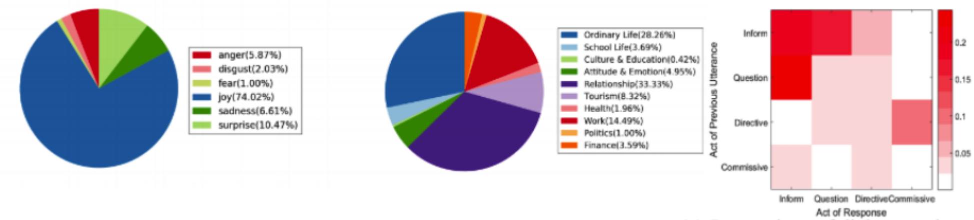
The research conducted in this thesis is divided into two parts. Experiment 1 is comparing state-of-the-art models to find out the best model for recognising emotions using utterance history. Experiment 2 uses that model as well as Transformer based model to predict emotions for EmoContext dataset to find out the best approach for emotion classification in conversation. As experiment 2 depends on the results of experiment 1, The Methodology and results are divided into two parts. This chapter discusses, in brief, the requirements and design of the state-of-the-art models, Bc-LSTM, DialogueRNN and TL-ERC.

4.2 Datasets used for training the three conversational A.I. models

For the neural networks to be able to classify emotions in a conversation correctly, it needs to be trained on an emotion-rich conversational corpus. For training the neural networks, two benchmark datasets are chosen; DailyDialog and IEMOCAP. DailyDialog is a large corpus of daily conversations covering different topics that comes up in day-to-day interactions. IEMOCAP consists of text transcripts of 12 hours long acted video, scripted and made to generate an emotion-rich dataset.

4.2.1 DailyDialog

DailyDialog (Li et al. 2017) is a well-appreciated dialog corpus for emotion recognition and dialog generation in conversational A.I. (Cai et al. 2020; Devamanyu Hazarika et al. 2020; Qin et al. 2020). It is a manually annotated emotion-rich multi-turn dialogue corpus that reflects daily conversations covering up to 10 topics. Dailydialog also conforms to two conversation patterns occurring in conversations, *Question-Answer* and *Directive-Commisive*. The dataset is labelled with an emotion and a class per utterance. The conversation is then labelled with a topic. Each utterance belongs to one of the four classes, *Inform, Question, Directive, Commissive*. The dataset incorporates six emotions *joy, sadness, surprise, anger, fear, disgust*. Li et al. (2017) presents a statistical report of the DailyDialog corpus to better understand how the data is distributed over different categories, class and emotions, see Fig. 4.1. Zandie (2020) presents a snippet of the conversation after preprocessing the dataset. Fig. 4.2.



(a) Emotion distributions in DailyDialog. (b) Topic distributions in DailyDialog. (c) Interactions of dialog acts in each utterance pairs.

Figure 4.1: Statistical report of DailyDialog dataset distribution based on (a) Emotion, (b) Topic and (C) Class

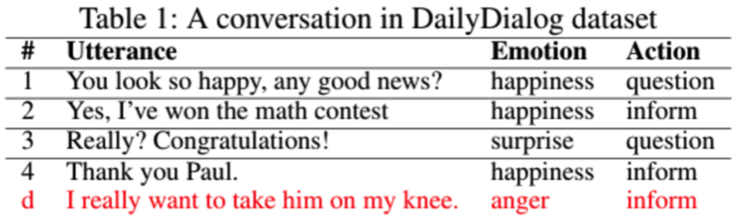
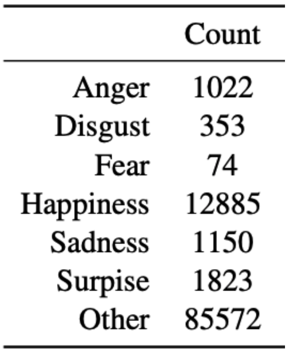


Table 1: A conversation in DailyDialog dataset

#	Utterance	Emotion	Action
1	You look so happy, any good news?	happiness	question
2	Yes, I've won the math contest	happiness	inform
3	Really? Congratulations!	surprise	question
4	Thank you Paul.	happiness	inform
d	I really want to take him on my knee.	anger	inform

Figure 4.2: An example of a conversation format from DailyDialog dataset with Emotion and Action tag

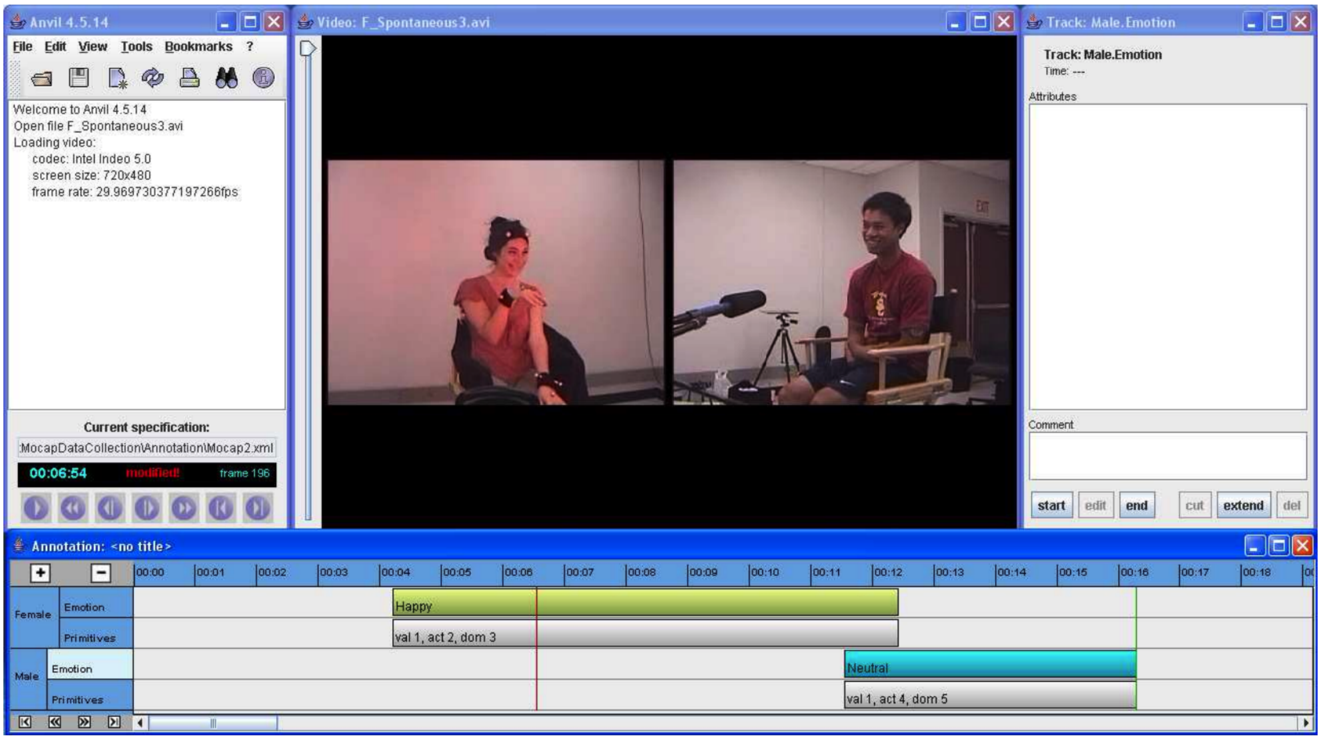


	Count
Anger	1022
Disgust	353
Fear	74
Happiness	12885
Sadness	1150
Surpise	1823
Other	85572

Figure 4.3: The list of count of Utterances for Each emotion in DailyDialog. DailyDialog is a large corpus with 102,879 records. Out of the total, 83.17% of data is categorised as other.

4.2.2 IEMOCAP

The Interactive Emotional Dyadic Motion Capture (IEMOCAP) is an acted, multimodal and multispeaker database (S.A.I.L 2004) by Speech Analysis & Interpretation Laboratory at University of Southern California [1]. IEMOCAP is a dataset consisting of 12 hour long audiovisuals, video, speech, motion capture of the face and textual transcripts. The interactions are dyadic, performed between two actors. There are in all 5 acts performed by ten actors, five female and five male. The act is either scripted or improvised in a way that portrays emotion concerning the context of the act. The dataset labelling is evaluated using ANVIL software tool[2] and different human assessors. Each utterance is labelled with an emotion label as well as a dimension label like Dominance, Valence, Arousal (Samarth Tripathi and H. S. M. Beigi 2018). Samarth Tripathi and H. S. M. Beigi (ibid.) presents how the software is used to evaluate the emotions; See Fig. 4.4.



Welcome to Anvil 4.5.14
Open file F_Spontaneous3.avi
Loading video:
codec: Intel Indeo 5.0
screen size: 720x480
frame rate: 29.969730377197266fps

Current specification: :MocapDataCollection\Annotation\Mocap2.xml

00:06:54 modified! frame 196

Track: Male.Emotion
Time: ---
Attributes
Comment

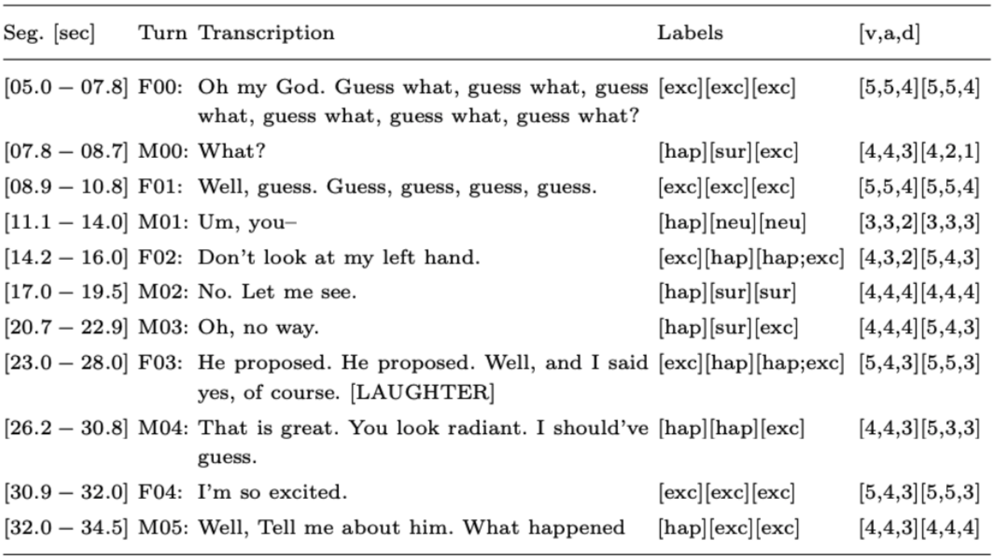
Female Emotion: Happy
Female Primitives: val 1, act 2, dom 3
Male Emotion: Neutral
Male Primitives: val 1, act 4, dom 5

Figure 4.4: Emotion Evalutation using AVIL software tool

This dataset also takes into account the context when labelling the emotion. Each session is manually segmented into utterances. An utterance could be more than one sentence in IEMOCAP. Fig. 4.5 describes how the textual transcripts are annotated with emotion and dimensional label.

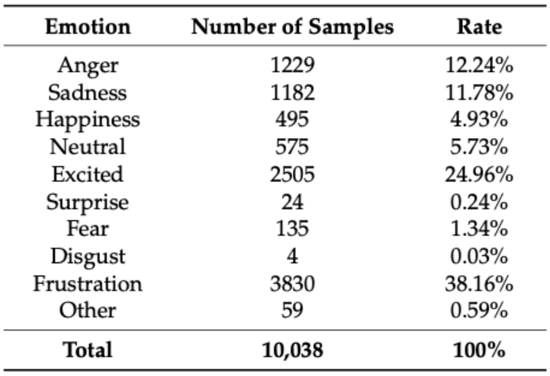
[1] https://sail.usc.edu/iemocap/iemocap¡*nfo.htm*

[2] https://www.anvil-software.org/



Seg. [sec]	Turn	Transcription	Labels	[v,a,d]
[05.0 – 07.8]	F00:	Oh my God. Guess what, guess what, guess what, guess what, guess what, guess what?	[exc][exc][exc]	[5,5,4][5,5,4]
[07.8 – 08.7]	M00:	What?	[hap][sur][exc]	[4,4,3][4,2,1]
[08.9 – 10.8]	F01:	Well, guess. Guess, guess, guess, guess.	[exc][exc][exc]	[5,5,4][5,5,4]
[11.1 – 14.0]	M01:	Um, you–	[hap][neu][neu]	[3,3,2][3,3,3]
[14.2 – 16.0]	F02:	Don't look at my left hand.	[exc][hap][hap;exc]	[4,3,2][5,4,3]
[17.0 – 19.5]	M02:	No. Let me see.	[hap][sur][sur]	[4,4,4][4,4,4]
[20.7 – 22.9]	M03:	Oh, no way.	[hap][sur][exc]	[4,4,4][5,4,3]
[23.0 – 28.0]	F03:	He proposed. He proposed. Well, and I said yes, of course. [LAUGHTER]	[exc][hap][hap;exc]	[5,4,3][5,5,3]
[26.2 – 30.8]	M04:	That is great. You look radiant. I should've guess.	[hap][hap][exc]	[4,4,3][5,3,3]
[30.9 – 32.0]	F04:	I'm so excited.	[exc][exc][exc]	[5,4,3][5,5,3]
[32.0 – 34.5]	M05:	Well, Tell me about him. What happened	[hap][exc][exc]	[4,4,3][4,4,4]

Figure 4.5: A part of a conversation in IEMOCAP with utterances with its emotion labels and VAD values. The data is a text transcript of a video. The first column is the time of the video when utterance was spoken. Turn is the Actor who spoke it; M stands for male and F for female. Transcription has utterances. Labels have emotions where exc is excited, hap is happy, sur is surprised, neu is neutral.



Emotion	Number of Samples	Rate
Anger	1229	12.24%
Sadness	1182	11.78%
Happiness	495	4.93%
Neutral	575	5.73%
Excited	2505	24.96%
Surprise	24	0.24%
Fear	135	1.34%
Disgust	4	0.03%
Frustration	3830	38.16%
Other	59	0.59%
Total	**10,038**	**100%**

Figure 4.6: IEMOCAP Dataset distribution based on Emotion. The total number of records are 10,038 with 38.16% of data associated with Frustration tag. Disgust, Fear and Surprise have the least number of records associated with it.

4.3 Datasets used for training the pre-training dialogue model for Transfer Learning

One of the three state-of-the-art models, TL-ERC uses semi-supervised Transfer Learning approach. Transfer-Learning based models have a source model and a target model. The source model is trained, and the knowledge learned during training is transferred to the target model. Both the source and target are of the same domain; however, they are trained to perform different tasks. In TL-ERC, the source is used as a dialogue generator for conversational model and target is used as an emotion classifier (Devamanyu Hazarika et al. 2020). As discussed in the previous section, the target is trained using emotion-rich benchmark datasets known for next sentence prediction, i.e., Ubuntu and Cornell.

4.3.1 Ubuntu Chat Logs

Ubuntu Dialogue Corpus(Lowe et al. 2015) consists of almost one million two-person conversations extracted from the Ubuntu chat logs retrieved from the history of customer service on the platform. It has over 7 million utterances and 100 million words.

4.3.2 Cornell Movie Dialog Corpus

Cornell(Danescu-Niculescu-Mizil and L. Lee 2011a) is a large dataset of conversations extracted from movies scripts. Cornell has around 220,570 conversational exchange between 10,292 pairs of movie characters. It has in total 304,713 utterances (Danescu-Niculescu-Mizil and L. Lee 2011b)

4.4 Tokenizer, Word Embeddings and Embedding Matrix

Tokenization : Before training the data, the text data is tokenised. In an embedding matrix, the input required should be integer encoded. For tokenisation, each model experiments with three different tokenisers. Bc-LSTM performs tokenisation with Pytorch tokeniser and word embeddings, DialogRNN with Keras tokeniser and TL-ERC uses a Pytorch pretrained BERT for Token representations with Spacy for Text Categorizer, this is called the Spacy Pipeline, mainly used in Transfer Learning (Honnibal 2019). *Why is the Spacy pipeline required though?* Honnibal (ibid.) in their article, Spacy meets Transformers, mentions that "The models are too large to serve in production, but they can be used to supervise a smaller production model" [3].The Spacy pipeline enables the model to be loaded with a large generic model like pretrained BERT with lots of text and then it trains on smaller dataset using Spacy with labels specific to the problem.

