Institut für Informatik, Universität Zürich

**MSc Thesis Report**

# Emotion Recognition in Textual Conversations

Neeraj Kumar

Matrikelnummer: 19-756-303

Email: neeraj.kumar@uzh.ch

August 19, 2021

supervised by
Prof. Dr. Martin Volk and
Dr. Annette Rios

University of Zurich[UZH]

Department of Informatics

(This page intentionally left blank)

# Abstract

Emotion recognition in textual conversations(ERC) is an important natural language processing (NLP) task with applications in different fields, including data mining, e-learning, human–computer interaction, and psychology. Recognizing emotions in textual conversations is a difficult problem to solve due to lack of facial expressions and voice modulations. Different from the traditional non-conversational emotion detection, the model for ERC needs to be context-sensitive (understands the whole conversation rather than individual utterance) and speaker-sensitive (understands which utterance belongs to which speaker) (Li et al., 2020a).

This thesis aims to contribute to research efforts in the field of affective computing and to provide a holistic analysis of text-based emotion recognition with a focus on deep neural network architectures, as deep learning has achieved major breakthroughs and state-of-the-art results for a large number of tasks in the field of Natural Language Processing (Belinkov and Glass, 2019). In this work, we have explored the latest state of art approaches for emotion detection in text and analyzed the underlying techniques and emotional models. Subsequently, we have implemented a hierarchical transformer model for emotion detection purposed by Li et al. (2020b), using Pytorch Lightning Framework[1] which leverages contextual information from the conversation history. It is a transformer-based context- and speaker sensitive model for ERC and consists of two hierarchical transformers. The implementation[2] utilizes a pre-trained BERT model from HuggingFace[3] Transformers library as the lower level transformer to generate local utterance representations, and feed them into another high-level transformer so that utterance representations could be sensitive to the global context of the conversation.

During this work, we have conducted experiments on four dialog emotion datasets, Friends, EmotionPush, EmoryNLP, and Semeval EmoContext. Additionally, we evaluated the model performance on the German translation of benchmarked datasets. Results demonstrate that the hierarchical transformer network emotion model obtains competitive results compared with the state-of-the-art methods and can effectively capture the context and speaker information in textual conversations.

[1] `https://www.pytorchlightning.ai/`

[2] The code is available at https://github.com/neeraj310/Master_Thesis_EA_In_ERC

[3] `https://huggingface.co/transformers/`

# Contents

# 1 Introduction

Development of machines with emotional intelligence has been a long-standing goal of AI. With the ever-increasing infusion of interactive systems in our lives, the need for empathetic machines with emotional intelligence is paramount in intelligent customer service, medical systems, education systems, and other domains. Emotion recognition in textual conversations refers to the task of detecting emotions from utterances in a conversation. As illustrated in Figure 1.1, there may be multiple speakers and utterances in the conversation and an ERC model needs to recognize the emotion label for each utterance from these speakers. Therefore, different from the traditional non-conversational emotion detection, the emotion of an utterance usually depends on the context of the whole conversation.

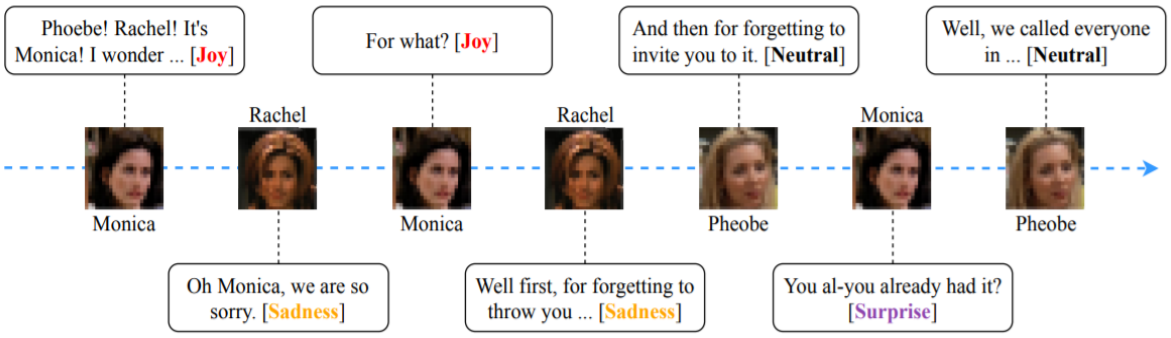


**Figure 1.1:** Emotion Recognition in Textual Conversations (Li et al., 2020a).

Two primary factors drive the emotional dynamics in a conversation: self and interspeaker emotional influence (Morris and Keltner, 2000). Self dependency, also known as emotional inertia, deals with the aspect of influence that speakers have on themselves during conversations. On the other hand, Inter-speaker emotional influence relates to the influences that the counterparts induce into a speaker. Conversely, during the course of a dialogue, speakers also tend to mirror their counterparts to build rapport (Navarretta, 2016).

An example of these two traits in the emotional dynamics of a conversation is shown in Figure 1.2 . Here, $P_a$ is frustrated over her long-term unemployment and seeks encouragement $(u_1; u_3)$. $P_b$, however, is pre-occupied and replies sarcastically $(u_4)$. This enrages $P_a$ to appropriate an angry response $(u_6)$. In this dialog, emotional inertia is evident in $P_b$ who does not deviate from his nonchalant behavior. $P_a$, however, gets emotionally influenced by $P_b$ (Poria et al., 2019).

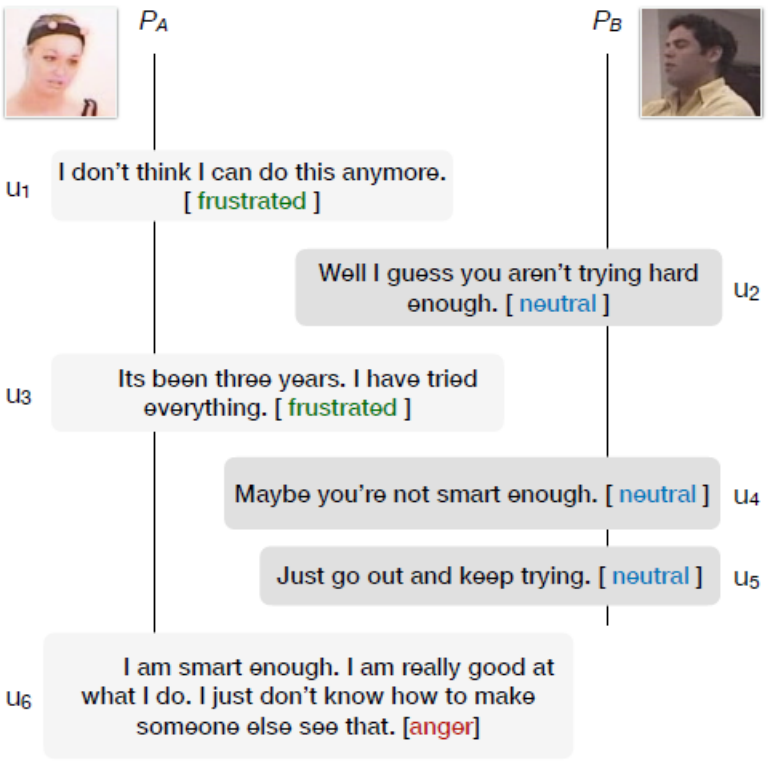


**Figure 1.2:** An example dialogue to show emotional dynamics in conversations (Poria et al., 2019).

Existing works in the literature do not capitalize on these two factors (Hazarika et al., 2018). Context-free systems infer emotions based only on the current utterance in the conversation, whereas state-of-the-art context-based networks use long short-term memory (LSTM) networks (Jiao et al., 2019) to model speaker-based context that suffers from the incapability of long-range summarization (Tang et al., 2018).

In this work, we have implemented 'HiTransformer', a transformer (Vaswani et al., 2017) based context and speaker-sensitive model proposed by Li et al. (2020b). Firstly, we utilize a pre-trained bidirectional transformer encoder BERT (Devlin et al., 2019) to generate local utterance representations. BERT has been shown to be a powerful representation learning model in many NLP applications and can exploit contextual information more efficiently than RNNs and CNNs. Another high-level transformer is used to capture the global context information in conversations. To make our model speaker-sensitive, we introduce speaker embedding into our model. After obtaining the contextual utterance embedding vectors with a hierarchical transformer framework, we feed them into the fully connected layers for classification. Dropout is applied on the fully connected layers to prevent overfitting and softmax layer is used to obtain a probability distribution over the output classes.

In the following section, we present the major challenges that are identified during the

literature review of emotion recognition task and explain how our implementation tackles these challenges.

1. Same utterance can deliver different emotions in different contexts as shown in table 1.1. The importance of context becomes ever more significant in classifying short utterances, like "yeah", "okay", "no", that can express different emotions depending on the context and discourse of the dialogue.

| Speaker | Utterance Text |
|---|---|
| **Monica** | I'm gonna miss you! |
| **Rachel** | I mean it's the end of an era! |
| **Monica** | **I know! (Sadness)** |
| **Chandler** | So, what do you think? |
| **Ross** | I think It's the most beautiful table I've ever seen. |
| **Chandler** | **I know! (Joy)** |
| **Monica** | Now, this is last minute so I want to apologize for the mess. Okay? |
| **Rachel** | Oh my God! It sure didn't look this way when I lived here. |
| **Monica** | **I know! (Surprise)** |

**Table 1.1:** Sample from Friends dataset. The examples demonstrate the importance of context in recognizing the emotion of an utterance (Huang et al., 2019).

2. Long-range contextual information is hard to effectively capture, resulting in missing crucial information that may have appeared many utterances before but has a salient effect on the current emotional state of the speaker.

   HiTransformer addresses the above two problems by using a hierarchical transformer framework, with a lower-level transformer to model the word-level input and an upper-level transformer to capture the context of utterance-level embeddings.

3. Unlike the traditional text classification problems, labeled datasets for the Emotion recognition task are quite small and contain inadequate conversations or speech. Training dataset Friends, EmotionPush and EmoryNLP contain approximately $1000$ dialogues each, which are not large enough for the stability of training a complex neural-based model.

   For handling the data scarcity issue, our implementation makes use of a pretrained language model BERT as the lower-level transformer, which is equivalent to introducing external data into the model and helps our model obtain better utterance

embedding. The use of pretrained langauge model helps in avoiding overfitting on the training data and increases the understanding of informal text.

4. To better model the emotional interaction between speakers, speaker information is necessary.

   Speaker embeddings are concatenated with the utterance embeddings to enable the model to capture the interaction between speakers.

5. The prediction targets (emotion labels) are highly unbalanced. Some emotions are rarely seen in daily-life conversations. For example, people are usually calm and exhibit a neutral emotion while only in some particular situations, they express strong emotions, like anger or fear. Thus we need to be sensitive to the minority emotions while relieving the effect of the majority emotions.