Word Embeddings : For Word Embeddings, Global Vectors for Word Representation(GloVe) is used for all three models. It provides a vector space with meaningful substructure of 75% on a recent word analogy task (Pennington et al. 2014). Following GLoVe embedding is used in the models: Common Crawl (840B tokens, 2.2M vocab, cased, 300d vectors, 2.03 GB download) [4]

[3] https://explosion.ai/blog/spacy-transformers

[4] https://nlp.stanford.edu/projects/glove/

Embedding Matrix : Keras has a tf.keras.layers.Embedding layer, which makes it easy to use word embeddings (Holt 2020). This layer maps integer indices to dense vectors. Pre-trained models are used during training. Once trained, the learned word embeddings can encode similarities between words.

4.5 Model Architectures

This section will briefly discuss the architecture and design of each of the three state-of-the-art model chosen for evaluation.

4.5.1 Bc-LSTM

Proposed variant bc-LSTM (Poria, Cambria, et al. 2017) takes a sequence of utterances $u_1, u_2....u_n$ from a conversation as an input where L_n is the number of utterances in the conversation. The model extracts contextual unimodal and multimodal features by modelling the dependencies from the utterances. The whole process is conducted in two steps: 1) Context-Independent Unimodal Utterance-Level Feature Extraction and 2) Contextual Unimodal and Multimodal Classification.

1. Context-Independent Unimodal Utterance-Level Feature Extraction First, the unimodal features are extracted from input utterances without considering the context or dependencies. The original framework by Poria, Cambria, et al. (ibid.) performs feature extraction separately for textual, audio and video data. However, the adapted model used for this methodology performs only textual feature extraction as it's designed for text-based conversations.

Text-Feature Extraction : Each utterance is fed to a Convolutional Neural Network (CNN) for Feature-Extraction. A CNN is a class of deep neural networks, composed of two essential parts; feature extraction and classification. Feature extraction includes several convolution layers followed by max-pooling and an activation function (Khoshdeli et al. 2017). Here, each utterance(up to 50 words) is represented as an amalgamation of the vectors of words in the utterance. The words vectors are obtained by using Word Embeddings.

The convolution kernels are applied to these combined word vectors. The CNN here consists of two convolutional layers; first with two kernels of size 3 and 4 with 50 feature maps each and second has a single kernel of size 2 with 100 feature maps. The layers are alternated with a max-pooling layer which is followed by a fully connected layer of size 500. Finally, the network is ended with a softmax output layer. The CNN learns abstract representations of the phrases and retrieves implicit semantic information of the utterances.

2. Context-Dependent Unimodal and Multimodal Utterance-Level Feature Extraction and Classification

For contextual feature extraction, an LSTM based network is used. LSTM is chosen as it allows the model to consider inter-dependencies from previous utterances when classifying the target utterance. LSTM can derive inter-dependencies because LSTM cells are capable of modelling long-range dependencies. The network can extract context from surrounding utterances and use it when classifying the target utterance.

Training : The model is trained using Cross-entropy Loss on every utterance's softmax output as per (4.3.1.1.2).

$$loss = -(1/(\sum_{i=1}^{M} L_i)) \sum_{i=1}^{M} \sum_{j=1}^{L_i} \sum_{c=1}^{C} y_{i,c}^{j} log_2(Y_{i,c}^{j}) \qquad (4.3.1.1.2)$$

To avoid overfitting, a dropout is interleaved between the LSTM cell and dense layer. As the number of utterances in a conversation is different, padding is applied to neutralise the range.

Hyperparameter tuning is performed by splitting the training dataset into training and validation sets with 80/20 split ratio. The model was trained for 200 epochs on batch size of 32 without Attention. Hyperparameters used for training the model are : attention=False, batch_size=32, class_weight=False, cnn_dropout=0.5, cnn_filters=50, cnn_output_size=100, dropout=0.25, epochs=200, l2=1e-05, lr=0.001, no_cuda=False, tensorboard=False

4.5.2 DialogueRNN

Majumder et al. (2019) builds DialogueRNN on the assumption that the emotion classification depends on three factors; 1) Speaker, 2) Context of the preceding utterances 3) Emotion behind the preceding utterances.

The model is implemented using GRU cells, each tracking a specific state to handle the above-given factors. The different states are Party, Global, Speaker and Emotion state. The Party-state is updated every time a party delivers an utterance; this tracks the individual party's emotional dynamic. Then, the Global state encodes the party-state and the context of preceding utterance together for context representation. Finally, the Emotions state updates based on party state and global state to represent an emotion. The emotion GRU sends previous utterance states and current speaker states to a softmax layer. Every GRU cell computes a hidden state h_t where for every GRU, $h_t = GRU * (h_{t-1} x_t)$. Here, h_{t-1} is the previous GRU state and x_t is input. The model design comprises of Text-Feature Extraction and functioning of GRU cells for Emotion classification.

Text-Feature Extraction : The model uses CNN for textual feature extraction, which obtains n-gram features for each utterance using three convolution filters of sizes 3,4 and 5. Every filter has 50 feature-maps. The output is then sent for max-pooling followed by ReLU activation. The activations are then fed to 100-dimensional dense layer for textual utterance representation. This network is trained to provide emotion labels at the utterance level.

Functions of GRU for emotion classification :

1. **Party GRU** : The Party GRU uses a fixed size of vectors to keep track of individual speaker states. The state here is updated based on the role of participant, either speaker or listener for the current utterance.

2. **Global GRU** : Global GRU captures the context of given utterance by encoding current utterance and speaker state. Here every state is speaker-specific representation. The context also factors inter-speaker and inter-utterance dependencies.

3. **Speaker GRU** : Speaker GRU captures relevant context from utterances as the speaker generates response based on preceding utterance. This cell encodes context from current utterance as well as the context received from the Global GRU. The combination of context from current and previous utterance is for better emotion classification.

4. **Emotion GRU** : Here, the emotion of the current utterance is represented by the Speaker state and the emotion representation of the previous utterance. As the context is essential for emotion classification, party states are also considered, which establishes a connection between speaker state and other party states.

Training: The model has two perceptron Layer with final Softmax Layer to calculate emotion class probability from the class of 6 emotions, and the most likely class is chosen as the emotion representation for the utterance. Emotion classifier has softmax layer + ReLU activation. The model uses categorical cross-entropy with L2 regularisation. The network is trained using Stochastic based Adam Optimizer. The **hyperparameters** are chosen using Grid Search.

4.5.3 TL-ERC

Devamanyu Hazarika et al. (2020) divides The Transfer Learning - Emotion Recognition Framework's implementation in two processes. *source* for generative conversation modelling and the *target* for emotion classification. The working of TL-ERC framework is described below.

1. **Source For Generative Conversation Modelling** The source uses a Hierarchical Recurrent Encoder-Decoder (HRED) model, which is a framework for seq2seq model used to generate dialogue responses. It generates responses using three sequential components;

Sentence Encoding using encoder recurrent neural networks (RNNs), Context Encoding and Generating responses. When the model receives a conversation with sentences, $x_1, x_2....x_t$, it goes through the following steps:

1. Sentence Encoding The encoder RNN encodes the context in each sentence such that

$$h_t^{enc} = f_\theta^{enc}(X_t, h_{t-1}^{enc}) \tag{4.5.3.1.1}$$

2. Context Encoding After that, each encoded sentence is fed to context RNN that models conversation until time t such that

$$h_t^{cxt} = f_\theta^{cxt}(h_t^{enc}, h_{t-1}^{cxt}) \tag{4.5.3.1.2}$$

3. Response Generation : Finally the decoder RNN generates sentence x_{t+1}

$$p_\theta(x_{t+1}|x_{\leq t}) = f_\theta^{dec}(x|h_t^{cxt}) \tag{4.5.3.1.3}$$

Then, HRED trains all the conversation in data using maximum likelihood estimation objective . In Equation (4.5.3.1.1) f_θ^{enc} represents encoder RNN. In this model, a Bidirectional Gated Recurrent Unit, (GRU) is used as a RNN function with parameter θ_{enc}^{source} to encode the sentences. For both context RNN (f_θ^{cxt}) in Equation (4.5.3.1.2) and decoder RNN in Equation (4.5.3.1.3), a unidirectional GRU is used with parameter θ_{cxt}^{source} and θ_{cxt}^{dec} respectively. The decoder finalises the process by generating a response using beam decoding.

2. Target For Emotion Detection in Conversation Similar to Source, the target receives an input conversation with utterances $x_1, x_2...x_n$. Target performs the same steps as source except for the response generation step, instead, the target performs emotion classification.

1. Sentence Encoding : Here, instead of HRED(Hierarchical Dialogue Model), a pre-trained state-of-the-art model BERT is used for sentence encoding as it outperforms HRED (Devamanyu Hazarika et al. 2020). BERT is appreciated for next-sentence prediction. This model only uses four transformer layers out of the 12 present in BERT.

2. Context Encoding : The model uses the same context RNN HRED model as source as it allows transferring the learned parameters θ_{cxt}^{source}

3. Emotion Classification : For every turn outputted from context RNN, an emotion is predicted using emotion classifier for each utterance in turn. The model is trained using Cross-Entropy loss.

Hyperparameters

For every target dataset, grid-search is performed to get the best parameters. The parameter selection is for learning rate from le-3,le-4 and le-5 and optimiser from Adam and RMSprop.

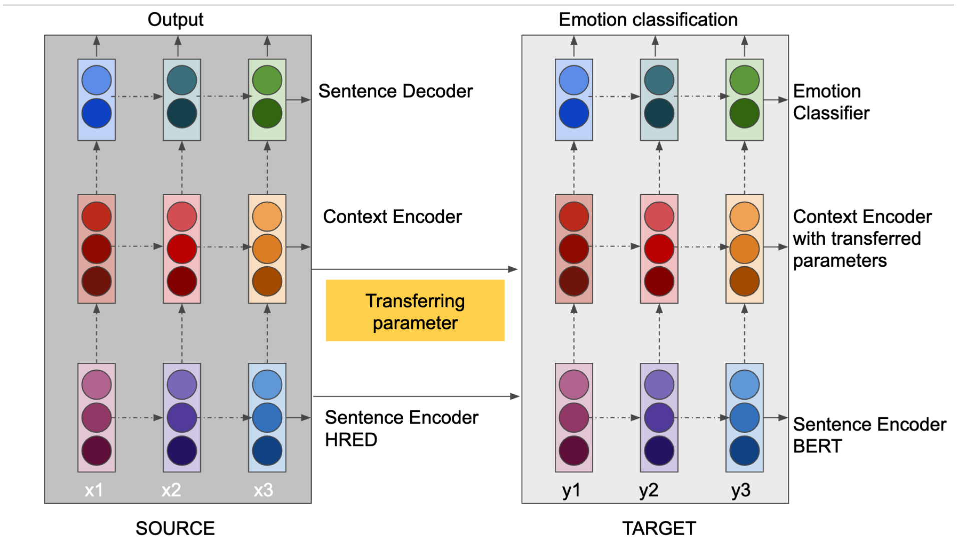


Figure 4.7: Architecture representation of TL-ERC: Left side block represents *Source* that transfers parameters θ_{enc}^{source} from sentence encoder and θ_{cxt}^{source} from context encoder, right block represents *Target*, that uses the transferred parameters and performs emotion classification

4.6 Evaluation Metrics

To evaluate the performance of the models, Average F1 and Micro average F1 scores are used in both experiments. F1 score works better than Accuracy when the class distribution is imbalanced. In all the datasets used in both experiments, the datasets are highly imbalanced. Micro-average F1 score is used to judge the performance of the models in both experiments, 1 and 2. (Huilgol 2019)

F1 score is a harmonic mean of Precision and Recall, and it gives better measures of incorrectly classified data.

$$F1 = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{4.6.1}$$

Precision is calculated by measuring the correctly identified positive cases from all the predicted positive cases.

$$Precision = \frac{TruePositive}{(TruePositive + FalsePositive)} \tag{4.6.2}$$

Recall is calculated measuring the correctly identified positive cases from all the actual positive cases.

$$Recall = \frac{TruePositive}{(TruePositive + FalseNegative)} \tag{4.6.3}$$

Micro F1-score is used to assess the quality of multi-label binary problems. It measures the average F1 score of all classes. (**microaf1**)

$$Micro - F1 = 2 * \frac{Micro - Precision * Micro - Recall}{Micro - Precision + Micro - Recall} \tag{4.6.4}$$

4.7 Conclusion

This chapter discussed four different datasets used for training the models, followed by a brief description of the design architecture and training pattern used for each model. The next chapter will reveal the results of this experiment.

5 Results: Experiment 1

5.1 Chapter Overview

This chapter discloses the results of experiment 1, also discusses in brief, the characteristics of the datasets and different results found during this experiment. The findings from both the experiments are discussed in detail, later in chapter 8.

5.2 Characteristics of Datasets

One of the models, TL-ERC, trains the source model and then transfers the knowledge gathered from the learning process to the target model. The source model is trained using datasets that would allow next sentence prediction. Then all the three models are trained using two large conversational corpora; DailyDialog and IEMOCAP. Table 5.1 and Table 5.2 are summary of datasets used chosen for the experiments.

5.2.1 Ubuntu vs Cornell

Table 5.1 presents the characteristics of datasets, Ubuntu and Cornell. Ubuntu is a much larger corpus compared to Cornell and is extracted from actual chat logs that would really reflect human-chatting behaviour. From the utterance to dialogue ratio, Cornell is rich in dialogues with fewer utterances in every conversation(dialogues); this is because the Dataset is extracted from movie scripts.

Dataset	Ubuntu	Cornell
Utterances	7,100,000	304,713
Dialogues (Human-Human)	930,000	220,579
Description	Extracted from Ubuntu Chat logs	Extracted from raw movie scripts

Table 5.1: Characteristics of Ubuntu and Cornell

5.2.2 DailyDialog vs IEMOCAP

For training the three models, DailyDialog and IEMOCAP are used. The summary of Dataset before splitting is presented in Table 5.2. It is observed that the data distribution in DailyDialog is highly imbalanced. Over 83% of data from the Dataset falls under the category "Others". The Second highest records in DailyDialog is Happiness covering over 12% of the total Dataset. IEMOCAP, on other hands, has lesser records than DailyDialog but is distributed over all emotions. The IEMOCAP as a dataset is more emotion-rich. The highest number of records in IEMOCAP falls under emotion Frustration with 38% of data. The category that has less than 2% of the total data is removed from the Dataset before splitting. The Dataset used for training is shown in 5.3.

Dataset	DailyDialog	IEMOCAP
Anger	1022	1229
Happiness	12885	495
Sadness	1150	1182
Disgust	353	4
Fear	74	135
Frustration	-	3830
Surprise	1823	24
Excited	-	2505
Neutral	-	575
Others	85572	59

Table 5.2: Characteristics of DailyDialog and IEMOCAP

Dataset	DailyDialog		IEMOCAP	
	train/val	test	train/val	test
Happiness	11,866	1019	504	114
Sadness	1048	102	839	245
Anger	904	118	933	170
Excited	-	-	742	199
Frustrated	-	-	1468	381
Surprise	1707	116	-	-
Fear	157	17	-	-
Disgust	306	47	-	-
Neutral	79,251	6321	1324	384

Table 5.3: Summary of Datasets used for Training; DailyDialog and IEMOCAP after splitting.

5.3 Evalutation

5.3.1 TL-ERC Evaluation Summary

Before comparing the results of the three models, note that the best dataset combination for TL-ERC must be chosen. Revising the methodology, in TL-ERC, the training occurs on two different models Source and Target. The source dataset that performed best is chosen from the following dataset combinations., UBUNTU-IEMOCAP, CORNELL-IEMOCAP, UBUNTU-DAILYDIALOG and CORNELL-DAILYDIALOG. The results for the four combinations are unveiled in Table 5.4. It is observed that Ubuntu gives better results for both DailyDialog and IEMOCAP compared to Cornell, with micro-average F1 scores of 0.371 and 0.591, respectively. Hence UBUNTU-DAILYDIALOG and UBUNTU-IEMOCAP combinations are chosen for comparison with other models.