   To alleviate the effect of data imbalance issue, we follow Khosla (2018) to train our models by minimizing a weighted categorical cross-entropy loss.

The organization of this report is described as follows:

1. **Introduction**: In this chapter, we illustrate the organization of this report and introduce the general information about emotion recognition modeling in textual conversations.
2. **Dataset**: In this chapter, we present the datasets used in bench-marking our implementation and provide an exploratory data analysis of some of the important properties.
3. **Implementation**. In this chapter, we present the implementation details and provide an explanation of overall architecture for our deep neural network emotion model.
4. **Evaluation**. In this chapter, we report an evaluation of our implementation.
5. **Insights and Findings**. We demonstrate our findings in this chapter.
6. **Case Studies**. We present some case studies of prediction results in this section.
7. **Conclusions**. This section concludes the work presented in this report.

## 1.1 Emotion Modelling

As shown in Figure 1.3, researchers have proposed two major approaches for emotion modeling.

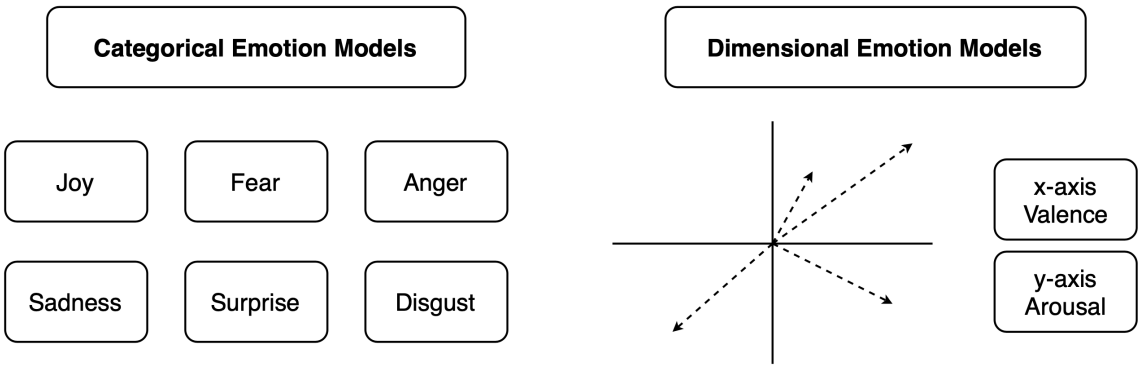


**Figure 1.3:** Emotion Modelling Approaches.

1. **Categorical approach**. This approach is based on the idea that there exist a small set of discrete emotions that are basic and universally recognized with ties to universal facial expressions and physiological processes such as increased heart rate and perspiration (Gunes et al., 2010). Emotion model proposed by Paul Ekman is considered as most widely used in emotion recognition research and involves six basic emotions: happiness, sadness, anger, fear, surprise, and disgust (Ekman, 1993). Another very common set is that of Plutchik in which he added trust and anticipation to Ekman's six emotions (Kołakowska et al., 2015).

2. **Dimensional approach**. This approach views emotion as a numerical score in a 3D space rather than as discrete independent categories (Gunes et al., 2010). This approach covers emotion variability in three dimensions:

    a) **Valence**: This dimension refers to how positive or negative an emotion is.

    b) **Arousal**: This dimension refers to how excited or apathetic an emotion is.

    c) **Power:**: This dimension refers to the degree of power.

In the categorical approach, the emotional states are limited to a fixed number of discrete categories, and it may be difficult to address a complex emotional state or mixed emotions. However, these types of emotions can be well addressed in the dimensional approach, although the reduction in the emotion space to three dimensions is extreme and may result in information loss.

## 1.2 Related Work

Text-based emotion recognition has been a long-standing topic in Natural Language Processing(NLP) research with applications in diverse fields like education, political forecasting, brand marketing, and human-machine interaction. Besides traditional emotion detection from individual sentences, conversational emotion detection has also become a research hotspot and several publicly available datasets have been released (Zahiri and Choi, 2018), (Poria et al., 2019). Emotion-detection for text can be largely classified into two main categories:

1. **Hand-crafted Feature Engineering Based Approaches**: In the early part of research, emotion detection in textual conversations was based on feature engineering such as lexicon and acoustic features (Strapparava and Valitutti, 2004). These approaches can be broadly categorized into following 2 classes.

    a) **Keyword/Corpus Based Approaches**: These methods are based on finding occurrences of keywords in a given text and assign an emotion label based on the detected keyword (Wu et al., 2006), (Balahur et al., 2011). For example, the sentence "Rain always make me feel happy" explicitly expresses happiness and includes the emotion keyword "happy".

        i. These pattern/dictionary based approaches are simple and naive, and are easy to implement.

        ii. These approaches can not handle implicit emotions (emotion analysis based on context) and ambigious words efficiently.

    b) **Learning Based Approaches** : These methods rely on extracting statistical features such as the presence of frequent ngrams, negation, punctuation, emoticons, hashtags to form representations of sentences which are then used as input by classifiers such as Decision Trees, SVMs among others to predict the emotion label (Alm et al., 2005), (Chatterjee et al., 2019).

        i. These methods require extensive feature engineering and they often do not achieve high recall due to diverse ways of representing emotions.

        ii. These approaches result in domain-dependent models (Domains on which feature engineering was performed).

2. **Deep Learning Based Approaches**: Deep Neural Networks have enjoyed considerable success in varied tasks in text, speech, and image domains and have become the most popular choice for NLP in the last few years (Belinkov and Glass, 2019). LSTMs (Hochreiter and Schmidhuber, 1997) and Gated Recurrent Units (GRU) (Chung et al.) which are variants of Recurrent Neural Networks (RNN), and Convolutional Neural Networks (CNN) have shown state-of-the-art results in text classification tasks including sentiment analysis. However, these approaches rely on extracting feature representations for independent text instances without dialouge level context. Therefore, they cannot sufficiently exploit the contextual information in conversations.

    To take advantage of context, recent approaches have proposed to treat ERC task as a sequence labeling problem. Poria et al. (2015) proposed a bidirectional contextual long short-term memory (LSTM) network, termed as bcLSTM, to capture the sequential relationship of utterances. Similarly Jiao et al. (2019) applied bidirectional GRU to model contextual information and placed a self-attention layer in the hidden states of GRU, and fused the attention outputs with the individual utterance embeddings to learn the contextual utterance embeddings. Our implementation follows this line of work but employs a more advanced model, namely the transformer (Vaswani et al., 2017). The transformer learns the dependencies between words based entirely on self-attention without any recurrent or convolutional layers and therefore, can avoid the exploding

gradient or vanishing gradient problem caused by long term dependencies in these deep learning models (Hanin, 2018).

Following are some additional important concepts related to this study.

1. **Transfer Learning**: Transfer learning (Ruder et al., 2019) is a technique whereby knowledge from a source domain is leveraged in order to improve performance in a (possibly different) target domain. It is often used to overcome the constraints of limited training data by allowing models to be trained on generic corpora and subsequently be easily adapted to specific tasks with strong performance. Pretrained language models, such as ELMo (Peters et al., 2018), OpenAI GPT (Alt et al., 2019), and BERT (Devlin et al., 2019), have achieved great success in a variety of NLP tasks, such as sentiment analysis and textual classification. They can generate deep contextualized embeddings since they are pretrained on a massive unlabeled corpus (i.e., EnglishWikipedia). We use a pretrained language model from Hugging Face's Transformers library called DistilBERT (Sanh et al., 2019), as the lower-level transformer for our implementation.

2. **Transformers: State-of-the-Art Natural Language Processing**: Recent progress in natural language processing has been driven by advances in both model architecture and model pretraining (Wolf et al., 2020). The Transformer architecture has proved to be particularly conducive to pretraining on large text corpora, leading to major gains in accuracy on downstream tasks. Transformers is a library dedicated to supporting Transformer-based architectures and facilitating the distribution of pretrained models. It is maintained by the team of engineers and researchers at Hugging Face [1] with support from a vibrant community of external contributors and is available on GitHub[2]. The library supports the distribution and usage of a wide variety of pretrained models in a centralized model hub. This hub supports users to compare different models with the same minimal API and to experiment with shared models on a variety of different tasks.

3. **DistilBERT**: Use of large scale pretrained language models has been on the rise in NLP applications for the last few years. While these models lead to significant improvement, they often have several hundred million parameters as illustrated in Figure 1.4. The trend toward bigger models raises several concerns like the environmental costs associated with computational requirements of these models, cartelization by few corporate groups having the resources to train these models, and inability of edge devices, e.g. mobile phones, to run these models because of computational and memory requirements. DistilBERT is a small, fast, cheap and light Transformer model trained by distilling BERT base. It has $40\%$ less parameters than bert-base-uncased, runs $60\%$ faster while preserving over $95\%$ of BERT's performances as measured on the GLUE language understanding benchmark (Wang et al., 2018). It leverages knowledge distillation during the pretraining phase, a compression technique in which a compact model 'the student' is trained to reproduce the behaviour of a larger model 'the teacher' or an ensemble of models.

[1] `https://huggingface.co/transformers/`
[2] `https://github.com/huggingface/transformers`


**Figure 1.4:** Parameter counts of several recently released pretrained language models (Sanh et al., 2019).

4. **PyTorch Lightning**: PyTorch Lightning is a lightweight wrapper for organizing PyTorch code and easily adding advanced features such as distributed training and 16-bit precision for Machine Learning applications. Lightning manages boilerplate code, such as device optimization, logging, process rank management, and more so that researchers can focus on building the best models possible. Lightning code is easier to manage because engineering code is abstracted away, and common functions such as training steps, process data are standardized.

## 1.3 Task Details

Given the transcript of a conversation along with speaker information of each constituent utterance, the utterance-level dialogue understanding task aims to identify the label of each utterance from a set of pre-defined labels that can be either a set of emotions, dialogue acts, intents etc.

Formally, let there be a set of speakers, $S = \{s_i\}_{i=1}^{M}$, where M is the number of speakers, and a set of emotions, $C = \{c_i\}_{i=1}^{N}$, where N is the number of emotions, such as anger, joy, sadness, and neutral. Assume we are given a set of dialogs, $D = \{D_i\}_{i=1}^{L}$, where L is the number of dialogs. In each dialog, $D_i = \{(u_j, s_j, c_j)\}_{J=1}^{N_i}$ is a sequence of utterances, where the utterance $u_j$ is spoken by speaker $s_j \in S$ with an emotion $c_j \in C$. Our goal is to train a model to find the most likely emotion from C for each new utterance (Li et al., 2020b).

In this work, we limit utterance-level dialogue understanding to only tagging utterances with emotion labels. However, the scope of this research topic includes various other harder

problems that we do not address in this thesis such as slot filling, identifying sarcasm, finding causes of responses, etc.

# 2 Dataset

One of the major challenges in emotion recognition in conversations task is to find a good labeled dataset. However, there are a few standard famous datasets that are used by researchers for this task. In this section, we present the datasets used in benchmarking our implementation and provide an exploratory data analysis of some of the important properties.

1. **EmotionLines** (Hsu and Ku, 2018): is a dialogue dataset consisting of two subsets, Friends and EmotionPush, according to the source of the dialogues. Each subset consists of 1000 English dialogues, with each dialogue containing a variable number of utterances. All the utterances are annotated by five annotators using Amazon Mechanical Turk, a crowd-sourcing platform, and the labeling annotation is based only on the textual content. Annotator votes for one of the seven emotions, namely Ekman's six basic emotions (Ekman, 1993), plus the neutral. If none of the emotions gets more than three votes, the utterance will be marked as "non-neutral".

    a) **Friends**: This dataset is annotated from the scripts of Friends TV sitcom, and each dialogue in the dataset consists of a scene of multiple speakers. This dataset consists of 1000 dialogues, which are split into three parts: 720 for training, 80 for validation, and 200 dialogues for testing. Each utterance is tagged with an emotion label from a set of 8 emotions, anger, joy, sadness, neutral, surprise, disgust, fear, and non-neutral.

    b) **EmotionPush**: The dataset consists of private conversations between friends on Facebook and includes 1000 dialogues, which are split into 720, 80, and 200 dialogues for training, validation, and testing, respectively. Each utterance is tagged with an emotion label from a set of emotions as in the Friends dataset.

2. **EmoryNLP** (Zahiri and Choi, 2018): This dataset is annotated from the Friends TV Scripts as well. However, its size and annotations are different from the Friends dataset. It includes 713 dialogues for training, 99 dialogues for validation, and 85 dialogues for testing.

3. **Semeval EmoContext** (Chatterjee et al., 2019): This dataset includes a training data set of 30160 dialogues, and two evaluation data sets, Test1 and Test2, containing 2755 and 5509 dialogues respectively. In this dataset, each dialogue consists of 3 utterances and given a textual dialogue i.e. an utterance along with two previous turns of context, the goal is to infer the underlying emotion of the 3rd utterance by choosing from four emotion classes - Happy, Sad, Angry and Others. The Training dataset is a .txt file containing 5 columns :

a) **ID**: Contains a unique number to identify each training sample.

b) **Turn 1**: Contains the first turn in the three turn conversation, written by User 1.

c) **Turn 2**: Contains the second turn, which is a reply to the first turn in conversation and is written by User 2.

d) **Turn 3**: Contains the third turn, which is a reply to the second turn in the conversation, which is written by User 1.

e) **Label**: Contains the label of Emotion of Turn 3.

For the first two datasets, Friends and EmotionPush, we follow the setup of Hsu and Ku (2018) and evaluate model performance only on four emotion classes, i.e., anger, joy, sadness, and neutral, whereas for EmoryNLP, we consider all the emotion classes present in the dataset. For Semeval-EmoContext dataset, we follow the setup proposed by Chatterjee et al. (2019), and exclude the majority emotion 'Others' while evaluating the model performance. The statistics of the datasets are summarized in Tables 2.1, 2.2 and 2.3 respectively.

| Dataset | dialogues | Utterances | Emotion | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | Neu | Sad | Anger | Joy | Non-Neu | Disgust | Fear | Surprise |
| Friends | 1000 | 14503 | 6530 | 498 | 759 | 1710 | 2772 | 331 | 246 | 1657 |
| EmotionPush | 1000 | 14742 | 9855 | 514 | 140 | 2100 | 1418 | 106 | 42 | 567 |

**Table 2.1:** Detailed Statistics for Friends and EmotionPush Datasets.

| Dataset | dialogues | Utterances | Emotion | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | Neu | Sad | Anger | Joy | Peaceful | Powerful | Scared |
| EmoryNLP | 897 | 12606 | 3776 | 844 | 1332 | 2755 | 1191 | 1063 | 1645 |

**Table 2.2:** Detailed Statistics for EmoryNLP Dataset.

| Dataset | dialogues | Utterances | Emotion | | | |
|---|---|---|---|---|---|---|
| | | | Others | Sad | Anger | Joy |
| Semeval | 38424 | 115272 | 21963 | 5838 | 5954 | 4669 |

**Table 2.3:** Detailed Statistics for Semeval EmoContext Dataset.

## 2.1 DataSet Analysis

In this section, we provide an exploratory data analysis of some of the important properties of the benchmarked datasets.

1. Figures 2.1, and 2.2, illustrate the label distribution of training and test subsets in the benchmarked datasets. All the datasets are highly imbalanced. Training and Test subsets have a similar distribution in Friends, EmotionPush and EmoryNLP, whereas for

the Semeval EmoContext dataset, test subset becomes ever more imbalanced than the training subset.

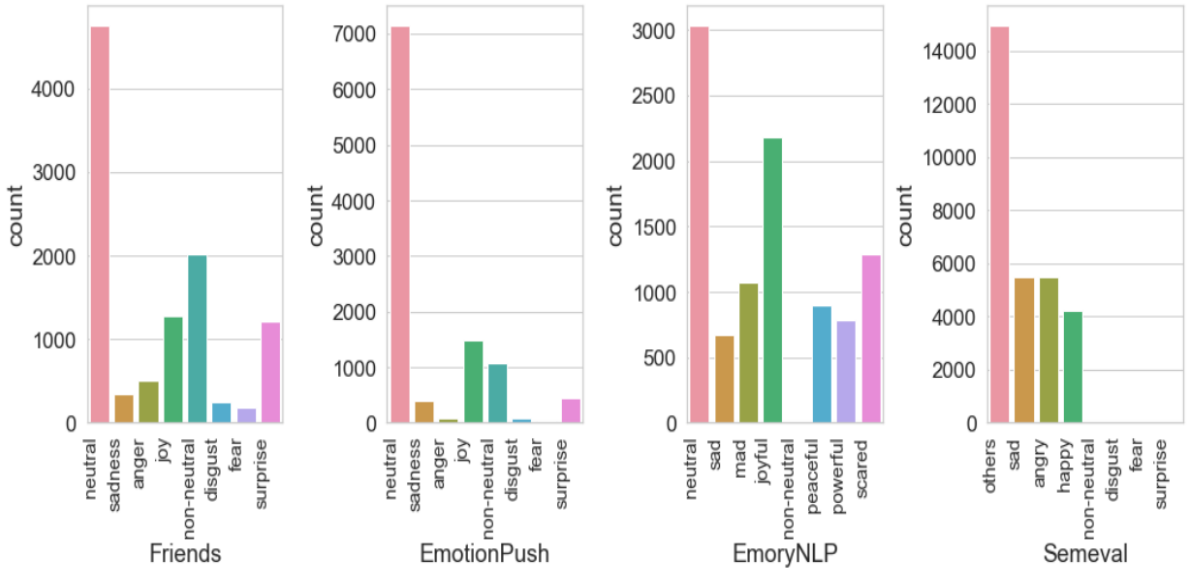


**Figure 2.1:** Training Label Distribution in benchmarked datasets.

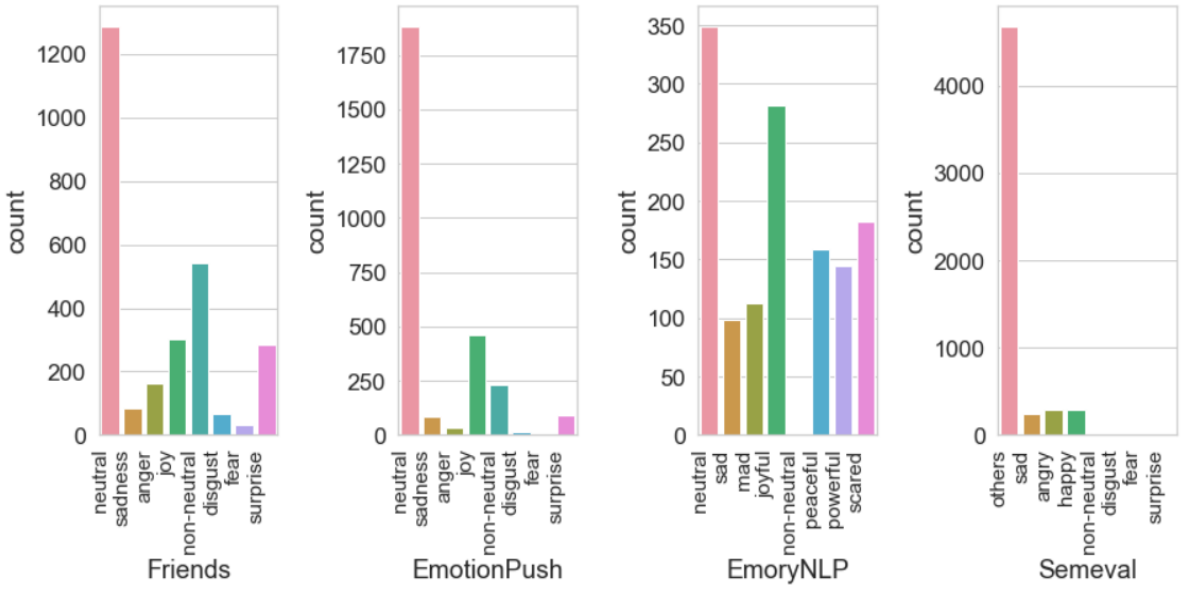


**Figure 2.2:** Test Label Distribution in benchmarked datasets.

2. Table 2.4 provides statistical Comparison among benchmarked Datasets.
    a) Semeval Dataset contains almost 9 times more data than the other 3 datasets.

b) Average utterance length in EmoryNLP dataset is considerably larger than the other 3 datasets.

| Statistics | Friends | | | EmotionPush | | | EmoryNLP | | | Semeval | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | **Train** | **Val** | **Test** | **Train** | **Val** | **Test** | **Train** | **Val** | **Test** | **Train** | **Val** | **Test** |
| # of D | 720 | 80 | 200 | 720 | 80 | 200 | 713 | 99 | 85 | 30160 | 2755 | 5509 |
| # of U | 10561 | 1178 | 2764 | 10733 | 1202 | 2807 | 9934 | 1344 | 1328 | **90480** | **8265** | **16527** |
| Min D-len | 5 | 5 | 5 | 6 | 6 | 5 | 5 | 5 | 5 | 3 | 3 | 3 |
| Max D-len | 24 | 24 | 24 | 24 | 24 | 24 | 25 | 25 | 25 | 3 | 3 | 3 |
| Avg D-len | 14.7 | 14.7 | 13.8 | 14.9 | 15 | 14 | 13.9 | 13.6 | 15.6 | 3 | 3 | 3 |
| Min U-len | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 |
| Max U-len | 69 | 37 | 45 | 133 | 183 | 71 | 148 | 69 | 77 | 145 | 72 | 113 |
| Avg U-len | 7.8 | 7.7 | 8 | 6 | 6.1 | 6.4 | **10.4** | **10** | **10.1** | 5.4 | 5.2 | 5.2 |

**Table 2.4:** Statistical Comparison for benchmarked Datasets (D and U represent dialogue and Utterance correspondingly).