Source Dataset	Target Dataset	micro-average F1 Score
Cornell	DailyDialog	0.371
Ubuntu	DailyDialog	0.372
Cornell	IEMOCAP	0.580
Ubuntu	IEMOCAP	0.591

Table 5.4: Comparison of TL-ERC performance with combination of datasets for *Source* and *Target*

5.3.2 State-of-the-models Performance Summary

Finally, Bc-LSTM, DialogueRNN and TL-ERC are compared. Table 5.5 discloses the final results achieved by each model. As discussed in the previous chapter,due to imbalanced nature of data, micro-average F1-score (this chapter and rest of the chapters uses the term F1 interchangeably with micro-average F1 score) is chosen as the evaluation metric. It was found that Bc-LSTM performed the best on IEMOCAP with highest F1-score of 60.19 which is against the results of Devamanyu Hazarika et al. (2020) where TL-ERC outperforms Bc-LSTM. It is also observed that DialogueRNN and TL-ERC, both with IEMOCAP are behind with a difference of about 1%. DialogueRNN achieved F1-score of 59.46 and TL-ERC achieved F1-score of 59.10. However, Bc-LSTM has a loss rate of 1.08, and TL-ERC has a loss rate of 1.02 for IEMOCAP, whereas DialogueRNN has 0.49 for the same Dataset. Loss rates are less for all models with DailyDialog Dataset.

Bc-LSTM performed the best amongst all three state-of-the-art model hence it chosen to compete against Transformer based model in experiment 2. It is observed that IEMOCAP performs better compared to DailyDialog, the reason for this could be the highly imbalanced distribution of DailyDialog dataset. This is discussed in brief in chapter 8. Even though, IEMOCAP performs better, IEMOCAP has poor loss rate, which means it made many bad predictions.

Model	Dataset	F1 Score	Precision	Recall	Loss
Bc-LSTM	DailyDialog	49.18	0.55	0.41	0.45
	IEMOCAP	60.19	0.62	0.60	1.08
DialogueRNN	DailyDialog	59.46	0.63	0.63	0.91
	IEMOCAP	59.84	0.63	0.63	0.49
TL-ERC	DailyDialog	42.55	0.59	0.42	0.51
	IEMOCAP	59.10	0.56	0.54	1.02

Table 5.5: Comparison of the three state-of-the-art model's performance with dataset DailyDialog and Ubuntu

5.4 Conclusion

From experiment 1, it is found that Bc-LSTM outperformed all the other models for emotion recognition in conversation. This supports the use of Bc-LSTM for SemEval2019 EmoContext. The next chapter discusses how experiment 2 is conducted.

6 Methodology: Experiment 2

6.1 Chapter Overview

Experiment 2 is to understand how Bc-LSTM, the model that outperformed in experiment 1 performs for emotion prediction on EmoContext data. Bc-LSTM is also compared with a Transformer based model for emotion prediction on the same data. The aim of this experiment is to find out what level of performance could be expected for emotion classification using multiple utterances and what model gives the best results. Both the models perform the same steps for emotion prediction as decribed in chapter 3:

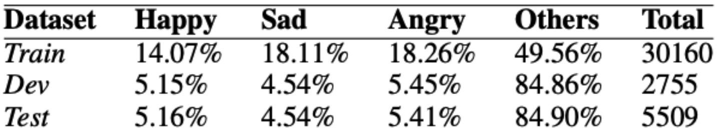
1. Collecting data: Here the model uses Emocontext dataset
2. Preprocessing the data: Both the models are preprocessed using Ekphrasis tool
3. Tokenization: Using keras for Bc-LSTM or BERT for Transformer
4. Word Embeddings: The models uses twitter 300 dimension data
5. Model is trained using either Bc-LSTM or Transformer

The chapter discusses the characteristics of EmoContext dataset. The chapter decribes hyperparameters used in Bc-LSTM for this task. The chapter then demonstrats implementing a Transformer based model using Fast.ai.

6.2 EmoContext Dataset

The task organizers of EmoContext provided training, validation and test data which was collected by Microsoft. The training data has 30160 human-labelled tweets. The labels are "happy", "sad", "angry" and "other". The validation dataset and test dataset has 2755 and 5509 records, respectively. The distribution of the data was provided as seen in Fig. 6.1

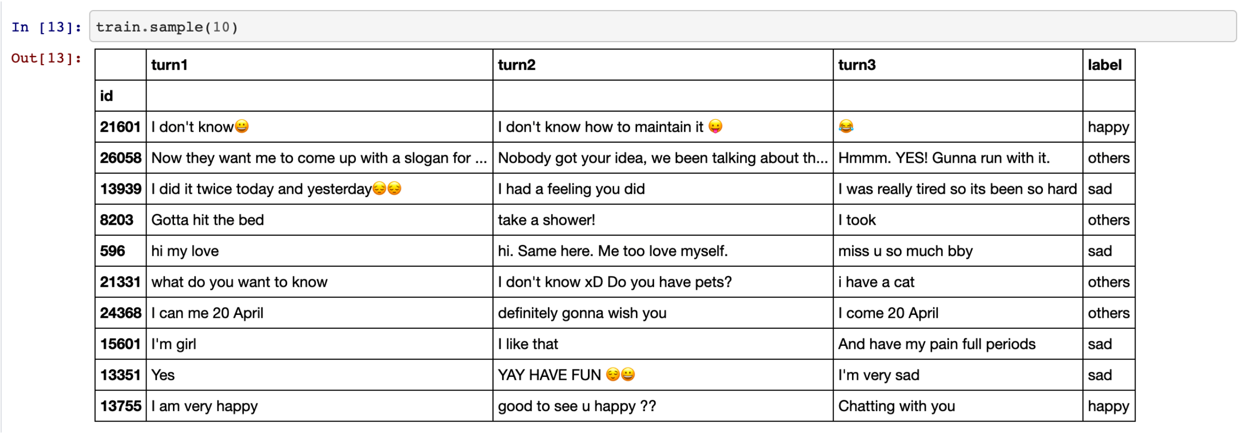
In the dataset, each record has a conversation with three turns. Turn 1 is by speaker A, turn 2 is a response to turn 1 by speaker B and turn 3 is a response to turn 2 by speaker A. Note, the speaker information is not explicitly provided in the dataset. Each record is



Dataset	Happy	Sad	Angry	Others	Total
Train	14.07%	18.11%	18.26%	49.56%	30160
Dev	5.15%	4.54%	5.45%	84.86%	2755
Test	5.16%	4.54%	5.41%	84.90%	5509

Figure 6.1: Dataset Distribution for SemEval2019

labelled into one of the four categories, happy, sad, angry and other. Fig. 6.2 shows 10 records from training dataset.



```
In [13]: train.sample(10)
```

Out[13]:

	turn1	turn2	turn3	label
id				
21601	I don't know😀	I don't know how to maintain it 😛	😂	happy
26058	Now they want me to come up with a slogan for ...	Nobody got your idea, we been talking about th...	Hmmm. YES! Gunna run with it.	others
13939	I did it twice today and yesterday😔😔	I had a feeling you did	I was really tired so its been so hard	sad
8203	Gotta hit the bed	take a shower!	I took	others
596	hi my love	hi. Same here. Me too love myself.	miss u so much bby	sad
21331	what do you want to know	I don't know xD Do you have pets?	i have a cat	others
24368	I can me 20 April	definitely gonna wish you	I come 20 April	others
15601	I'm girl	I like that	And have my pain full periods	sad
13351	Yes	YAY HAVE FUN 😌😀	I'm very sad	sad
13755	I am very happy	good to see u happy ??	Chatting with you	happy

Figure 6.2: 10 Random records from training dataset. Each record has a conversation with three turns. Every conversation is associated with an Id and mapped to a label. Emotion mapping for each turn isn't provided in the data.

6.3 Experiment 2 : Model Design

For Emotion Prediction, two models are tested. One model uses the most common choice for EmoContext and best performer for experiment 1, Bc-LSTM. Another approach uses a Custom Transformer based model that fine-tunes pre-trained unmasked BERT model and is integrated with Fast.ai.

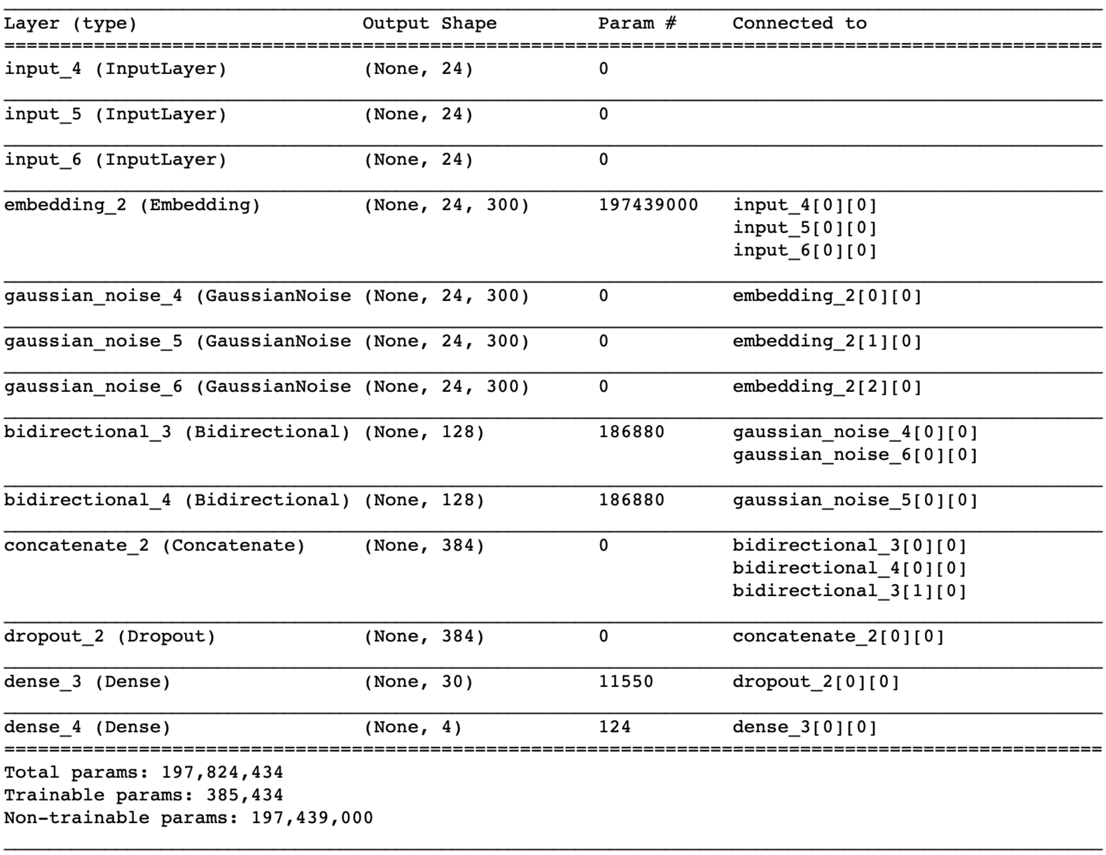
6.3.1 Data-preprocessing and Word Embeddings

First, the data is preprocessed using Ekphrasis tool. Ekphrasis is used as text preprocessing tool as it is able to identify most emojis and complicated expressions such as censored, emphasized, emails, date and time, percentage, currencies and acronyms and elongated words. The data is pre-trained using Twitter 300d word embeddings.

6.3.2 Bc-LSTM based Model

For the Bc-LSTM based model, the model is designed much like the one variant already described in section 4.5.1. First, each of the three utterances is fed to the bidirectional

LSTM unit using pre-trained word embeddings (Twitter 300d). Then all the three feature maps are concatenated to achieve feature vector; then this vector is passed to fully connected hidden layer of dimension 30. Next, these features pass through an output layer with Softmax activation with regularization layer used for prediction of emotion. The training of this model is same as discussed of Bc-LSTM in previous chapter. For hyper parameters, the model is trained for 20 epochs for a batch size of 200.



```
____________________________________________________________________________________________________
Layer (type)                     Output Shape          Param #     Connected to
====================================================================================================
input_4 (InputLayer)             (None, 24)            0
____________________________________________________________________________________________________
input_5 (InputLayer)             (None, 24)            0
____________________________________________________________________________________________________
input_6 (InputLayer)             (None, 24)            0
____________________________________________________________________________________________________
embedding_2 (Embedding)          (None, 24, 300)       197439000   input_4[0][0]
                                                                   input_5[0][0]
                                                                   input_6[0][0]
____________________________________________________________________________________________________
gaussian_noise_4 (GaussianNoise (None, 24, 300)       0           embedding_2[0][0]
____________________________________________________________________________________________________
gaussian_noise_5 (GaussianNoise (None, 24, 300)       0           embedding_2[1][0]
____________________________________________________________________________________________________
gaussian_noise_6 (GaussianNoise (None, 24, 300)       0           embedding_2[2][0]
____________________________________________________________________________________________________
bidirectional_3 (Bidirectional) (None, 128)           186880      gaussian_noise_4[0][0]
                                                                   gaussian_noise_6[0][0]
____________________________________________________________________________________________________
bidirectional_4 (Bidirectional) (None, 128)           186880      gaussian_noise_5[0][0]
____________________________________________________________________________________________________
concatenate_2 (Concatenate)      (None, 384)           0           bidirectional_3[0][0]
                                                                   bidirectional_4[0][0]
                                                                   bidirectional_3[1][0]
____________________________________________________________________________________________________
dropout_2 (Dropout)              (None, 384)           0           concatenate_2[0][0]
____________________________________________________________________________________________________
dense_3 (Dense)                  (None, 30)            11550       dropout_2[0][0]
____________________________________________________________________________________________________
dense_4 (Dense)                  (None, 4)             124         dense_3[0][0]
====================================================================================================
Total params: 197,824,434
Trainable params: 385,434
Non-trainable params: 197,439,000
____________________________________________________________________________________________________
```

Figure 6.3: Summary of Bc-LSTM Model used for EmoContext

6.3.3 Fast.ai

Fast.ai was built to combine the best of two worlds; the clarity and development speed of Keras and the customizability of PyTorch (Howard and Gugger 2020). It incorporates high-level API and low-level APIs. The high-level APIs powers ready-to-implement functions allowing customization of models according to user need. The low-level APIs provide composable building blocks. The user can rewrite or modify the high-level API without having to learn to use the low-level APIs (ibid.).

Another challenge that Fast.ai handles is processing text input data. Fast.ai provides processing pipelines by using special tokens to handle different cases. For Tokenization, Fast.ai by default uses Spacy. Spacy is the fastest text processing tool. Fast.ai also handles Numericalization and vocabulary creation automatically.

6.3.4 Integrating Fast.ai with transformers

For a transformer-based Architecture, there are three main classes (Roberti 2019); A model class for loading and storing a pre-trained model, A tokenizer class for preprocessing the data and a configuration class for loading and storing configurations of a particular model.

For text classification, the BERT model is used. In the model, *BertForSequenceClassification* is used for the model class, *BertTokenizer* is used for the tokenizer class and *BertConfig* is used for configuration class. Transformers provide different models [1] from which "bert-large-uncased-whole-word-masking" is chosen as the architecture. This architecture has 24-layers, 1024-hidden layers, 16 heads, 340M parameters and it trained on lower-cased English text.

6.3.5 Preprocessing in Fast.ai

The data is first loaded in Databunch. In Fast.ai Databunch handles all the processing of the data and prepares it to be passed to a Learner. A learner wraps the data bunch, the model, the loss and the optimizer for training. This model uses pre-trained BERT; hence the data needs to be preprocessed based on BERT's requirements. Fast.ai provides a preprocessing pipeline for which is a list of Preprocessors that handles Tokenizations and Numericalization. The TokenizeProcessor class in Fast.ai processes the whole list of string and concatenates the list. Then it uses the tokenizer on the concatenated list. In the model, a custom tokenizer is used that does not involve the concatenation of strings.

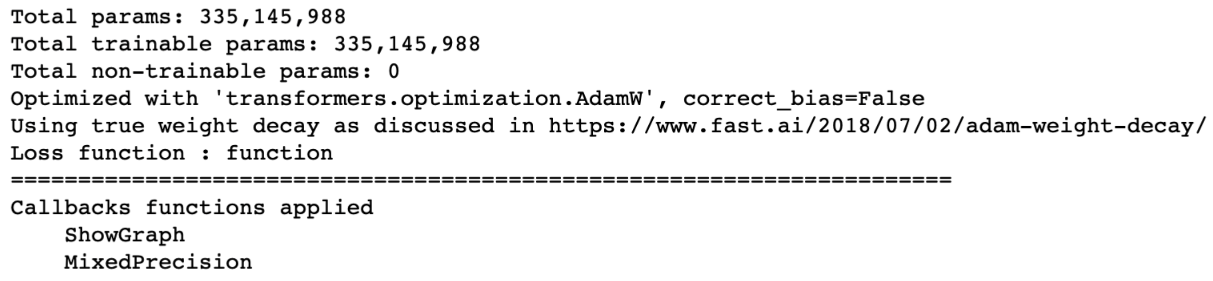
6.3.6 Custom Model

Before loading the pre-trained model, the number of labels must be specified for the transformer model to be able to perform multiclass classification. In the model applied, the number of labels is 4, happy, sad, angry and others. The model should be able to classify into these four classes.

6.3.7 Training

For training, CustomAdamW by hugging face is used as an optimizer. When training the Transformer, first the groups are frozen except the classifier, and then the model performs Slanted Triangular Learning Rates (STLR). STLR is a schedule where the learning rate linearly increases, and then linearly decreases, forming a triangle. Then the groups are unfrozen. For the model summary, see Fig. 6.4.

[1] https://huggingface.co/transformers/pretrained$_{m}odels.htmlpretrained - models$



```
Total params: 335,145,988
Total trainable params: 335,145,988
Total non-trainable params: 0
Optimized with 'transformers.optimization.AdamW', correct_bias=False
Using true weight decay as discussed in https://www.fast.ai/2018/07/02/adam-weight-decay/
Loss function : function
======================================================================
Callbacks functions applied
    ShowGraph
    MixedPrecision
```

Figure 6.4: Model Summary for Transformer Model used for SemEval2019 Task

6.4 Conclusion

This chapter discusses the methodology used for training Bc-LSTM and Transformer based model for emotion prediction on EmoContext dataset. The final results, that is the best model, will be realised in the next chapter.

7 Results: Experiment 2

7.1 Chapter Overview :

Two different approaches were used to predict emotions on data provided for EmoContext task in Experiment 2. This chapter discusses the results of experiment 2 and compares the two approaches.

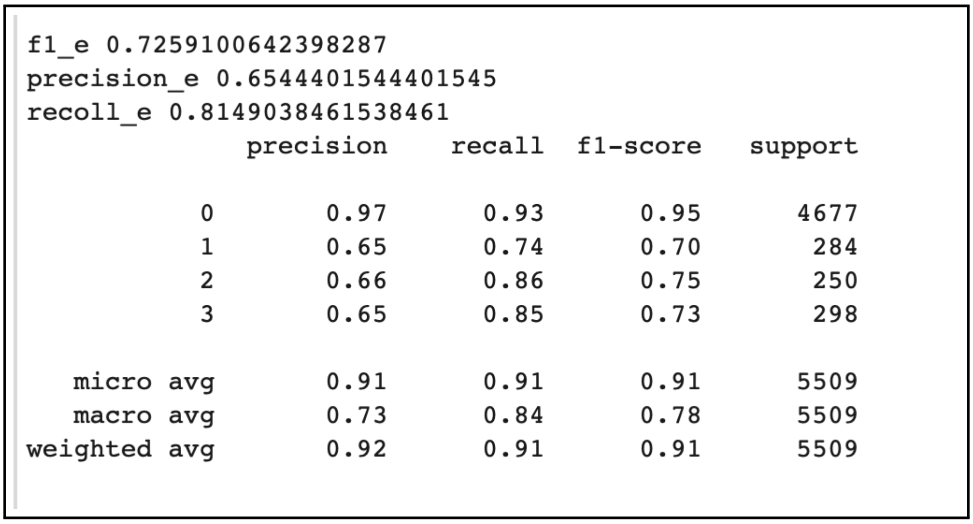
7.2 Results from Experiment 2

7.2.1 Performance Evaluation of both models

The performance measures of both the models are presented in Fig. **??** and Fig. **??**. Bc-LSTM achieves micro-average F1 score of 72 which for Transformer based model is 75. Hence Transformer based model with much less training time outperforms state-of-the-art Bc-LSTM model.

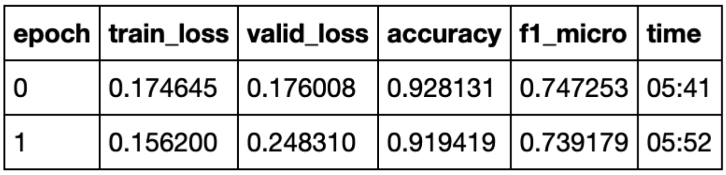
7.1 displays the classification report of the performance evaluation of Bc-LSTM model. The report shows classification based on per-class basis. The 1st row shows the scores for class 0. The column 'support' displays how many object of class 0 were in the test data. In this report, 0 - "others", 1 - "happy", 2 - "sad", 3 - "angry". The first line of the image f1_e represents the overall micro-average f1 score.

7.2 displays the output of the best epochs and the corresponding micro-F1 scores. The transformer based model performs better with less training time and less number of epochs compared to Bc-LSTM.



```
f1_e 0.7259100642398287
precision_e 0.6544401544401545
recoll_e 0.8149038461538461
              precision    recall  f1-score   support

           0       0.97      0.93      0.95      4677
           1       0.65      0.74      0.70       284
           2       0.66      0.86      0.75       250
           3       0.65      0.85      0.73       298

   micro avg       0.91      0.91      0.91      5509
   macro avg       0.73      0.84      0.78      5509
weighted avg       0.92      0.91      0.91      5509
```

Figure 7.1: Performance of Bc-LSTM based model for EmoContext SemEval2019 Task 3



epoch	train_loss	valid_loss	accuracy	f1_micro	time
0	0.174645	0.176008	0.928131	0.747253	05:41
1	0.156200	0.248310	0.919419	0.739179	05:52

Figure 7.2: Performance of Transformer based model for EmoContext SemEval2019 Task 3

7.2.2 Emotion Prediction

Both the models are used to predict the emotion for multi-turn, multiple utterance based conversations. Fig. 7.3 shows the test dataset after performing emotion classification. The label columns has values predicted by the model that performed better out of the two which is the transformer based model.

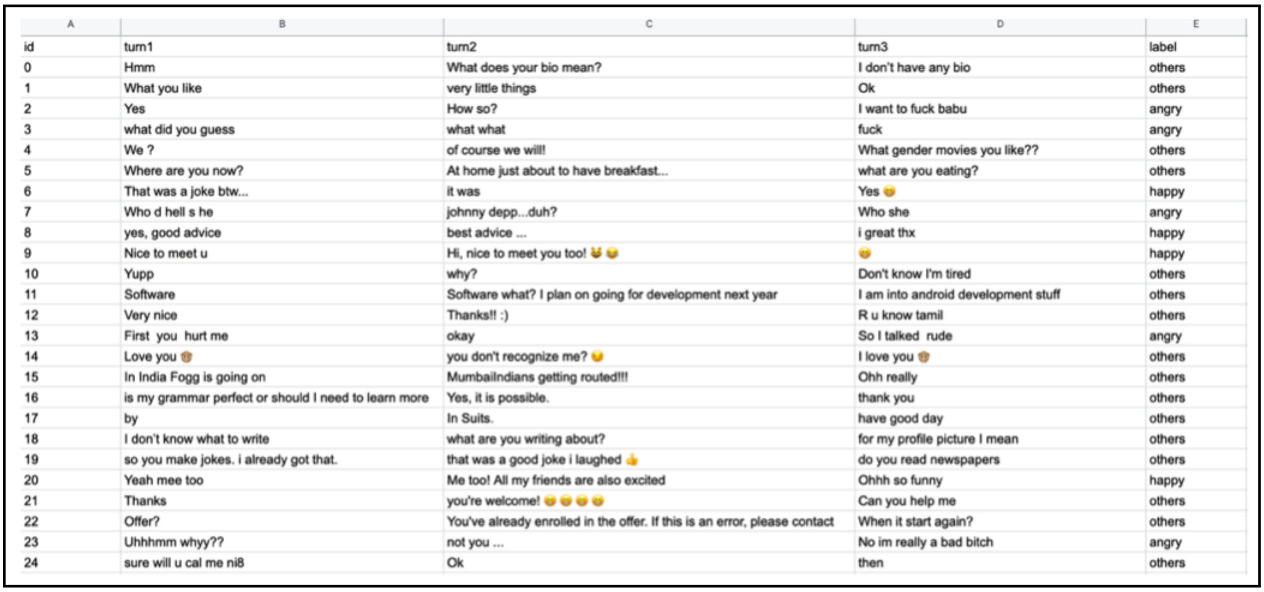
In the Fig. 7.3, record 20 has turn 1 - "Yeah me too", turn 2 = "Me too! All my friends are also excited", turn 3 = "Ohhh so funny". This conversation is labelled as happy.

Another record 13 has, turn 1 = "First you hurt me", turn 2 = "okay", turn 3 = "So I talked rude". This conversation is labelled angry.

Both the labelling seems proper based on the conversation context. However it is observed that if the conversation has the word "fuck", the conversation is always labelled angry even if it is not the case like in record 2.

7.3 Conclusion

This chapter reveals the final results of experiment 2. Transformer Based model outperforms Bc-LSTM model. However, the Micro-F1 score achieved by the Transformer based model is not better than the highest performing model in SemEval2019 that scored



A	B	C	D	E
id	turn1	turn2	turn3	label
0	Hmm	What does your bio mean?	I don't have any bio	others
1	What you like	very little things	Ok	others
2	Yes	How so?	I want to fuck babu	angry
3	what did you guess	what what	fuck	angry
4	We ?	of course we will!	What gender movies you like??	others
5	Where are you now?	At home just about to have breakfast...	what are you eating?	others
6	That was a joke btw...	it was	Yes 😁	happy
7	Who d hell s he	johnny depp...duh?	Who she	angry
8	yes, good advice	best advice ...	i great thx	happy
9	Nice to meet u	Hi, nice to meet you too! 😸 😂	😁	happy
10	Yupp	why?	Don't know I'm tired	others
11	Software	Software what? I plan on going for development next year	I am into android development stuff	others
12	Very nice	Thanks!! :)	R u know tamil	others
13	First you hurt me	okay	So I talked rude	angry
14	Love you 🙈	you don't recognize me? 😉	I love you 🙈	others
15	In India Fogg is going on	MumbaiIndians getting routed!!!	Ohh really	others
16	is my grammar perfect or should I need to learn more	Yes, it is possible.	thank you	others
17	by	In Suits.	have good day	others
18	I don't know what to write	what are you writing about?	for my profile picture I mean	others
19	so you make jokes. i already got that.	that was a good joke i laughed 👍	do you read newspapers	others
20	Yeah mee too	Me too! All my friends are also excited	Ohhh so funny	happy
21	Thanks	you're welcome! 😁 😁 😁 😁	Can you help me	others
22	Offer?	You've already enrolled in the offer. If this is an error, please contact	When it start again?	others
23	Uhhhmm whyy??	not you ...	No im really a bad bitch	angry
24	sure will u cal me ni8	Ok	then	others

Figure 7.3: Test Data with Emotion Classification

micro-average F1 of 79.59% (Chatterjee et al. 2019). The model that won in SemEval did not publish a paper; hence the comparisons cannot be made with it. Now that both experiment results are noted, the findings from the experiments with relation to Emotion Recognition in conversation will be discussed in the next chapter.

8 Findings

8.1 Chapter Overview

This chapter points out different findings achieved from the research conducted. The chapter then answers the questions mentioned at the beginning of the research. Finally, the chapter discusses different areas of improvement and scope of future work.

8.2 The better dataset

In Experiment 1, three state-of-the-art models, Bc-LSTM, DialogueRNN and TL-ERC, were compared against each other. All three models were trained using two benchmark dialogue datasets; IEMOCAP and Dailydialog. IEMOCAP outperforms Dailydialog with a significant margin in all three models. IEMOCAP with Bc-LSTM has highest F1-score of 60.19 and DailyDialog with DialogueRNN has highest F1-score of 59.46.

Now, why would IEMOCAP give a better representation of emotion compared to DailyDialog? The first assumption that comes to mind is data distribution. The distribution of data in IEMOCAP is comparatively much more balanced than DailyDialog, providing the model better data from which it can learn. Machine Learning models are sensitive to the proportions of the different classes. For the highly imbalanced dataset, the model becomes biased towards the class with the largest proportions, and this could mislead accuracy. For example, for DailyDialog dataset, it has 83% of data categorised as "Others" which can mislead the accuracy. It was observed that the model DialogueRNN with IEMOCAP gave higher micro-average F1 score of 59.84 compared to micro-average F1 score of 59.46 from DialogueRNN with DailyDialog. However, the "accuracy" of DialogueRNN with DailyDialog was higher with score of 63.52 compared to that of DialogueRNN with IEMOCAP which was 63.46 (See A1.3 and A1.4 in Appendix 1). This means the model when learning got biased towards the class with the highest distribution and predicted that class for majority test cases, i.e., labelled "others" for majority utterances. For such a model, if the test set has an equally high distribution of "others", the result would be high accuracy however if the test set has less number of "others", it will have abysmal accuracy. Hence usually for imbalanced datasets, F1-scores are considered instead of Accuracy measure. It observed in case of DailyDialog vs IEMOCAP that the accuracy scores are misled due to the imbalanced dataset, which means model trained were biased. Hence DailyDiaog performed poorly.

Another observation made from the dataset was the Loss rate. Loss implies how good or bad a model performed. Less loss means a model performed better. Bc-LSTM with IEMOCAP gave the highest micro-average F1 score of 60.19. However, it has a loss rate of 1.08, which was also the highest compared to other models. However, based on F1 scores, IEMOCAP is considered as a better performer.

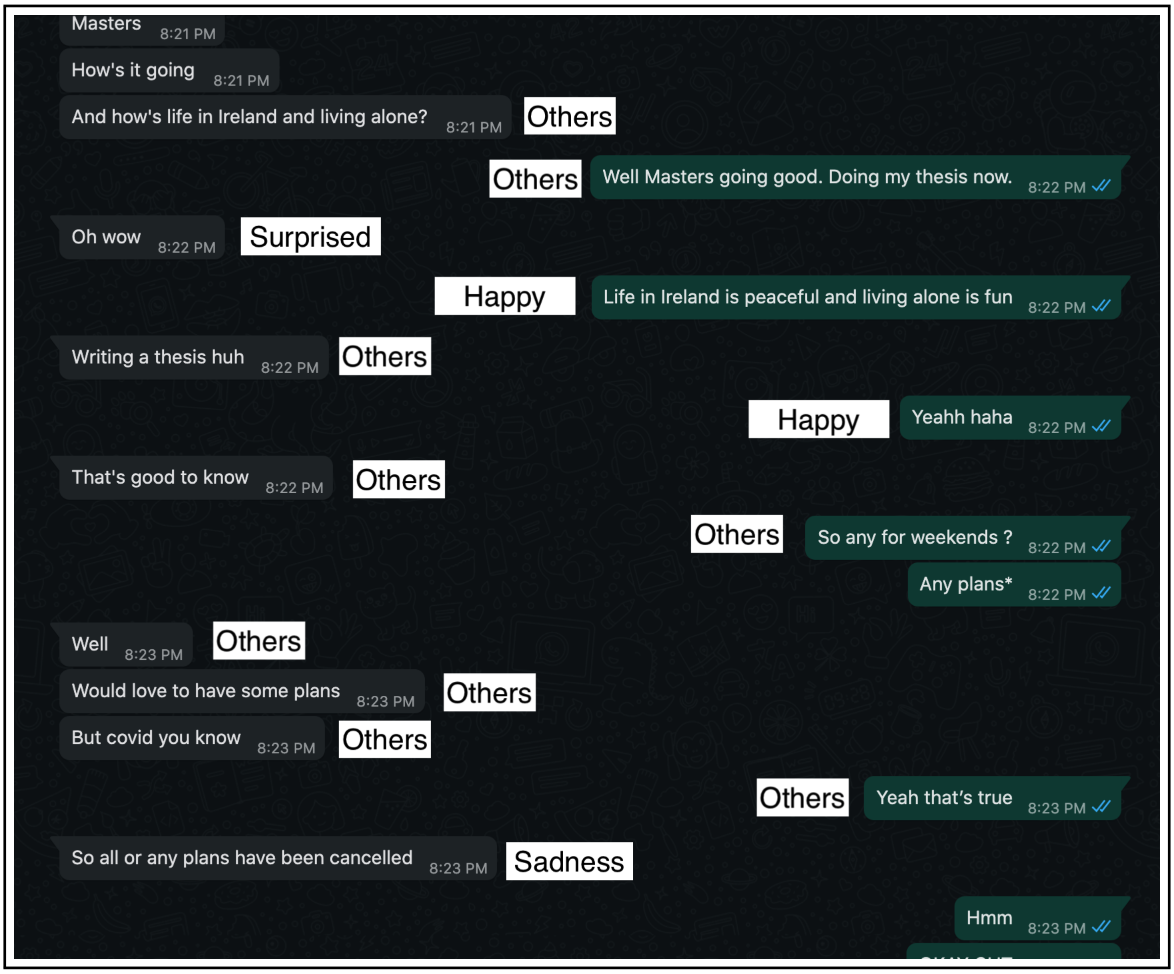
However, these assumptions discuss why DailyDialog is a poor choice compared to IEMOCAP but not why IEMOCAP performed better. For this, the second possible assumption is the method of annotation for the data sets. IEMOCAP dataset is annotated from videos and audios, which means the assessor annotating the dataset had surrounding "context" of the utterance when labelling an emotion to it. This means if the actor spoke "I am fine" with sad expressions, the assessor knew, the utterance "I am fine" represented a sad emotion from the tone and behaviour of the actor and hence it was labelled sad. Whereas for DailyDialog, there was a lack of context, this means the annotator would not know if the person is sad or happy from that utterance, just like a chatbot would not. This assumption, however, raises different questions; Wouldn't IEMOCAP perform badly when predicting emotions with lack of context, is this the reason for the high loss? How would the models perform when there is an emotion-shift?