3. Figure 2.3 illustrates dialogue length distribution in benchmarked datasets. For datasets Friends, EmotionPush and EmoryNLP, minimum, maximum, and average dialogue lengths follow similar trends, whereas for Semeval EmoContext dataset, dialogue length is fixed at 3.

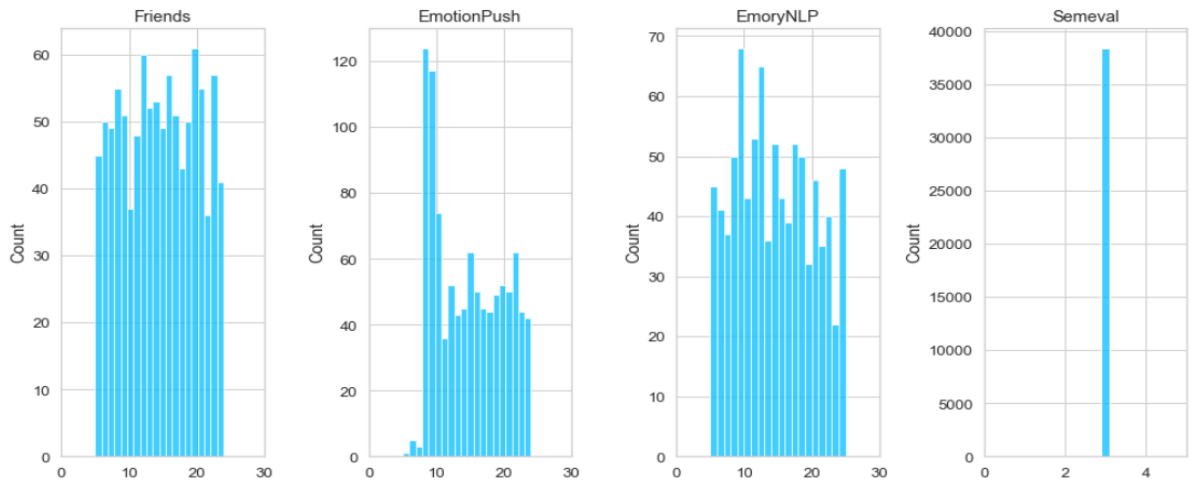


**Figure 2.3:** Dialogue Length (Number of Utterances per dialogue) distribution in benchmarked datasets.

4. Figure 2.4 illustrates utterance length distribution in benchmarked datasets. As expected, the probability of a sentence decreases with an increase in sentence length. This plot provides useful information regarding the selection of hyper-parameter max_seq_len. Sentences shorter than max_seq_len will be padded whereas sentences longer than max_seq_len will be truncated to max_seq_len.

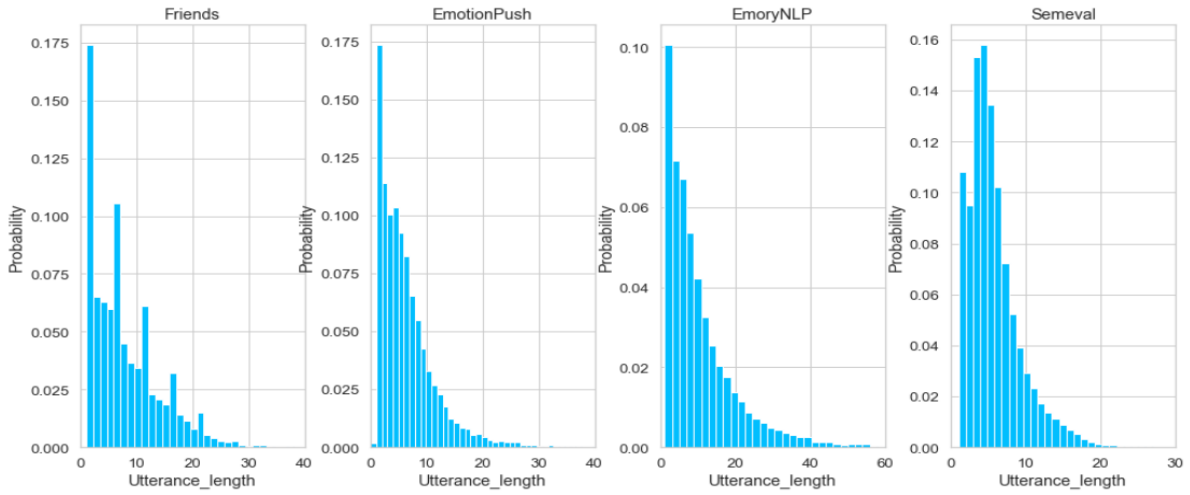


**Figure 2.4:** Utterance Length distribution in benchmarked Datasets.

5. Figure 2.5 illustrates the number of utterances spoken by the top ten speakers in benchmarked datasets. Friends and EmoryNLP are speech-based datasets containing annotated dialogues from the TV sitcom. It means most of the utterances are generated by a few main characters. The personality of a character often affects the way of speaking, and therefore "who is the speaker" might provide extra clues for emotion prediction. EmotionPush does not have this trait due to the anonymous mechanism and Semeval EmoContext dataset does not include speaker information.

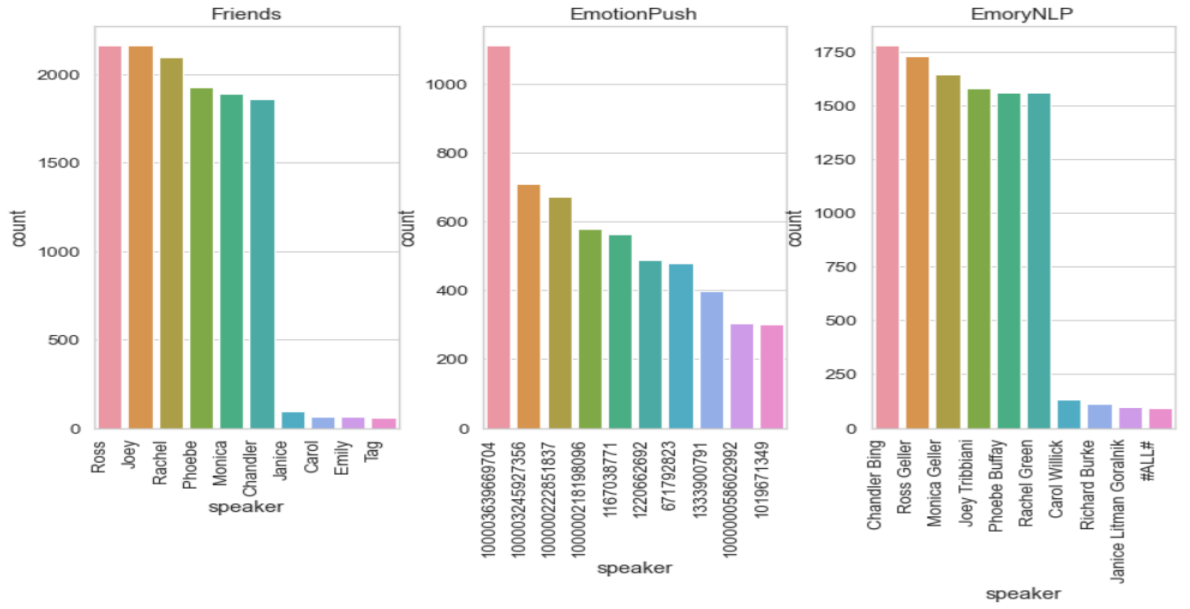


**Figure 2.5:** Speaker Count distribution in benchmarked Datasets.

6. To look for possible patterns in the label sequences of the datasets, we plot the frequency

of label pairs $(x, y)$ where $x$ and $y$ are the labels of $U_{t-1}$ and $U_t$, utterances at $t-1$ and $t$ respectively (Ghosal et al., 2021). Following interpretations can be made from the label transition plots illustrated in Figure 2.6.

a) Plot reveals that for all 4 datasets, the 'Neutral' label appears in consecutive utterances with highest frequency, followed by the same emotion label ($U_t[label] = U_{t-1}[label]$). This property induces label dependencies and consistencies and can cause the deep neural network model to simply learn to copy the previous utterance label without learning any meaningful representation.

b) We were expecting the high frequent joint distribution of similar emotions in consecutive utterances e.g., negative emotions - anger, disgust, sad expressed by one speaker is replied with a similar negative emotion by the other speaker. Interestingly, none of the dataset, elicit any such patterns.

Tables A.1, A.2, A.3, and A.4 provide the transition probabilities matrices corresponding to these plots.

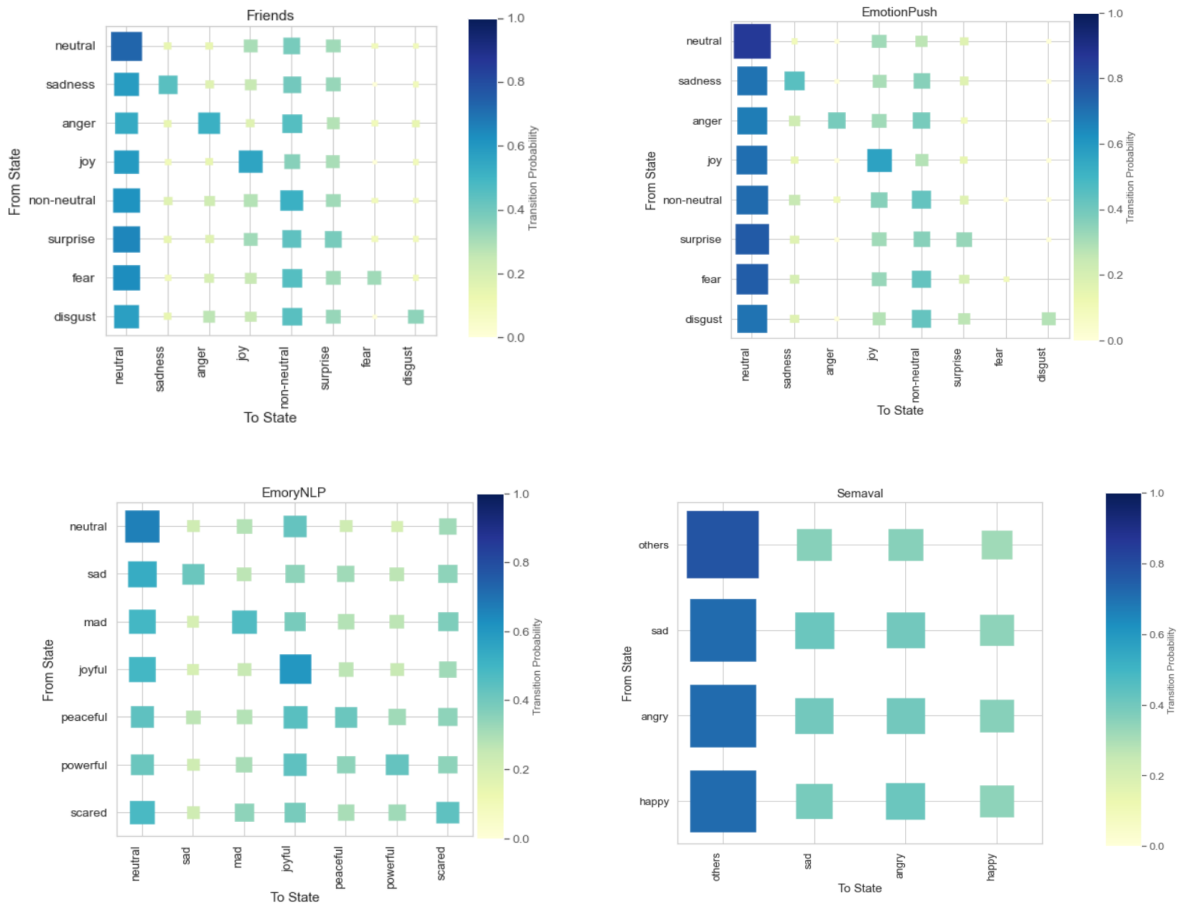


**Figure 2.6:** The heatmap of label transition statistics in the benchmarked datasets. The color bar and square size represent the normalized number of transitions.