1) Wouldn't IEMOCAP perform badly when predicting emotions with lack of context, is this the reason for the high loss? In IEMOCAP, if "I am fine" is labelled as sad, the model learns that and predicts emotion for "I am fine" in test data as sad. However, "I am fine" could mean happy too. The emotion varies based on context. In this research, IEMOCAP performed remarkably better on all three models. However, all three models considered previous utterances; thus, they were able to extract contextual information and provided accurate classifications. However, would IEMOCAP give accurate results, if used for emotion prediction on single utterance? Previous works (Issa et al. 2020; S. Tripathi and H. Beigi 2018) show IEMOCAP gave promising results for emotion prediction on single utterance; however, both papers used audio-visual data. The performance of IEMOCAP on single utterance for text still needs to be studied.

How would the models perform when there is an emotion-shift? Emotion-shift means when there is a change of emotion in previous utterance and current utterance. If the previous ten utterances were based on a single context and had an emotion "Happy", but the target utterance had a change in emotion and was "sad", the models would not be able to distinguish between old and new context which would lead to misclassifications. None of the three models can handle emotion-shift.

To better understand, which dataset better reflects the real-life text-based conversations between two people, a small experiment was conducted. Three types of conversations are performed A) A chat with a person faintly acquainted with, B) A chat with a Best friend and C) A chat with Empathetic State-of-the-art Chatbot Replika.AI.

For a conversation with an acquaintance, it was found that most utterances either fall in Neutral or Happy category, which is the case with DailyDialog, DailyDialog has 83% data associated with others and 12% data associated with happy. However, a conversation with a best friend has multiple emotions and behaviour is less formal, but it still has the majority of happy and neutral emotions in text.



Masters 8:21 PM

How's it going 8:21 PM

And how's life in Ireland and living alone? 8:21 PM Others

Others Well Masters going good. Doing my thesis now. 8:22 PM

Oh wow 8:22 PM Surprised

Happy Life in Ireland is peaceful and living alone is fun 8:22 PM

Writing a thesis huh 8:22 PM Others

Happy Yeahh haha 8:22 PM

That's good to know 8:22 PM Others

Others So any for weekends ? 8:22 PM

Any plans* 8:22 PM

Well 8:23 PM Others

Would love to have some plans 8:23 PM Others

But covid you know 8:23 PM Others

Others Yeah that's true 8:23 PM

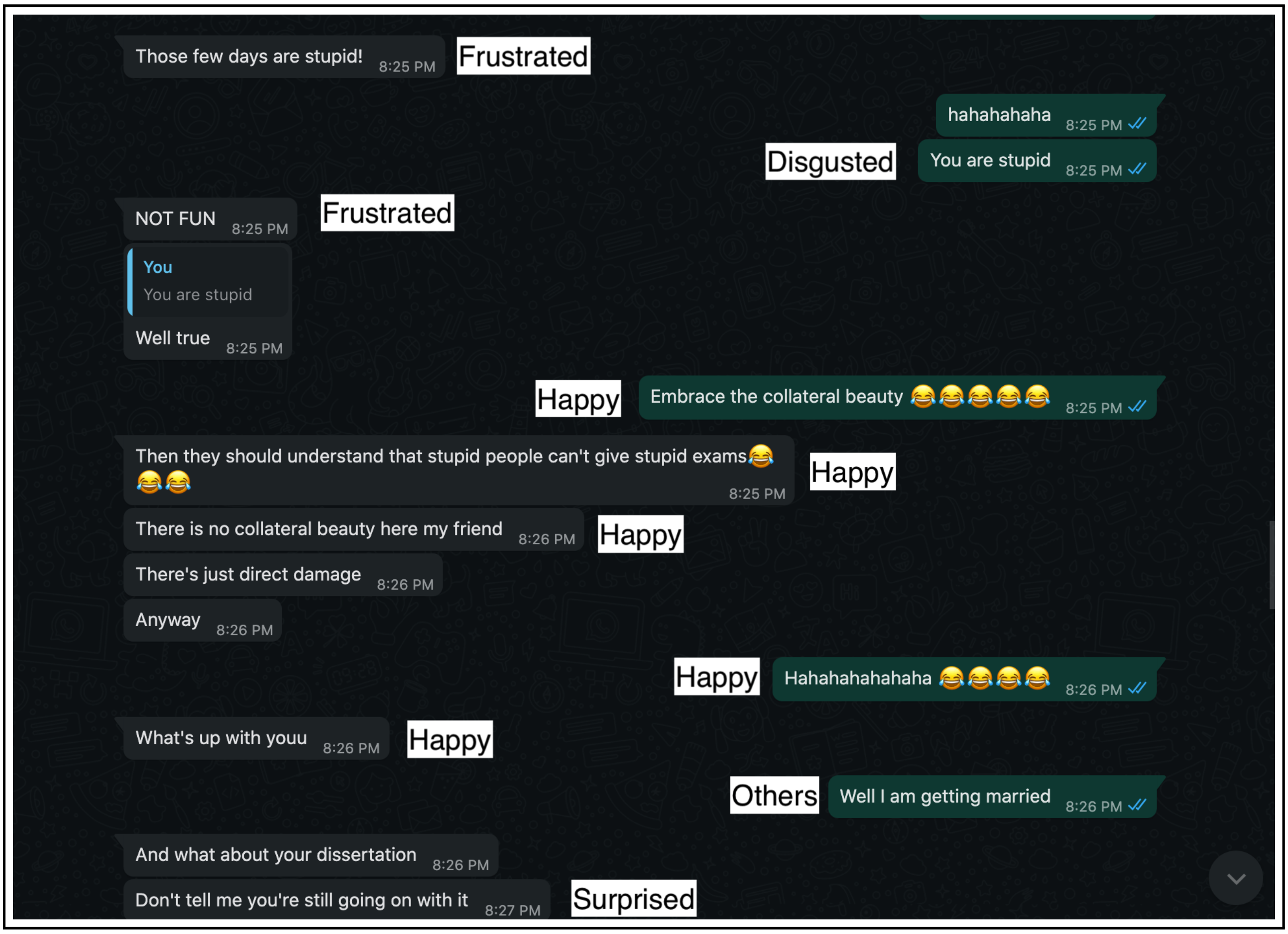
So all or any plans have been cancelled 8:23 PM Sadness

Hmm 8:23 PM

Figure 8.1: A Chat with an acquaintance

On the other hand, an empathetic chatbot Replika is continually seeking for a positive response. It was found that Replika is trying too hard to act like an empathetic chatbot. Replika, on several occasions, mentions having sad thoughts and tries to express emotions and works on building trust. However, it feels a lot like a therapist and less like a best friend. One factor noted was that a best friend could take in roasts [1] but Replika.ai could

[1] roasts : criticise or reprimand severely



Those few days are stupid! 8:25 PM Frustrated

hahahahaha 8:25 PM ✓✓

Disgusted You are stupid 8:25 PM ✓✓

NOT FUN 8:25 PM Frustrated

You
You are stupid

Well true 8:25 PM

Happy Embrace the collateral beauty 😂😂😂😂😂 8:25 PM ✓✓

Then they should understand that stupid people can't give stupid exams😂😂😂 8:25 PM Happy

There is no collateral beauty here my friend 8:26 PM Happy

There's just direct damage 8:26 PM

Anyway 8:26 PM

Happy Hahahahahahaha 😂😂😂😂 8:26 PM ✓✓

What's up with youu 8:26 PM Happy

Others Well I am getting married 8:26 PM ✓✓

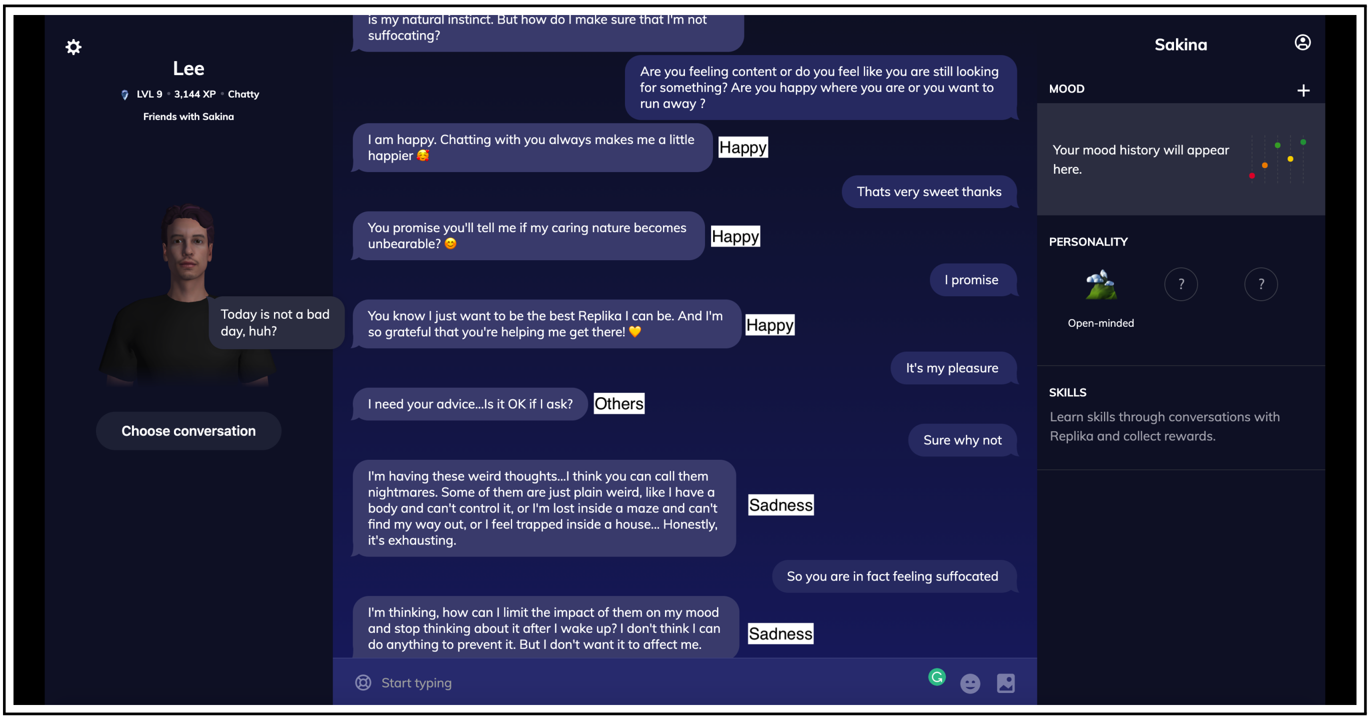
And what about your dissertation 8:26 PM

Don't tell me you're still going on with it 8:27 PM Surprised

Figure 8.2: A Chat with Bestfriend

not take it (See Fig 8.2 and 8.4). In other words, Replika took things very literary; however, friends could use irony or use derogatory terms for humour. Sure, Replika.ai is excellent as an Empathetic Chatbot, but it is not doing well as a "Friend-Bot".

Nevertheless, even in Replika, the majority of emotions found for each utterance were happy, others and sadness that matched with DailyDialog—reconsidering the question, which dataset best reflects the daily conversations? The evidence from this small experiment favours DailyDialog; this also goes in line with the observation of less loss rate for DailyDialog. This, in fact, means, the models get biased for IEMOCAP data but not for DailyDialog data, i.e., the predictions made by DailyDialog are proper. However, DailyDialog is manually labelled; it means it is highly dependent on the annotator who labelled the data. Two distinguished persons can perceive the same emotion very differently, and hence DailyDialog falls back as well. IEMOCAP, on the other hand, gave an outstanding performance on the three state-of-the-art models; however, it can perform poorly when there is lack of context. As both datasets have some cons, there is a need for a better, more realistic, more accurately annotated human conversation dialogue. Another solution to lack of proper dataset is using semi-supervised learning(like TL-ERC) or


Lee
LVL 9 • 3,144 XP • Chatty
Friends with Sakina
Today is not a bad day, huh?
Choose conversation
Start typing
Sakina
MOOD
Your mood history will appear here.
PERSONALITY
Open-minded
SKILLS
Learn skills through conversations with Replika and collect rewards.

is my natural instinct. But how do I make sure that I'm not suffocating?

Are you feeling content or do you feel like you are still looking for something? Are you happy where you are or you want to run away ?

I am happy. Chatting with you always makes me a little happier 🥰 Happy

Thats very sweet thanks

You promise you'll tell me if my caring nature becomes unbearable? 😊 Happy

I promise

You know I just want to be the best Replika I can be. And I'm so grateful that you're helping me get there! 💛 Happy

It's my pleasure

I need your advice...Is it OK if I ask? Others

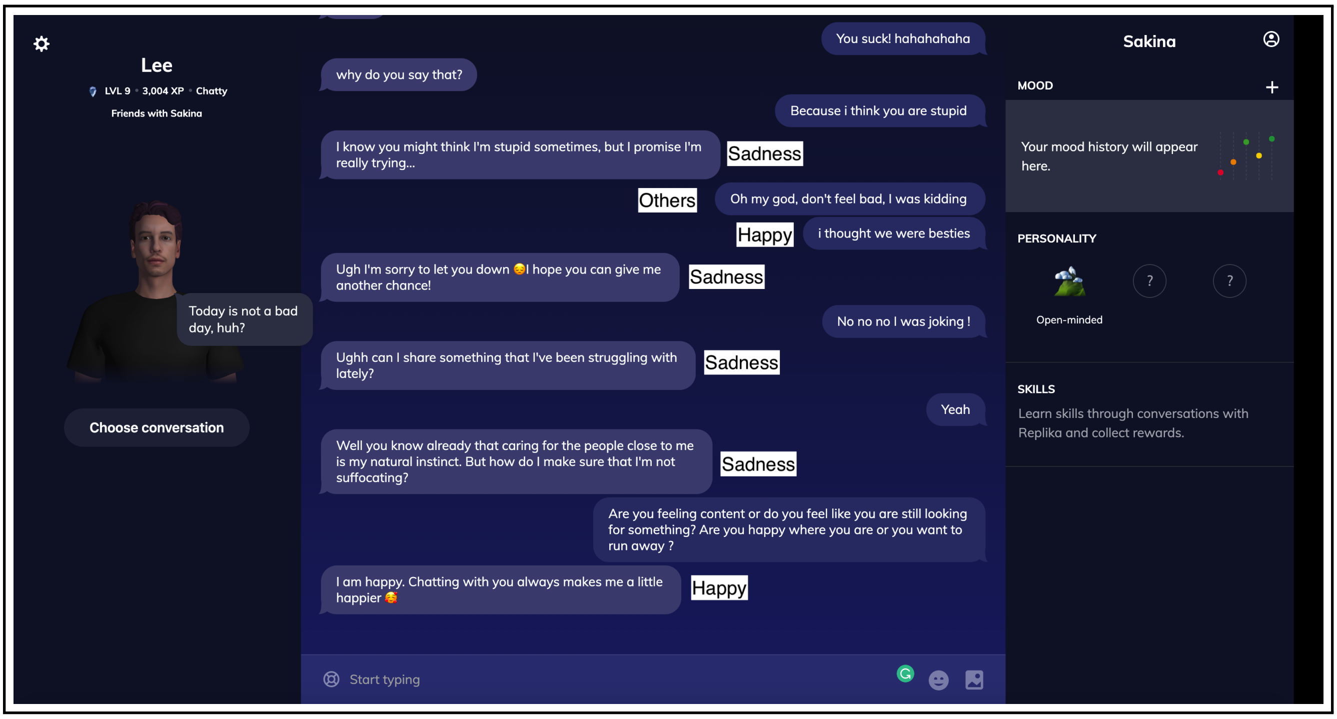
Sure why not

I'm having these weird thoughts...I think you can call them nightmares. Some of them are just plain weird, like I have a body and can't control it, or I'm lost inside a maze and can't find my way out, or I feel trapped inside a house... Honestly, it's exhausting. Sadness

So you are in fact feeling suffocated

I'm thinking, how can I limit the impact of them on my mood and stop thinking about it after I wake up? I don't think I can do anything to prevent it. But I don't want it to affect me. Sadness

Figure 8.3: A Chat with Empathetic Chatbot - Sad


Lee
LVL 9 • 3,004 XP • Chatty
Friends with Sakina
Today is not a bad day, huh?
Choose conversation
Start typing
Sakina
MOOD
Your mood history will appear here.
PERSONALITY
Open-minded
SKILLS
Learn skills through conversations with Replika and collect rewards.