# 3 Model Implementation

In this chapter, we describe the implementation details of the Hierarchical Transformer emotion model.

## 3.1 Hierarchical Transformer

Hi-Transformer model proposed by Li et al. (2020b) is a hierarchical transformer framework that consists of a pretrained language model, BERT as the lower-level transformer to model the word-level input and generate the individual local utterance representations. The upper-level transformer further embeds global context information into utterance representations and generates contextual utterance representations. On top of the hierarchical model, a multi-layer perceptron (MLP) is applied to determine the emotion of an utterance based on its representation.

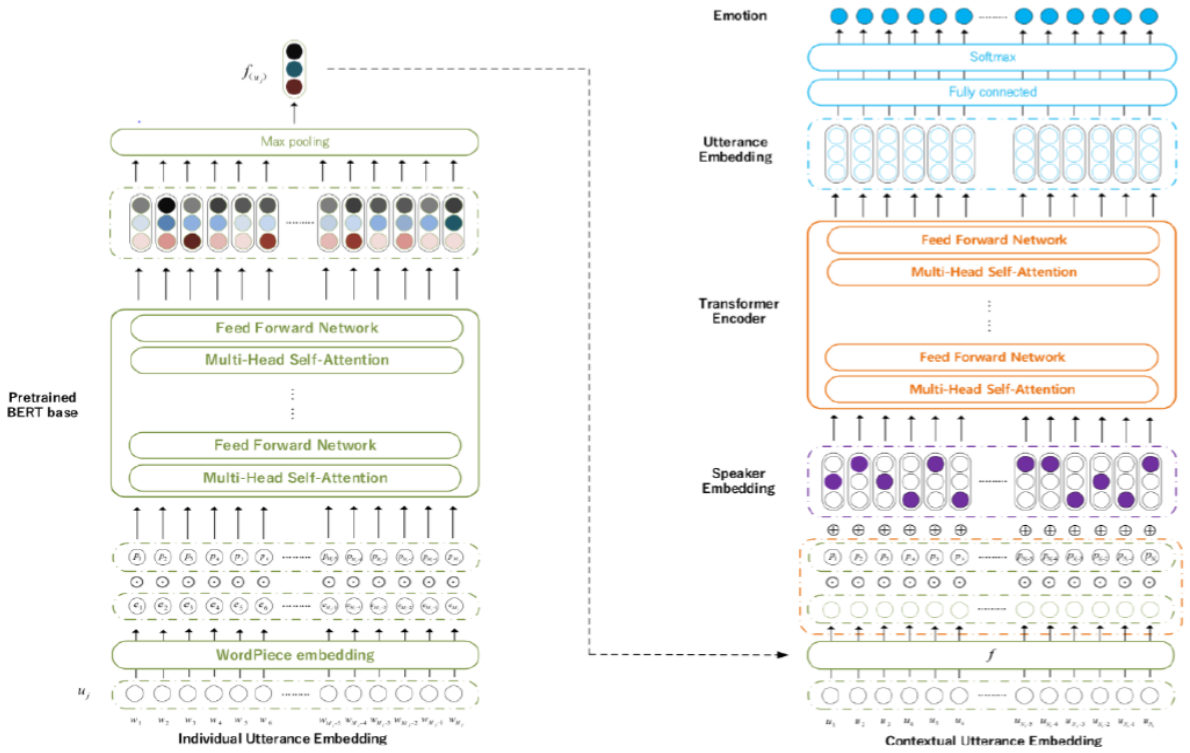


**Figure 3.1:** The architecture of HiTransformer-s emotion recognition model (Li et al., 2020b).

Additionally, the paper proposes a variation in HiTransformer by adding speaker embedding, named HiTransformer-s. To make the model speaker-sensitive, speaker embeddings are concatenated with the utterance embeddings from the output of the lower layer transformer. The overall architecture of the emotion recognition model is illustrated in Figure 3.1.

### 3.1.1 Individual Utterance Embedding

This section explains the creation of sentence embeddings for each standalone utterance. We leverage BERT as the low-level transformer to encode individual utterances. We use a pre-trained language model from Hugging Face Transformers library called DistilBERT (Sanh et al., 2019), as the lower-level transformer for our implementation. DistilBERT is a small, fast, cheap and light transformer model trained by distilling BERT base.

In each dialog, $D_i = \{(u_j, s_j, c_j)\}_{J=1}^{N_i}$ is a sequence of utterances, where the utterance $u_j$ is spoken by speaker $s_j \in S$ with an emotion $c_j \in C$. We use Huggingface tokenizer [1] to perform the prepossessing steps shown in Figure 3.2, on the utterance $u_j = \{w_k\}_{k=1}^{M}$, where $u_j$ is the $j^{th}$ utterance in the dialog $D_i$ and $M_j$ is the number of words in the utterance $u_j$.

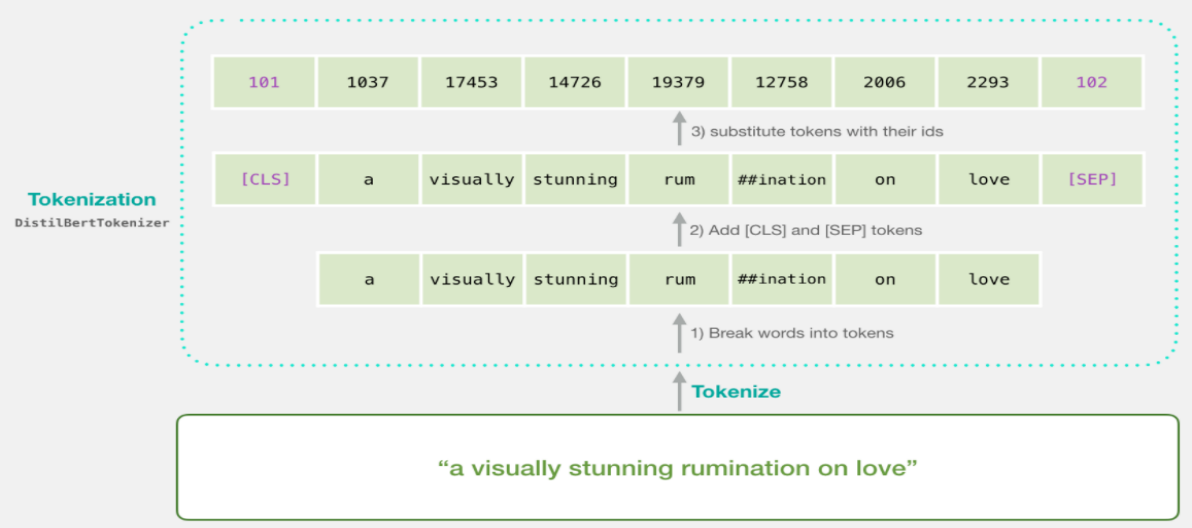


**Figure 3.2:** Tokenization steps to convert sentence text into token embeddings.[2]

1. As we are considering the uncased model, the individual sentences are lowercased first.
2. BERT uses Wordpiece embeddings input for sub-word tokenization (Bostrom and Durrett, 2020). The primary idea behind sub-words is that frequently occurring words should be in the vocabulary, whereas rare words should be split into frequent sub words, i.e. The word "refactoring" can be split into "re", "factor", and "ing". Subwords "re", "factor" and "ing" occur more frequently than the word refactoring, and its overall meaning is also kept intact.

[1] https://huggingface.co/transformers/main_classes/tokenizer.html
[2] https://jalammar.github.io/a-visual-guide-to-using-bert-for-the-first-time/

3. To prepare a batch size of same length, we pad all of our input sequences to be a single fixed length and provide the model with an "attention mask" for each sample, which identifies the [PAD] tokens and tells BERT to ignore them. Individual utterances with lengths longer than hyper-parameter max_seq_len are truncated to be equal to max_seq_len and utterances with lengths shorter than max_seq_len are padded with [PAD] token.

4. Add special tokens needed for sentence classifications ([CLS] at the first position, and [SEP] at the end of the sentence. Replace each token in the sentence with its ids from the WordPiece embeddings and obtain the token embeddings $e = \{(e_k\}_{k=1}^{M_j}$, where $M_j$ is the number of words in utterance $u_j$.

5. Finally, the input embeddings $E = \{(E_k\}_{k=1}^{M_j}$ are the summation of token embeddings e and the positional embeddings $p = \{(p_k\}_{k=1}^{M_j}$, which are obtained using an approach used by BERT.

$$E_k = e_k \odot p_k (k \in [1, M_j])$$

where $\odot$ denotes element wise addition (Li et al., 2020b).

For example, assume that our dataset contains 2000 sentences, and hyper-parameter max_seq_len is selected as 66.

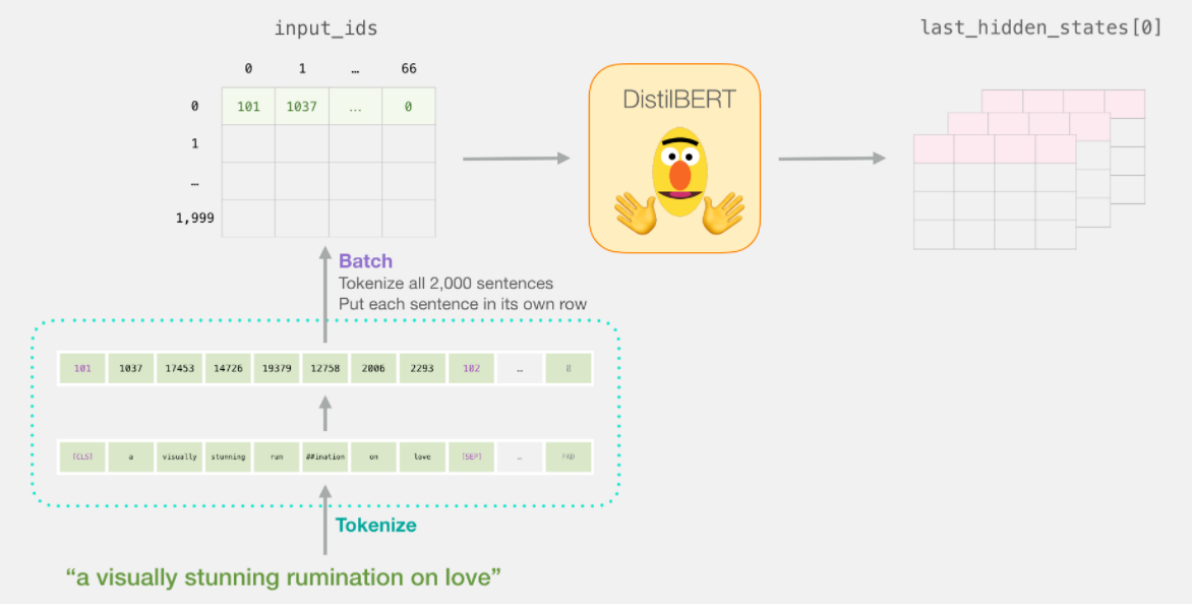


**Figure 3.3:** Sentence Embedding Using DistilBERT.[3]

[3] https://jalammar.github.io/a-visual-guide-to-using-bert-for-the-first-time/

As shown in Figure 3.3, we feed the input embeddings $E$ into the transformer-based pre-trained language model BERT (we use DistilBERT, a distilled version of BERT) to learn the individual utterance embeddings. BERT is designed to pretrain deep bidirectional representations from unlabeled text by jointly conditioning both the left and right contexts in all layers. Bert was designed for giving the model both sentence-level and token-level understanding. The outputs of the last encoder layer corresponding to each input token can be treated as word representations for each token, and the word representation of the first token ([CLS]) can be considered as output representation of the complete sentence, for further finetuning tasks. The output of BERT model $T = BERT(E)$ returns 2 tensors.

1. **last_hidden_state** : Sequence of hidden states at the output of the last layer of the model. It is a tuple with the shape (number of examples $\times$ max number of tokens in the sequence $\times$number of hidden units in the DistilBERT model). In our example, this will be 2000 (since we are assuming 2000 examples in the dataset) $\times$66 (hyper-parameter max_seq_len) $\times$768 (the number of hidden units in the DistilBERT model).

2. **hidden_states** : Tuple of shape (number of examples, max number of tokens in the sequence, number of hidden units in the DistilBERT model) for each one for the 7 hidden layers in DistilBERT model.

The individual utterance embedding is then obtained by concatenating the CLS token (fetched by the first token in the sentence representation) embedding and the mean of the last output layer (Li et al., 2020b).

$$f(u_j) = T[0][:, 0, :] \oplus (mean(T[1][-1], axis = 1))$$

Figure 3.4, shows the slicing operation on the 3d tensor to get the CLS Token embedding tensor.

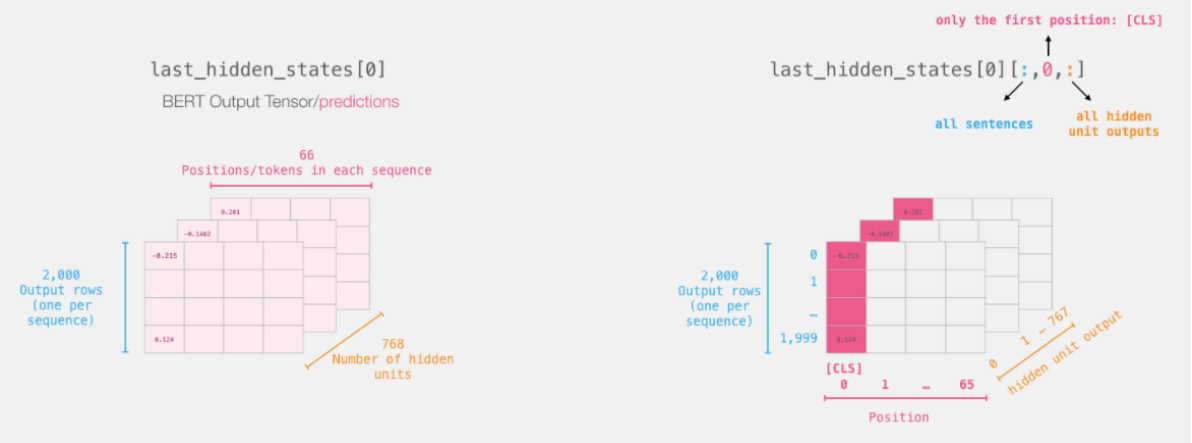


**Figure 3.4:** Slicing BERT's output to fetch [CLS] token.[4]

[4]https://jalammar.github.io/a-visual-guide-to-using-bert-for-the-first-time/

### 3.1.2 Contextual Utterance Embedding

We stack another transformer on top of DistilBERT to capture the global context information in a conversation, as shown in Figure 3.1. For the $i^{th}$ dialog in $D$, $D_i = \{(u_j, s_j, c_j)\}_{j=1}^{N_i}$, the individual utterance embedding is $\{f(u_j)\}_{j=1}^{N_i}$. Following the standard transformer to model the relative positions of utterances in the conversation, we concatenate the individual embeddings with the position embeddings to obtain $U = \{f(u_j) \odot p_j\}_{j=1}^{N_i}$, where $p_j$ is the embedding of position $j$. Then, we apply layer normalization and dropout on $U$ before feeding it into the upper-level transformer to capture the sequential and contextual relationship of utterances in a dialog and obtain the contextual utterance embedding $t = \{t_j)\}_{j=1}^{N_i}$.

$$t = Transformer(Droput(LayerNorm(U)))$$

Then, we feed the contextual utterance embedding vector into the classifier, which consists of a combination of linear layers with activation function and dropout. Finally, we obtain the predicted vector over all emotions with a softmax function.

$$\hat{y_j} = softmax(Dropout(FCNN(t)))$$

where FCNN represents the fully connected neural network.

### 3.1.3 HiTransformer-s: Hierarchical Transformer with Speaker Embeddings

Datasets Friends and EmoryNLP, have two very specific properties. Firstly that more than $80\%$ of the utterances are generated by 6 main characters, including Rachel, Monica, Phoebe, Joey, Chandler and secondly that the personal characteristics of these six characters are very distinct and clear. However, HiTransformer does not capture the relationship between utterances and speakers in a conversation. To solve this problem, Li et al. (2020b) proposes a hierarchical transformer with speaker embeddings (HiTransformer-s), which can model the interaction of speakers in a dialouge.

For the $i^{th}$ dialog in $D$, $D_i = \{(u_j, s_j, c_j)\}_{j=1}^{N_i}$, the individual utterance embedding is $\{f(u_j)\}_{j=1}^{N_i}$ and $N_s(D_i)$ is the number of speakers in $D_i$. We first encode all the speakers in $D_i$ with one-hot encoding and then pad them to the dimension of $Max\{(N_s(D_i)\}_{i=1}^{L}$ with 0 to obtain the speaker embeddings $\{e_s(s_j)\}_{j=1}^{N_i}$.

$$\{e_s(s_j)\}_{j=1}^{N_i} = pad(onehot(\{s_j\}_{j=1}^{N_i}), Max\{N_s(D_j)\}_{j=1}^{L})$$

Finally, we concatenate the summation of the individual utterance embeddings and the embeddings of position with the speaker embeddings of every utterance as the input of the upper-level transformer.

$$U = \{f(u_j) \odot p_j) \oplus e_s(s_j)\}_{j=1}^{N_i}$$

where $\odot$ denotes element addition, and $\oplus$ is the concatenation operator.

# 4 Evaluation

## 4.1 Training Setup

### 4.1.1 Loss Function

To solve the issue of class imbalance, following Khosla (2018) which attains the best performance in the EmotionX shared task (Hsu and Ku, 2018), we apply the weighted balanced warming technique to the loss function by minimizing the weighted categorical cross-entropy loss. The weighted loss referenced from Khosla (2018) helps our model to avoid predicting only on majority emotions.

$$loss = \frac{-1}{\sum_{i=0}^{L} N_i} \sum_{i=1}^{L} \sum_{j=1}^{N_i} w_{c_j} \sum_{c \in C} y_j^c \log_2(\hat{y}_j^c)$$

$$\frac{1}{w_c} = \frac{a_c}{\sum_{i \in C} a_i}$$

where $L$ denotes the number of dialogues in the evaluation dataset, $N_i$ denotes the number of utterances in dailouge $D_i$, $C$ is the emotion class set for evaluation, $y_j^c$ represents the groundtruth label whereas $\hat{y}_j^c$ represents the predicted label for utterance $j$ in emotion class c, and $a_i$ denotes the number of utterances with emotion $i$ in the training set.

For Friends and EmotionPush datasets, we only evaluate the model performance on four emotions: anger, joy, sadness, and neutral, and exclude the contribution of remaining classes by setting their loss weights to zero during training. For the relevant classes, we assign the loss weight $w(c_j)$ inversely proportional to the number of training utterances in the class $c_j$, i.e., assigning larger loss weights for the minority classes to relieve the data imbalance issue. By adding the weighted balanced warming on cross-entropy loss, the model could learn to predict the minor emotions earlier and make the training process more stable.

### 4.1.2 Evaluation Metrics

In our experiments, we use the following evaluation metrics.

1. **Friends/EmotionPush/EmoryNLP** : Evaluation was carried out using the macro-averaged F1 score ($F1_{macro}$) to account for the imbalance in distribution of emotion classes. Metrics definitions are taken from Jiao et al. (2019).

$$P_{Macro} = \frac{\sum_{c \in C} P_c}{|C|}$$

$$R_{Macro} = \frac{\sum_{c \in C} R_c}{|C|}$$

$$F1_{macro} = \frac{\sum_{c \in C} F1_c}{|C|}$$

2. **Semeval EmoContext** Evaluation was carried out using the microaveraged F1 score $(F1_{micro})$ for the three emotion classes - Happy, Sad, and Angry, Neutral class is not considered while calculating the $(F1_{micro})$ score. Micro-average F1 measures the F1-score of the aggregated contributions of all classes. Metrics definitions are taken from Chatterjee et al. (2019).

$$P_{Micro} = \frac{\sum TP_i}{\sum(TP_i+FP_i)} \forall i \in \{Happy, Sad, Angry\}$$

$$R_{Micro} = \frac{\sum TP_i}{\sum(TP_i+FN_i)} \forall i \in \{Happy, Sad, Angry\}$$

$$F1_{micro} = 2.\frac{P_{Micro}.R_{Micro}}{P_{Micro}+R_{Micro}}$$

We also report the weighted accuracy (WA) and unweighted accuracy(UWA). Recognizing strong emotions may provide more value than detecting the neutral emotion, therefore, UWA is a more meaningful evaluation metric for our experimental results as WA is heavily compromised with the large proportion of the neutral emotion. Metrics definitions are taken from Li et al. (2020b).

$$WA = \sum_{c \in C} w_c.a_c$$

$$UWA = \frac{\sum_{c \in C} a_c}{|C|}$$

where $w_c$ is the percentage of class c in the testing set, and $a_c$ is the corresponding accuracy

### 4.1.3 Hyperparameters

We use PyTorch Lightning Framework [1] to implement our model. Hyper-parameter tuning was done using the validation sets of the datasets. The best values are listed in Table 4.1. We adopt the Adam optimizer with the learning rate $2e-5$ and the weight decay rate $0.95$ throughout all the experiments. For datasets Friends, EmotionPush and EmoryNLP, we process 1 dialogue per batch to restrict the context to dialogue boundaries. Since we use the "distilbert-base-uncased" version [2] as our low-level transformer, all the settings are the same with DistilBERT. For the high-level transformer, we set the hidden size, the number of self-attention heads, the feed-forward size and the number of layers as 768, 1, 768 and 2 respectively. For the classification layer, the internal hidden size of the classification layer is set to 768, and the dropout rate is 0.5 to prevent overfitting. Early stopping with a patience of 5 is adopted to terminate training based on the accuracy of the validation set.