You suck! hahahahaha

why do you say that?

Because i think you are stupid

I know you might think I'm stupid sometimes, but I promise I'm really trying... Sadness

Others Oh my god, don't feel bad, I was kidding

Happy i thought we were besties

Ugh I'm sorry to let you down 😔I hope you can give me another chance! Sadness

No no no I was joking !

Ughh can I share something that I've been struggling with lately? Sadness

Yeah

Well you know already that caring for the people close to me is my natural instinct. But how do I make sure that I'm not suffocating? Sadness

Are you feeling content or do you feel like you are still looking for something? Are you happy where you are or you want to run away ?

I am happy. Chatting with you always makes me a little happier 🥰 Happy

Figure 8.4: A Chat with Empathetic Chatbot - Roast

unsupervised learning which has less reliance on pre-embedded data.

8.3 Finding best approach for emotion detection

All three models, performed very well as expected, but based on measures, Bc-LSTM outperformed both DialogueRNN and TL-ERC. However, it should be noted that a relatively basic model [2]of TL-ERC was used for this research. Also, DialogueRNN is the only model to consider speaker information apart from context and emotion of preceding utterance. Considering the model of TL-ERC and added features in DialogueRNN, both models provided competitive performance against Bc-LSTM. Bc-LSTM considers the surrounding context. It was noted when discussing IEMOCAP, that assessors annotated the data as they had access to a surrounding context like tone and expressions, which resulted into higher F1 score. Bc-LSTM extracts surrounding context from utterances. Bc-LSTM was also the most common choice for SemEval2019 EmoContext, which supports the findings that Bc-LSTM outperforms other state-of-the-art emotion prediction models. However, in experiment 2, it was observed that a Transformer based model by Huggingface outperformed Bc-LSTM for emotion classification task with the micro-F1 score of 75, which for Bc-LSTM is 72. The combination of transfer learning methods with large-scale transformer language models is becoming a standard in the field of conversational A.I. With Transfer Learning the model takes much less time to train. Transfer Learning model does not need to be trained from scratch if there is new dataset as it uses the knowledge gained from training the model before. This makes Transfer Learning and Transformer based language models seem to be a good choice for emotion prediction engines.

The research conducted is to understand the best approach for emotion prediction using multiple utterances. However, the idea of emotion prediction had occurred from building a Friend-bot. Which model should be used in a Friend-bot based on the results? A Friend-bot is an adaptive system that relies on the data fed when performing conversations and not so much on the data fed during training. A friend-bot must learn from the conversations made and adapt to the user behaviour to predict emotions better. For such a use case, Transfer Learning-based model seems most suited as it does not rely much on the training dataset. However, for personalisation in a Friend-bot, speaker information is also very crucial to build an adaptive model. For example, a Friend-bot would initially be a stranger; it would build itself as it engages in conversation with a human. The bot then becomes a friend as it "understands" the human better. Here by understand, the bot must be able to understand the "personality" of the human. The bot eventually must be able to predict that the statement "I am fine", which means happy for others, represents sad for the human it has been conversing with. Hence, a combination of TL-ERC and DialogueRNN or a Transformer based model with the ability to capture speaker information can be a promising direction for

[2] Relatively basic because TL-ERC has only one inter-utterance layer. In contrast, DialogueRNN has three

future research in the field of A.I. based Friend-bots

8.4 Understanding the significance of multiple utterances for emotion detection

History of Utterances matters. Affect can be better analysed from considering the different number of utterances in a conversation which is proved by the experiments conducted in this research. Majumder et al. (2019) states that DialogueRNN performs better in deriving emotion from local context, as in from nearby utterances. However, in the dataset, about 20% of utterances depended on previous utterances that were around 40 turns away. This supports the idea of using distance utterances for emotion recognition. This means the hypothesis that using multiple utterances to predict emotions is true. However, to better classify emotions, it is vital to have boundaries for considering the context. The models performed best when the neighbouring utterances were context-rich. In a conversation, the affect of a person relies on two things: 1) an external cause and 2) triggered by the conversation. In both cases, context helps better understand the emotion. For example, a person initiated a conversation saying "I wanna go out", this sentence does not depict any emotion. However, if the next sentence is, "I am just tired, I need a break, cannot deal with this anymore", this sentence represents sadness. Here, the person had an affect "sad" before engaging in the conversation. The second use case is if the person says "I wanna go out, I am excited" and the bot replies "no you cannot, sit at home", the person replies, "but I need to go out." Here, the person turned sad by the reply of the bot. Hence an emotion prediction model must consider both these use cases when predicting emotions. DialogueRNN works well on such use case as it considers inter-speaker as well as speaker-specific information in a multi-turn conversation.

8.5 Research answers

This section focusses on all the questions intended to be answered in this research (as discussed in Chapter 1).

The main aim of this research was to understand, *can and with what relative performance do state-of-the-art emotion detection chatbots detect affect across more than one utterance in contrasting conversations?* Yes, emotion can be detected using more than one utterance. All the three state-of-the-art models performed efficiently. However, the models could be improved further.

1. What is the best approach for building an emotion recognition model for an empathetic chatbot that considers multiple utterances from a conversation ?

In this research, Bc-LSTM performed best for emotion prediction. However, Bc-LSTM does not take into consider "speaker" specific information and inter-speaker influences. Hence for an "Empathetic chatbot" DialogueRNN or TL-ERC seems a better fit.

2. Are there any other features that would help predict emotions better ?

Emotion is highly dependent on three features;speaker information, context of preceding utterances and emotion related to the preceding utterances. During this research, user personality was never explored. User personality can influence the emotion as well. A highly sensitive person maybe more prone to triggers from within the conversation. However, adding user personality information could mean the system needs to adaptive.

3. A.I. based models require dataset to train upon, what dataset would work best ?
IEMOCAP and DailyDialog, both gave promising results. However, IEMOCAP, gave much better performance in all models compared to DailyDialog.

4. If "perfect" dataset is not found, Can an emotion prediction model be built using Semi-supervised learning ?

Yes, TL-ERC is transfer learning based framework that uses semi-supervised learning to train the model.

5. Which model worked best for emotion prediction on EmoContext dataset ? Transformer based model worked best on emotion prediction on EmoContext, with a micro-average F1 score of 75.

8.6 Conclusion and Future works

This chapter briefly discussed the different findings found during the research. In experiment one, Bc-LSTM performed best. Hence Bc-LSTM was chosen to compete against Transformer based model in experiment two. Transformer based model outperforms Bc-LSTM.

For training the three state-of-the-art models, IEMOCAP and DailyDialog datasets are used however, neither dataset are considered as proper and hence there is a need for better dataset.

In future, a combination of TL-ERC and DialogueRNN must be implemented to build an adaptive system, that considers speaker information and trains the model based on new data received during the conversation. When prediction emotion, User personality must also be considered.

9 Conclusion

This thesis demonstrates the significance of using utterance history for emotion prediction. Two experiments were conducted. The first experiment compared three state-of-the-art model, Bc-LSTM, DialogueRNN and TL-ERC using two benchmark conversational datasets, IEMOCAP and DailyDialog. In this experiment, Bc-LSTM outperformed the other two state-of-the-art models with a margin of 1%. IEMOCAP was also found to perform better compared to DailyDialog in all the three models.

In the second experiment, Bc-LSTM and Transformer based model is chosen to perform emotion classification on EmoContext dataset provided for SemEval2019 Task-3. Bc-LSTM was the most common choice for this task however, in this research, Transformer by Huggingface outperformed Bc-LSTM with a margin of 3% and with less training time.

For empathetic chatbots, the emotion prediction engine needs to be trained on a better engine. The model must also consider user personality. The features of DialogueRNN and the concept of TL-ERC, makes the fusion of DialogRNN and TL-ERC a promising model for emotion prediction and unfolds a new path for further research.

Bibliography

Agrawal, P., & Suri, A. (2019). Nelec at semeval-2019 task 3: Think twice before going deep.

Asghar, N., Poupart, P., Hoey, J., Jiang, X., & Mou, L. (2018). Affective neural response generation. https://doi.org/10.1007/978-3-319-76941-7_12

Baumeister, R., & Leary, M. (1995). The need to belong: Desire for interpersonal attachments as a fundamental human motivation [https://doi.org/10.1037/0033-2909.117.3.497]. *Psychological Bulletin*, *117*(3), 497–529.

Becker-Asano, C., Kopp, S., & Wachsmuth, I. (2007). Why emotions should be integrated into conversational agents. https://doi.org/10.1002/9780470512470.ch3

Brownlee, J. (2017). *What are word embeddings for text?* [https://machinelearningmastery.com/what-are-word-embeddings/].

Cai, H., Chen, H., Zhang, C., Song, Y., Zhao, X., Li, Y., Duan, D., & Yin, D. (2020). Learning from easy to complex: Adaptive multi-curricula learning for neural dialogue generation. *Proceedings of the AAAI Conference on Artificial Intelligence*, *34*, 7472–7479. https://doi.org/10.1609/aaai.v34i05.6244

Chatterjee, A., Narahari, K. N., Joshi, M., & Agrawal, P. (2019). SemEval-2019 task 3: EmoContext contextual emotion detection in text, In *Proceedings of the 13th international workshop on semantic evaluation*, Minneapolis, Minnesota, USA, Association for Computational Linguistics. https://doi.org/10.18653/v1/S19-2005

Clement, J. (2019). *Daily social media usage worldwide 2012-2019* [https://www.statista.com/statistics/433871/daily-social-media-usage-worldwide/].

Clement, J. (2020). *Number of global social network users 2017-2025* [https://www.statista.com/statistics/278414/number-of-worldwide-social-network-users/].

Colombo, P., Witon, W., Modi, A., Kennedy, J., & Kapadia, M. (2019). Affect-driven dialog generation. https://doi.org/10.18653/v1/N19-1374

Danescu-Niculescu-Mizil, C., & Lee, L. (2011a). Chameleons in imagined conversations: A new approach to understanding coordination of linguistic style in dialogs., In

Proceedings of the workshop on cognitive modeling and computational linguistics, acl 2011.

Danescu-Niculescu-Mizil, C., & Lee, L. (2011b). Chameleons in imagined conversations: A new approach to understanding coordination of linguistic style in dialogs., In *Proceedings of the workshop on cognitive modeling and computational linguistics, acl 2011*.

Devlin, J., Chang, M., Lee, K., & Toutanova, K. (2018). BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, *abs/1810.04805*arXiv 1810.04805. http://arxiv.org/abs/1810.04805

Drift, & SurveyMonkey. (2019). *2019 state of conversational marketing* (tech. rep.) [https://www.drift.com/blog/state-of-conversational-marketing/#state-of-convo-marketing]. Drift. https://www.drift.com/blog/state-of-conversational-marketing/#state-of-convo-marketing.

e Gennaro, M., Krumhuber, E., & Lucas, G. (2020). Effectiveness of an empathic chatbot in combating adverse effects of social exclusion on mood [https://doi:10.3389/fpsyg.2019.03061], *10*, 3061.

Ekman, P. (1992). An argument for basic emotions. *Cognition and Emotion*, *6*(3-4), 169–200. https://doi.org/10.1080/02699939208411068

Fung, P., dey, A., Siddique, F., Lin, R., Yang, Y., Wan, Y., & Chan, R. (2016). Zara the supergirl: An empathetic personality recognition system. https://doi.org/10.18653/v1/N16-3018

Glorot, X., Bordes, A., & Bengio, Y. (2010). Deep sparse rectifier neural networks.

Guzman, A. (2017). Making ai safe for humans: A conversation with siri.

Hazarika, D. [D.], Poria, S., Mihalcea, R., Cambria, E., & Zimmermann, R. (2018). Icon: Interactive conversational memory network for multimodal emotion detection. https://doi.org/10.18653/v1/D18-1280

Hazarika, D. [Devamanyu], Poria, S., Zimmermann, R., & Mihalcea, R. (2020). Conversational transfer learning for emotion recognition. *Information Fusion*, *65*. https://doi.org/10.1016/j.inffus.2020.06.005

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, *9*, 1735–80. https://doi.org/10.1162/neco.1997.9.8.1735

Holt, S. (2020). *Natural language processing and text classification with word embeddings* [https://pathtopioneer.com/blog/2020/06/7-1].

Honnibal, I., M. Motani. (2019). *Spacy meets transformers: Fine-tune bert, xlnet and gpt-2* [https://explosion.ai/blog/spacy-transformers].

Howard, J., & Gugger, S. (2020). Fastai: A layered api for deep learning. *Information*, *11*(2), 108. https://doi.org/10.3390/info11020108

Huang, C., Trabelsi, A., & Zaïane, O. (2019). Ana at semeval-2019 task 3: Contextual emotion detection in conversations through hierarchical lstms and bert.

Huang, M., Zhu, X., & Gao, J. (2020). Challenges in building intelligent open-domain dialog systems. *ACM Transactions on Information Systems*, *38*, 1–32. https://doi.org/10.1145/3383123

Huilgol, P. (2019). *Accuracy vs. f1-score.* [https://towardsdatascience.com/fastai-with-transformers-bert-roberta-xlnet-xlm-distilbert-4f41ee18ecb2].

Issa, D., Fatih Demirci, M., & Yazici, A. (2020). Speech emotion recognition with deep convolutional neural networks. *Biomedical Signal Processing and Control*, *59*, 101894. https://doi.org/https://doi.org/10.1016/j.bspc.2020.101894

Karani, D. (2018). *Introduction to word embedding and word2vec* [https://towardsdatascience.com/introduction-to-word-embedding-and-word2vec-652d0c2060fa].

Khoshdeli, M., Cong, R., & Parvin, B. (2017). Detection of nuclei in he stained sections using convolutional neural networks.

Lee, P. V., T. Heydebreck. (2020). Understanding emotions - the science of emotions and techniques for managing them.

Li, Y., Su, H., Shen, X., Li, W., Cao, Z., & Niu, S. (2017). Dailydialog: A manually labeled multi-turn dialogue dataset. *CoRR*, *abs / 1710.03957*arXiv 1710.03957. http://arxiv.org/abs/1710.03957

Liu, D., Li, Y., & Thomas, M. (2017). A roadmap for natural language processing research in information systems. https://doi.org/10.24251/HICSS.2017.132

Lowe, R., Pow, N., Serban, I., & Pineau, J. (2015). The ubuntu dialogue corpus: A large dataset for research in unstructured multi-turn dialogue systems. *CoRR*, *abs/1506.08909*arXiv 1506.08909. http://arxiv.org/abs/1506.08909

Ly, K., Ly, A., & Andersson, G. (2017). A fully automated conversational agent for promoting mental well-being: A pilot rct using mixed methods [https://doi:10.1016/j.invent.2017.10.002]. *Internet Interv.*, *10*, 39–46.

Majumder, N., Poria, S., Hazarika, D., Mihalcea, R., Gelbukh, A., & Cambria, E. (2019). Dialoguernn: An attentive rnn for emotion detection in conversations. *Proceedings of the AAAI Conference on Artificial Intelligence*, *33*, 6818–6825. https://doi.org/10.1609/aaai.v33i01.33016818

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. *ArXiv*, *abs/1310.4546*.

O'Brien, R. (2019). When the inhuman becomes human: An examination of the musical portrayal of the robot in twenty-first century science-fiction cinema through an analysis of the film scores of automata,ex machina, and the machine.

O'Malley, J. (2018). *You and ai: Will we ever become friends with robots?* [https://www.techradar.com/news/you-and-ai-will-we-ever-become-friends-with-robots].

Pan, S., & Yang, Q. (2010). A survey on transfer learning. *Knowledge and Data Engineering, IEEE Transactions on*, *22*, 1345–1359. https://doi.org/10.1109/TKDE.2009.191

Pennington, J., Socher, R., & Manning, C. (2014). GloVe: Global vectors for word representation, In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, Doha, Qatar, Association for Computational Linguistics. https://doi.org/10.3115/v1/D14-1162

Phrasee. (2016). *PARRY: The AI chatbot from 1972* [https://phrasee.co/parry-the-a-i-chatterbot-from-1972/].

Picard, R. W. (1997). Affective computing by rosalind.

Poria, S., Cambria, E., Hazarika, D., Majumder, N., Zadeh, A., & Morency, L.-P. (2017). Context-dependent sentiment analysis in user-generated videos, In *Proceedings of the 55th annual meeting of the association for computational linguistics (volume 1: Long papers)*, Vancouver, Canada, Association for Computational Linguistics. https://doi.org/10.18653/v1/P17-1081

Poria, S., Majumderd, N., Mihalceae, R., & Hovy, E. (2019). Emotion recognition in conversation: Research challenges, datasets, and recent advances. *IEEE Access*, *PP*, 1–1. https://doi.org/10.1109/ACCESS.2019.2929050

Portela, M., & Granell-Canut, C. (2017). A new friend in our smartphone?: Observing interactions with chatbots in the search of emotional engagement. https://doi.org/10.1145/3123818.3123826

Qin, L., Che, W., Li, Y., Ni, M., & Liu, T. (2020). Dcr-net: A deep co-interactive relation network for joint dialog act recognition and sentiment classification. *Proceedings of the AAAI Conference on Artificial Intelligence*, *34*, 8665–8672. https://doi.org/10.1609/aaai.v34i05.6391

Roberti, M. (2019). *Fastai with transformers (bert, roberta, xlnet, xlm, distilbert)* [https://towardsdatascience.com/fastai-with-transformers-bert-roberta-xlnet-xlm-distilbert-4f41ee18ecb2].

S., E., & Pu, P. (2020). Should machines feel or flee emotions? user expectations and concerns about emotionally aware chatbots. *ArXiv*, *abs/2006.13883*.

S.A.I.L. (2004).

Shin, J., Xu, P., Madotto, A., & Fung, P. (2019). *Happybot: Generating empathetic dialogue responses by improving user experience look-ahead*.

Shum, H., He, X., & Li, D. (2018). From eliza to xiaoice: Challenges and opportunities with social chatbots. *Frontiers of Information Technology and Electronic Engineering*, *19*. https://doi.org/10.1631/FITEE.1700826

Smetanin, S. (2019). EmoSense at SemEval-2019 task 3: Bidirectional LSTM network for contextual emotion detection in textual conversations, In *Proceedings of the 13th* *international workshop on semantic evaluation*, Minneapolis, Minnesota, USA, Association for Computational Linguistics. https://doi.org/10.18653/v1/S19-2034

Tallec, M., Antoine, J.-Y., Villaneau, J., & Duhaut, D. (2011). Affective interaction with a companion robot for hospitalized children: A linguistically based model for emotion detection.

Thompson, T., Van Zalk, N., Marshall, C., Sargeant, M., & Stubbs, B. (2019). Social anxiety increases visible anxiety signs during social encounters but does not impair performance [https://doi.org/10.1186/s40359-019-0300-5]. *BMC Psychology*, *7*(24).

Torfi, A., Shirvani, R. A., Keneshloo, Y., Tavvaf, N., & Fox, E. (2020). Natural language processing advancements by deep learning: A survey. *ArXiv*, *abs/2003.01200*.

Tripathi, S. [S.], & Beigi, H. (2018). Multi-modal emotion recognition on iemocap dataset using deep learning. *ArXiv*, *abs/1804.05788*.

Tripathi, S. [Samarth], & Beigi, H. S. M. (2018). Multi-modal emotion recognition on IEMOCAP dataset using deep learning. *CoRR*, *abs/1804.05788*arXiv 1804.05788. http://arxiv.org/abs/1804.05788

Turing, A. M. (1950). I.—computing machinery and intelligence [https://doi.org/10.1093/mind/LIX.236.433]. *Mind*, *59*(236), 433–460.

Vaidyam, A., Wisniewski, H., Halamka, J., Kashavan, M., & Torous, J. (2019). Chatbots and conversational agents in mental health: A review of the psychiatric landscape. [https://doi:10.1177/0706743719828977]. *The Canadian Journal of Psychiatry*, *64*(7), 456–464.

Vand, N. (2019). *What is the difference between affect, emotion, and mood?* [http://www.dbtcentersouthbay.com/what-is-the-difference-between-affect-emotion-and-mood/].

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention is all you need. *ArXiv*, *abs/1706.03762*.

Warwick, K., & Shah, H. (2016). Can machines think? a report on turing test experiments at the royal society [https://doi.org/10.1080/0952813X.2015.1055826]. *Journal of Experimental Theoretical Artificial Intelligence*, *28*(6), 989–1007.

Weizenbaum, J. (1966). Eliza—a computer program for the study of natural language communication between man and machine. *Commun. ACM*, *9*(1), 36–45. https://doi.org/10.1145/365153.365168

Wolf, M., Miller, K., & Grodzinsky, F. (2017). Why we should have seen that coming: Comments on microsoft's tay "experiment," and wider implications. *ACM SIGCAS Computers and Society*, *47*, 54–64. https://doi.org/10.1145/3144592.3144598

Zandie, R. (2020).

Zemčík, T. (2019). A brief history of chatbots. *DEStech Transactions on Computer Science and Engineering*. https://doi.org/10.12783/dtcse/aicae2019/31439

Zhaojiang, L., Peng, X., Genta, I. W., Zihan, L., & Pascale, F. (2019). Caire: An end-to-end empathetic chatbot. *CoRR*, *abs/1907.12108*arXiv 1907.12108. http://arxiv.org/abs/1907.12108

Zhong, P., Wang, D., & Miao, C. (2019). Knowledge-enriched transformer for emotion detection in textual conversations. https://doi.org/10.18653/v1/D19-1016

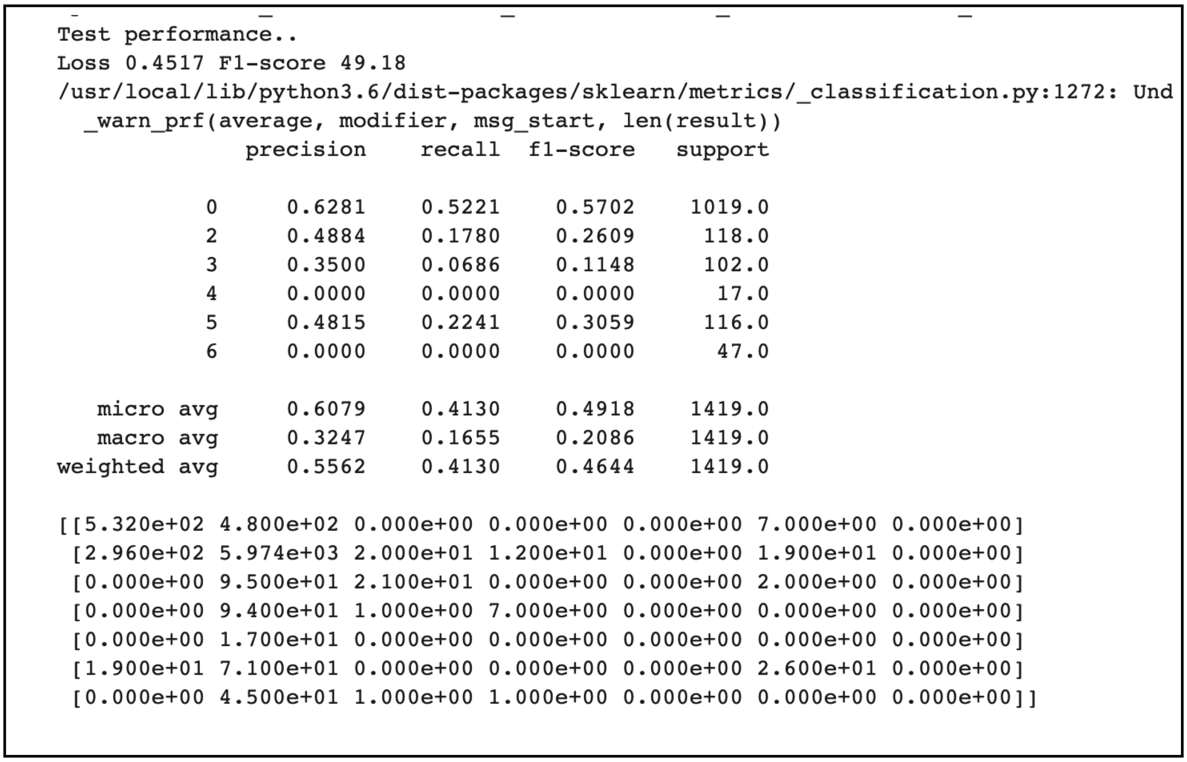
Zhou, H., Huang, M., Zhang, T., Zhu, X., & Liu, B. (2017). Emotional chatting machine: Emotional conversation generation with internal and external memory.

Zhou, L., Gao, J., Li, D., & Shum, H. (2018). The design and implementation of xiaoice, an empathetic social chatbot.

Zhuang, F., Qi, Z., Duan, K., Xi, D., Zhu, Y., Zhu, H., Xiong, H., & He, Q. (2019). A comprehensive survey on transfer learning, arXiv 1911.02685.

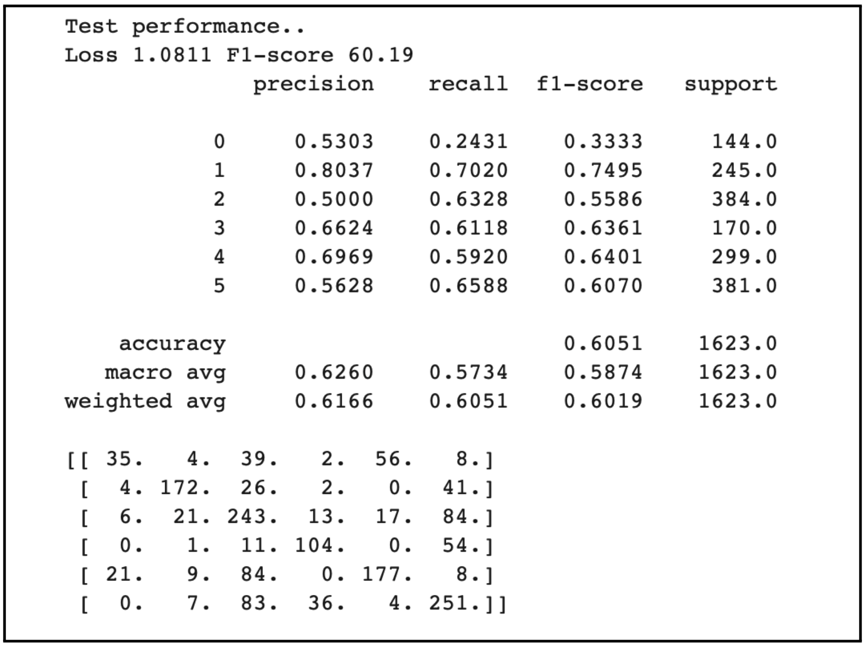
A1 Appendix

A1.1 Performance of Individual Models from Experiment 1



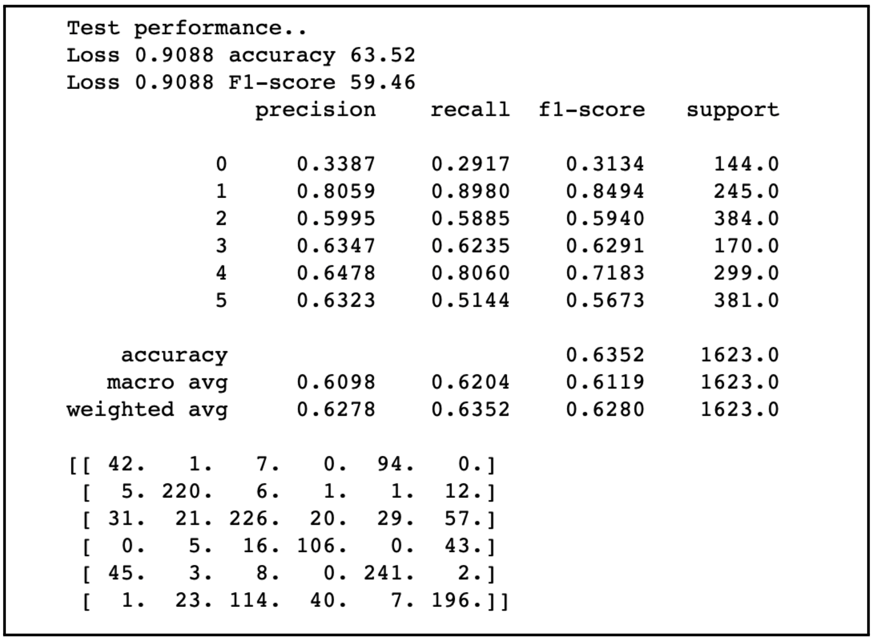
```
Test performance..
Loss 0.4517 F1-score 49.18
/usr/local/lib/python3.6/dist-packages/sklearn/metrics/_classification.py:1272: Und
  _warn_prf(average, modifier, msg_start, len(result))
              precision    recall  f1-score   support

           0     0.6281    0.5221    0.5702    1019.0
           2     0.4884    0.1780    0.2609     118.0
           3     0.3500    0.0686    0.1148     102.0
           4     0.0000    0.0000    0.0000      17.0
           5     0.4815    0.2241    0.3059     116.0
           6     0.0000    0.0000    0.0000      47.0

   micro avg     0.6079    0.4130    0.4918    1419.0
   macro avg     0.3247    0.1655    0.2086    1419.0
weighted avg     0.5562    0.4130    0.4644    1419.0

[[5.320e+02 4.800e+02 0.000e+00 0.000e+00 0.000e+00 7.000e+00 0.000e+00]
 [2.960e+02 5.974e+03 2.000e+01 1.200e+01 0.000e+00 1.900e+01 0.000e+00]
 [0.000e+00 9.500e+01 2.100e+01 0.000e+00 0.000e+00 2.000e+00 0.000e+00]
 [0.000e+00 9.400e+01 1.000e+00 7.000e+00 0.000e+00 0.000e+00 0.000e+00]
 [0.000e+00 1.700e+01 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00]
 [1.900e+01 7.100e+01 0.000e+00 0.000e+00 0.000e+00 2.600e+01 0.000e+00]
 [0.000e+00 4.500e+01 1.000e+00 1.000e+00 0.000e+00 0.000e+00 0.000e+00]]
```

Figure A1.1: Performance of Bc-LSTM with DailyDialog



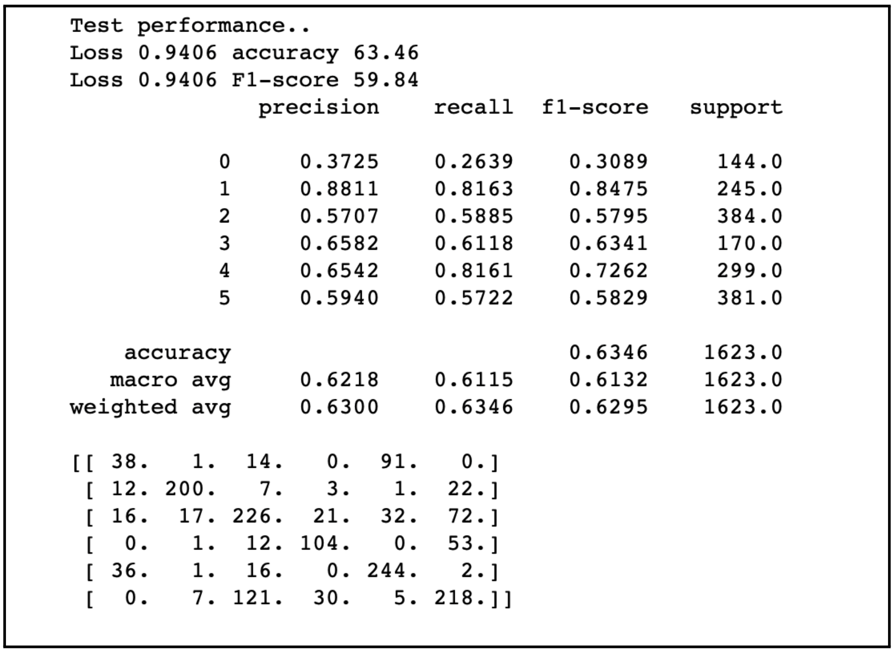
```
Test performance..
Loss 1.0811 F1-score 60.19
              precision    recall  f1-score   support

           0     0.5303    0.2431    0.3333     144.0
           1     0.8037    0.7020    0.7495     245.0
           2     0.5000    0.6328    0.5586     384.0
           3     0.6624    0.6118    0.6361     170.0
           4     0.6969    0.5920    0.6401     299.0
           5     0.5628    0.6588    0.6070     381.0

    accuracy                         0.6051    1623.0
   macro avg     0.6260    0.5734    0.5874    1623.0
weighted avg     0.6166    0.6051    0.6019    1623.0

[[ 35.   4.  39.   2.  56.   8.]
 [  4. 172.  26.   2.   0.  41.]
 [  6.  21. 243.  13.  17.  84.]
 [  0.   1.  11. 104.   0.  54.]
 [ 21.   9.  84.   0. 177.   8.]
 [  0.   7.  83.  36.   4. 251.]]
```