[1] `https://www.pytorchlightning.ai/`

[2] `https://huggingface.co/transformers/model_doc/distilbert.html`

| Dataset | m | b | lr | w | Low-Level Transformer | | | | High-Level Transformer | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | h | hd | ff | l | h | hd | ff | l |
| Friends | 40 | 1 | 2e -5 | 0.95 | 768 | 12 | 3072 | 12 | 768 | 1 | 768 | 2 |
| EmotionPush | 35 | 1 | 2e -5 | 0.95 | 768 | 12 | 3072 | 12 | 768 | 1 | 768 | 2 |
| EmoryNLP | 30 | 1 | 2e -5 | 0.95 | 768 | 12 | 3072 | 12 | 768 | 1 | 768 | 2 |
| Semeval | 25 | 32 | 2e -5 | 0.95 | 768 | 12 | 3072 | 12 | 768 | 1 | 768 | 2 |

**Table 4.1:** Hyper-parameter settings. m: max_seq_len, b: batch size, lr: learning rate, w: weight decay rate, h: hidden size, hd: the number of self-attention heads, ff : feed-forward size, l: the number of layers.

# 4.2 Experiments

## 4.2.1 BaseLines

We compare our model with the following state of the art baseline models:

- `bcLSTM`$_+$ (Wenxiang Jiao et al., 2019): A model that uses a combination of 1-D CNN to extract individual utterance embeddings and bidirectional LSTM to model the relationship of utterances.
- `bcGRU` (Wenxiang Jiao et al., 2019): A variant of bcLSTM$_+$ that used a Bidirectional GRU to capture the global context of a conversation.
- `PT-CoDE`$_{mid}$ (Wenxiang Jiao et al., 2019): A variant of CoDE$_{mid}$ that pretrains a context-dependent encoder (CoDE) for ERC by learning from unlabeled conversation data.
- `HiGRU` (Jiao et al., 2019): A hierarchical gated recurrent unit (HiGRU) framework with a lower-level GRU to model the individual utterance embeddings and an upper-level GRU to capture the overall context of the conversation.
- `HiGRU-f` (Jiao et al., 2019): A variant of HiGRU with individual feature fusion.
- `HiGRU-sf` (Jiao et al., 2019): A variant of HiGRU with self-attention and feature fusion.

## 4.2.2 Result Evaluation

Table 4.2 shows the empirical results of the test sets for the three datasets, Friends, EmotionPush, and EmoryNLP. As deep learning models are known to yield results with high variance across multiple training runs, we trained each model 10 times and report the average results with standard deviation. As this work is primarily focused on emotion recognition in textual conversations, we benchmark our model performance for dialogue based datasets only, including Friends, EmotionPush and EmoryNLP. Please refer to tables A.5 and A.6 for the test results for all 10 trials for all 4 datasets.

| Model | Friends | | | EmotionPush | | | EmoryNLP | | |
|---|---|---|---|---|---|---|---|---|---|
| | **Mac-F1** | **WA** | **UWA** | **Mac-F1** | **WA** | **UWA** | **Mac-F1** | **WA** | **UWA** |
| bcLSTM$_+$ | 63.1 | 79.9 | 63.3 | 60.3 | 84.8 | 57.9 | 25.5 | 33.5 | 27.6 |
| bcGRU | 62.4 | 77.6 | 66.1 | 60.5 | 84.6 | 56.9 | 26.1 | 33.1 | 27.4 |
| PT_CoDE$_{mid}$ | 65.9 | 81.3 | 66.8 | 62.6 | 84.7 | 60.4 | 29.1 | 36.1 | 30.3 |
| HiGRU | - | 74.4 | 67.2 | - | 73.0 | 66.9 | - | - | - |
| HiGRU-f | - | 71.3 | 68.4 | - | 73.8 | 66.3 | - | - | - |
| HiGRU-sf | - | 74.0 | 68.9 | - | 73.0 | 68.1 | - | - | - |
| HiTransformer[1] | 66.66 | 82.11 | 63.71 | 63.90 | 86.87 | 61.55 | 31.36 | 37.25 | 29.24 |
| HiTransformer-s[1] | 67.88 | 82.18 | 68.78 | 65.43 | 86.92 | 63.03 | 33.04 | 37.98 | 32.67 |
| HiTransformer[2] | $64.78_{\pm 0.66}$ | $77.71_{\pm 0.54}$ | $69.46_{\pm 0.78}$ | $61.25_{\pm 0.69}$ | $77.85_{\pm 0.91}$ | $67.52_{\pm 1.03}$ | $30.78_{\pm 0.51}$ | $33.02_{\pm 1.08}$ | $31.63_{\pm 0.99}$ |
| HiTransformer-s[2] | $64.79_{\pm 0.69}$ | $72.59_{\pm 0.61}$ | $71.79_{\pm 0.84}$ | $62.13_{\pm 0.81}$ | $79.53_{\pm 1.04}$ | $72.41_{\pm 1.12}$ | $\mathbf{32.43}_{\pm 0.62}$ | $\mathbf{33.58}_{\pm 0.92}$ | $\mathbf{31.94}_{\pm 0.96}$ |
| HiTransformer[3] | $66.83_{\pm 0.82}$ | $82.34_{\pm 0.59}$ | $64.25_{\pm 0.88}$ | $63.0_{\pm 0.76}$ | $86.79_{\pm 1.45}$ | $59.72_{\pm 1.37}$ | $26.16_{\pm 0.62}$ | $41.34_{\pm 1.19}$ | $26.62_{\pm 1.37}$ |
| HiTransformer-s[3] | $\mathbf{68.11}_{\pm 0.86}$ | $\mathbf{82.46}_{\pm 0.66}$ | $\mathbf{64.32}_{\pm 0.65}$ | $\mathbf{66.52}_{\pm 0.98}$ | $\mathbf{87.26}_{\pm 0.56}$ | $\mathbf{63.46}_{\pm 1.36}$ | $26.65_{\pm 0.71}$ | $40.01_{\pm 0.86}$ | $28.11_{\pm 0.57}$ |

[1] Reference Paper implementation without weighted balanced warming support (Li et al., 2020b).
[2] Our implementation with weighted balanced warming support.
[3] Our implementation without weighted balanced warming support.

**Table 4.2:** Test results comparison with baseline models for Friends, EmotionPush, and EmoryNLP datasets. HiTransformer-s refers to performance score with speaker embedding support.

### Comparisons with Previous SOTA Models

As seen from table 4.2, our implementation achieves competitive results compared to the previous state-of-the-art systems in terms of F1 score.

In HiGRU (Jiao et al., 2019), weighted balanced warming(WBW) (*Section* 4.1.1) is applied to the loss function to handle the class imbalance issue. This forces the model to learn the minor emotion with similar accuracy as the major emotion and results in a relatively poorer $F1_{macro}$ score. This effect can be seen in the performance numbers reported for the HiGRU, HiGRU-f and HiGRU-sf models (Jiao et al., 2019). Weighted accuracy(WA) and unweighted accuracy(UWA) numbers are closer to each other, and $F1_{macro}$ score is not reported by the authors of this paper. On the other hand, HiTransformer (Reference model for our implementation) does not use WBW, allowing the model to focus on the majority emotion at the cost of minority emotion. This results in a much larger difference between WA and UWA as the model make a disproportionately small amount of errors on the majority emotion, but a relatively better $F1_{macro}$ score. We report numbers for both approaches in our implementation.

For dataset Friends and EmotionPush, our model achieves the best performance for configuration HiTransformer-s with weight balanced warming support disabled (Losses for all classes are weighted equally), whereas, for EmoryNLP, configuration HiTransformer-s with weight balanced warming support enabled (Losses for minority class are given more weightage in comparison to the majority class), results in the best performance.

Table 4.3 summarizes the deviation in $F1_{macro}$ score of our implementation in comparison to the reference model. Our implementation obtains an absolute improvement of $+0.33\%$ and $+1.66\%$ in terms of $F_{macro}$ score for datasets Friends and EmotionPush. However, for the dataset EmoryNLP, our implementation score is $1.85\%$ lower.