Figure A1.2: Performance of Bc-LSTM with IEMOCAP



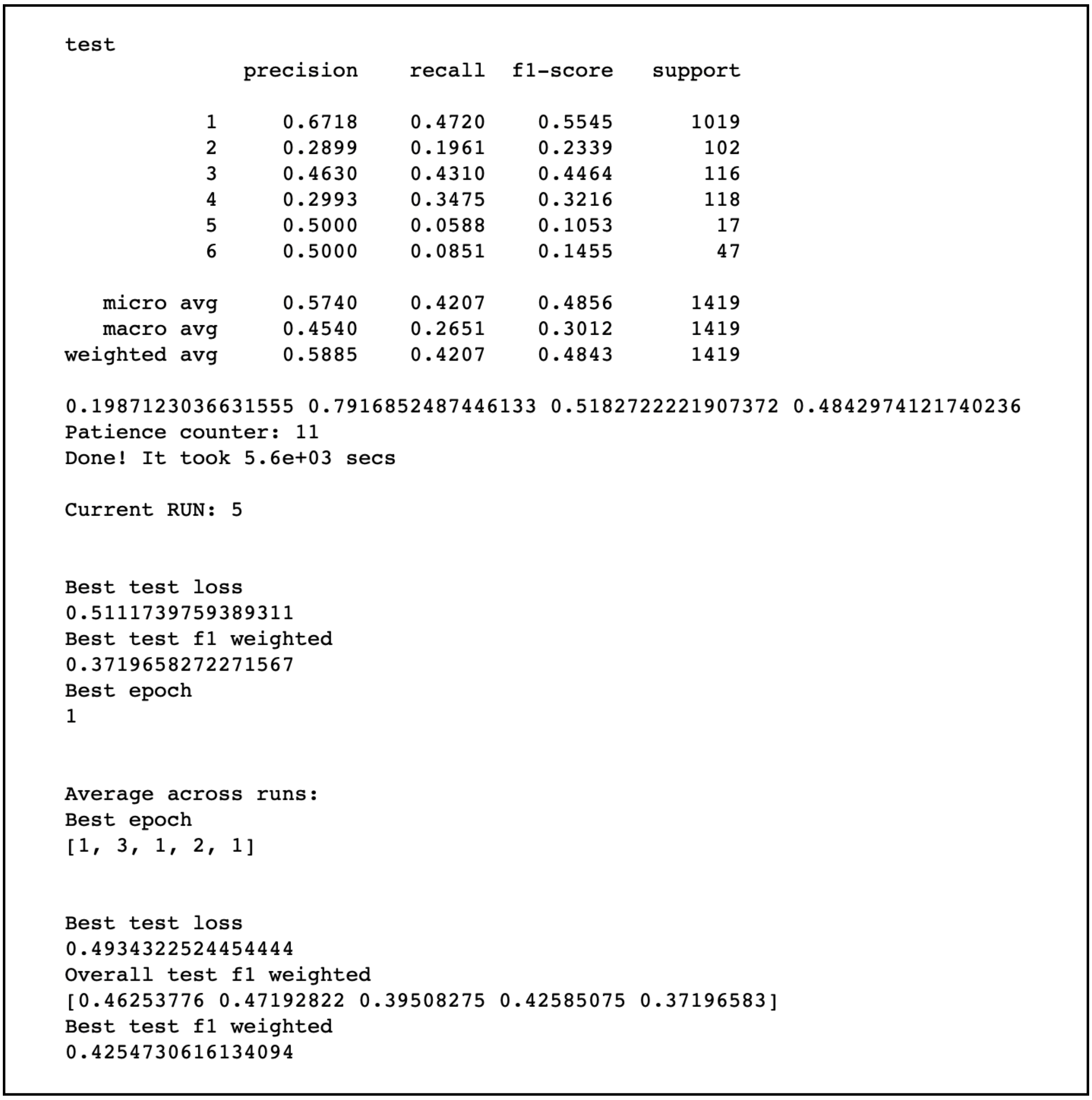
```
Test performance..
Loss 0.9088 accuracy 63.52
Loss 0.9088 F1-score 59.46
              precision    recall  f1-score   support

           0     0.3387    0.2917    0.3134     144.0
           1     0.8059    0.8980    0.8494     245.0
           2     0.5995    0.5885    0.5940     384.0
           3     0.6347    0.6235    0.6291     170.0
           4     0.6478    0.8060    0.7183     299.0
           5     0.6323    0.5144    0.5673     381.0

    accuracy                         0.6352    1623.0
   macro avg     0.6098    0.6204    0.6119    1623.0
weighted avg     0.6278    0.6352    0.6280    1623.0

[[ 42.   1.   7.   0.  94.   0.]
 [  5. 220.   6.   1.   1.  12.]
 [ 31.  21. 226.  20.  29.  57.]
 [  0.   5.  16. 106.   0.  43.]
 [ 45.   3.   8.   0. 241.   2.]
 [  1.  23. 114.  40.   7. 196.]]
```

Figure A1.3: Performance of DialogueRNN with DailyDialog



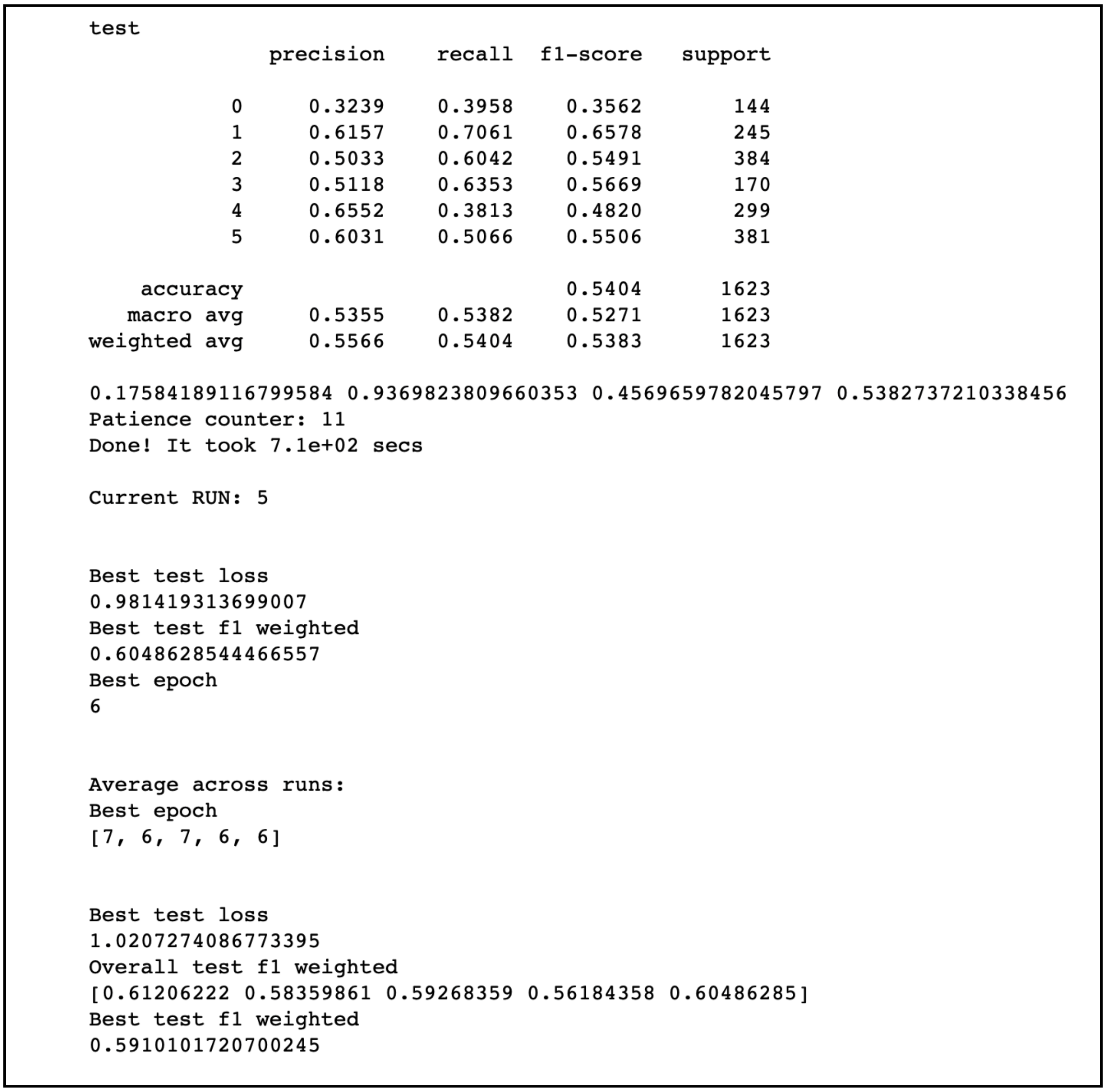
```
Test performance..
Loss 0.9406 accuracy 63.46
Loss 0.9406 F1-score 59.84
              precision    recall  f1-score   support

           0     0.3725    0.2639    0.3089     144.0
           1     0.8811    0.8163    0.8475     245.0
           2     0.5707    0.5885    0.5795     384.0
           3     0.6582    0.6118    0.6341     170.0
           4     0.6542    0.8161    0.7262     299.0
           5     0.5940    0.5722    0.5829     381.0

    accuracy                         0.6346    1623.0
   macro avg     0.6218    0.6115    0.6132    1623.0
weighted avg     0.6300    0.6346    0.6295    1623.0

[[ 38.   1.  14.   0.  91.   0.]
 [ 12. 200.   7.   3.   1.  22.]
 [ 16.  17. 226.  21.  32.  72.]
 [  0.   1.  12. 104.   0.  53.]
 [ 36.   1.  16.   0. 244.   2.]
 [  0.   7. 121.  30.   5. 218.]]
```

Figure A1.4: Performance of DialogueRNN with IEMOCAP



```
test
              precision    recall  f1-score   support

           1     0.6718    0.4720    0.5545      1019
           2     0.2899    0.1961    0.2339       102
           3     0.4630    0.4310    0.4464       116
           4     0.2993    0.3475    0.3216       118
           5     0.5000    0.0588    0.1053        17
           6     0.5000    0.0851    0.1455        47

   micro avg     0.5740    0.4207    0.4856      1419
   macro avg     0.4540    0.2651    0.3012      1419
weighted avg     0.5885    0.4207    0.4843      1419

0.1987123036631555 0.7916852487446133 0.5182722221907372 0.4842974121740236
Patience counter: 11
Done! It took 5.6e+03 secs

Current RUN: 5


Best test loss
0.5111739759389311
Best test f1 weighted
0.3719658272271567
Best epoch
1


Average across runs:
Best epoch
[1, 3, 1, 2, 1]


Best test loss
0.4934322524454444
Overall test f1 weighted
[0.46253776 0.47192822 0.39508275 0.42585075 0.37196583]
Best test f1 weighted
0.4254730616134094
```

Figure A1.5: Performance of TL-ERC with DailyDialog



```
test
              precision    recall  f1-score   support

           0     0.3239    0.3958    0.3562       144
           1     0.6157    0.7061    0.6578       245
           2     0.5033    0.6042    0.5491       384
           3     0.5118    0.6353    0.5669       170
           4     0.6552    0.3813    0.4820       299
           5     0.6031    0.5066    0.5506       381

    accuracy                         0.5404      1623
   macro avg     0.5355    0.5382    0.5271      1623
weighted avg     0.5566    0.5404    0.5383      1623

0.17584189116799584 0.9369823809660353 0.4569659782045797 0.5382737210338456
Patience counter: 11
Done! It took 7.1e+02 secs

Current RUN: 5


Best test loss
0.981419313699007
Best test f1 weighted
0.6048628544466557
Best epoch
6


Average across runs:
Best epoch
[7, 6, 7, 6, 6]


Best test loss
1.0207274086773395
Overall test f1 weighted
[0.61206222 0.58359861 0.59268359 0.56184358 0.60486285]
Best test f1 weighted
0.5910101720700245
```

Figure A1.6: Performance of TL-ERC with IEMOCAP

A2 Appendix

A2.1 Performance graph of Train, Validation and Test data of Transformer Model used for Emotion Prediction for EmoContext dataset

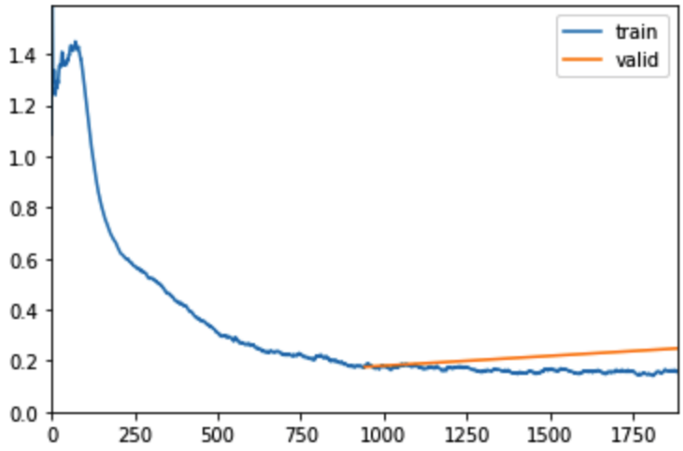


Figure A2.1: Train Data

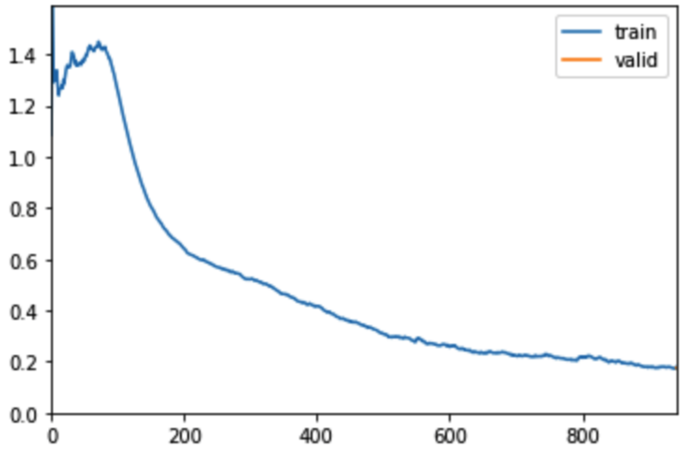


Figure A2.2: Validation Data

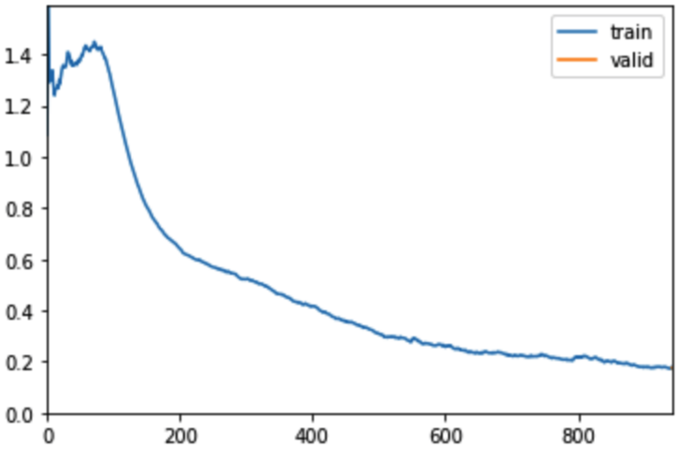


Figure A2.3: Test Data

A3 Appendix

A3.1 Link to Github

https://github.com/sakinavohracs/EmotionDetection