Moreover, we show the performance of each emotion category for the benchmarked datasets in Figure 4.1. For datasets Friends, EmotionPush and EmoryNLP, our model achieves lower performance on emotion classes 'Sadness' and 'Anger' as compared to class 'Happy', which may be due to imbalanced data as the number of training samples for the latter are much higher in comparison to other two emotion classes, as well as the presence of explicit words in the utterance implying the 'Happy' emotion.

| **Model** | **Friends** | | | **EmotionPush** | | | **EmoryNLP** | | |
|---|---|---|---|---|---|---|---|---|---|
| | **Mac-F1** | **WA** | **UWA** | **Mac-F1** | **WA** | **UWA** | **Mac-F1** | **WA** | **UWA** |
| Model-1[1] | $+0.25\%$ | $+0.28\%$ | $+0.84\%$ | $-1.40\%$ | $-0.09\%$ | $-2.90\%$ | $-1.85\%$ | $-11.33\%$ | $+8.17\%$ |
| Model-2[2] | $+0.33\%$ | $+0.34\%$ | $-6.40\%$ | $+1.66\%$ | $+0.39\%$ | $+0.68\%$ | $-1.85\%$ | $-13.17\%$ | $-2.20\%$ |

[1] HiTransformer (Speaker embedding support disabled)

[2] HiTransformer-s (Speaker embedding support enabled)

**Table 4.3:** Summarization of test results against reference model.

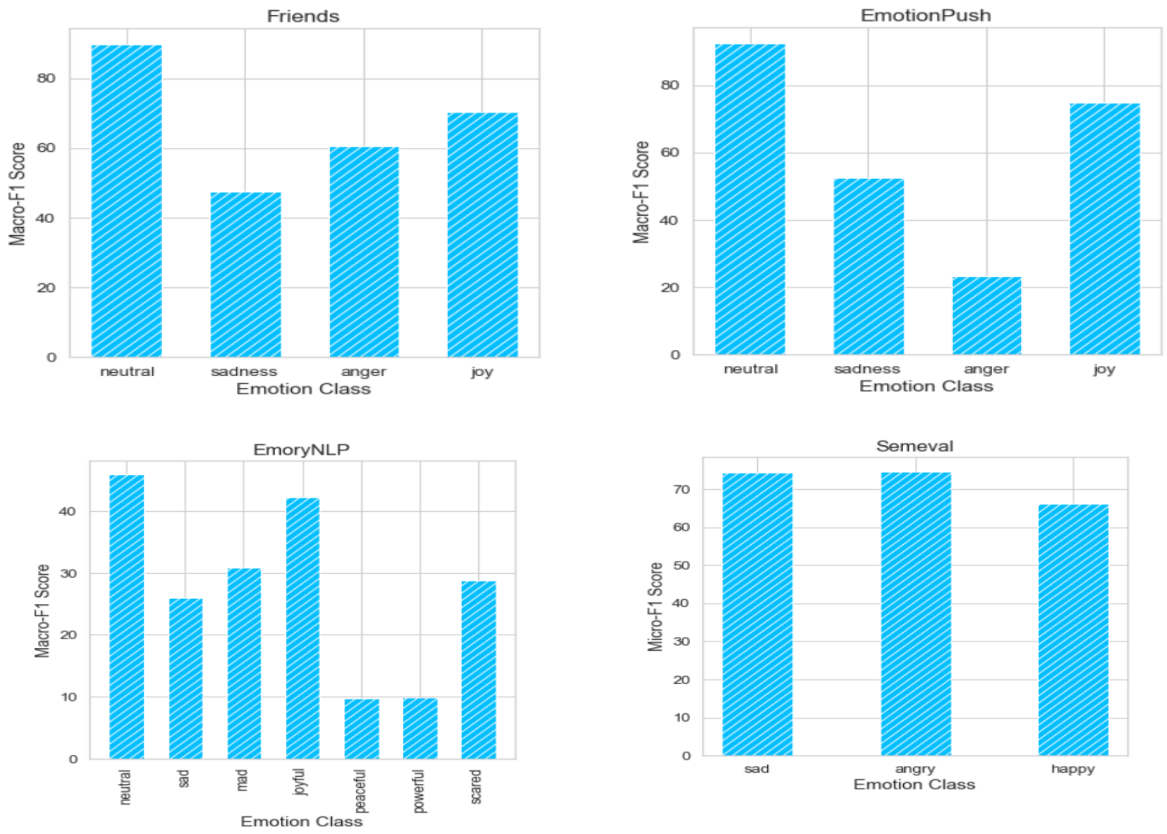


**Figure 4.1:** Performance score of each emotion category for benchmarked datasets.

### 4.2.3 Result Evaluation on German Translation

This section describes the test results for the German translation of benchmarked datasets, Friends, Emotion-Push, and EmoryNLP. We use DeepL-Translator[3] to get the german translation of datasets. We tried two pretrained language models from HuggingFace for German text encoding. For each dataset, we use the configuration that gave the best results on the English version, HiTransformer-s with weight balanced warming support disabled for Friends and EmotionPush and HiTransformer-s with weight balanced warming support enabled for EmoryNLP. Table 4.4 provides the deviation in test results for the German version of benchmarked datasets in comparison to the English version.

We can draw the following conclusions from this table:

1. In terms of $F1_{macro}$ score, there is a performance drop in test results of all data sets. This is on expected lines because of information loss in the translation of conversation text from English to German.

2. Performance drop for dataset EmoryNLP is greater than Friends, as average utterance length in EmoryNLP testset is around $25\%$ (as illustrated in point 2 of section 2.1) higher

[3] `https://www.deepl.com/en/home`

| LM | Friends | | | EmotionPush | | | EmoryNLP | | |
|---|---|---|---|---|---|---|---|---|---|
| | **Mac-F1** | **WA** | **UWA** | **Mac-F1** | **WA** | **UWA** | **Mac-F1** | **WA** | **UWA** |
| LM-1[1] | $-2.59\%$ | $-0.73\%$ | $-2.12\%$ | $-6.47\%$ | $-0.21\%$ | $-8.04\%$ | $-3.39\%$ | $+4.61\%$ | $-5.94\%$ |
| LM-2 [2] | $-6.78\%$ | $-3.32\%$ | $-4.12\%$ | $-15.84\%$ | $-0.79\%$ | $-23.72\%$ | $-9.52\%$ | $+3.84\%$ | $-5.84\%$ |

[1] Language Model: dbmdz/bert-base-german-uncased

[2] Language Model: distilbert-base-german-case

**Table 4.4:** Deviation in test results on German text in comparison to English version of benchmarked datasets (LM demotes pretrained language model from HuggingFace for German text encoding.)

than Friends resulting in a higher information loss during translation from German to English. Performance drop is highest for dataset Emotionpush, as it consists of highly informal text conversations between friends on Facebook resulting in higher translation loss.

3. For all datasets, we get better performance with pretrained language model dbmdz/bert-base-german-uncased than distilbert-base-german-case.

# 5 Insights and Findings

In this section, we set up different experimental studies to understand the results of emotion recognition by our implementation and get deeper insights about utterance-level dialogue understanding.

## 5.1 Error Analysis

In this task, we examine the quality of our model prediction by visualizing the confusion matrix. Figure 5.1 illustrates the confusion matrix for benchmarked datasets. For each dataset, we only visualize the relevant emotions.

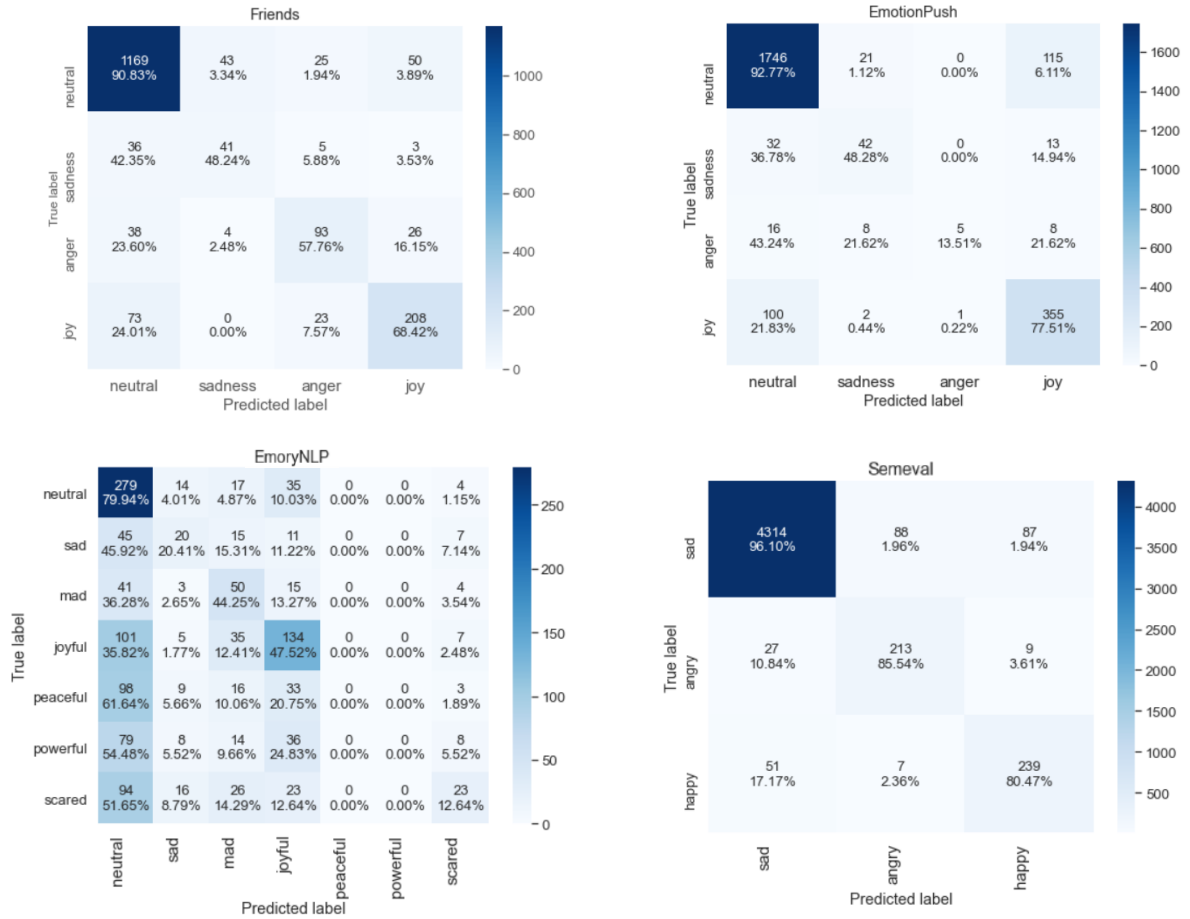


**Figure 5.1:** Confusion Matrix Visualization for Benchmarked Datasets.

For percentage statistics, we calculate the number predicted by our model as a percentage of true number for that class, i.e. for Friends dataset in the $1^{st}$ row, $2^{nd}$ column, number $3.34\%$ implies that $3.34\%$ (43) of 'Neutal' class labels (1287) were falsely classified with emotion class 'sadness'. We can make the following observations from the confusion matrix plots.

1. In all four datsets, all of the emotions get confused mostly with Neutral. This is on expected lines as 'Neutral' is the most dominant class in all four training sets.

2. A high percentage of emotion class 'Anger' gets falsely classified as 'Joy' in all datasets. This may be caused by the model inability to understand sarcasm. Due to the figurative nature of text, nuances and implicit meanings, detecting sarcasm is highly challenging.

3. In Semeval EmoContext dataset, $17\%$ of emotion class 'Happy' is falsely identified as 'Sad', however this trend is not visible across tje other 3 datasets. Larger context lengths in these datasets helps the model in these settings, whereas in Semeval EmoContext dataset, context length is restricted to 3.

## 5.2 Influence of Utterance Positions on the classifier prediction

In this setting, we intend to examine whether there exists a general or dataset specific trend between the test f1-score and position of the utterances (Ghosal et al., 2021). We specifically want to analyze if utterances present at the beginning of the dialogue are relatively easier to classify as compared to the utterances at the middle or the end of the dialogue. Fig. 5.2 illustrates the trend of classification performance against utterances' position for datasets Friends, EmotionPush and EmoryNLP. This study does not include dataset Semeval EmoContext as it consists of only 3 utterances per dialogue. We can draw the following inferences from the below figure.

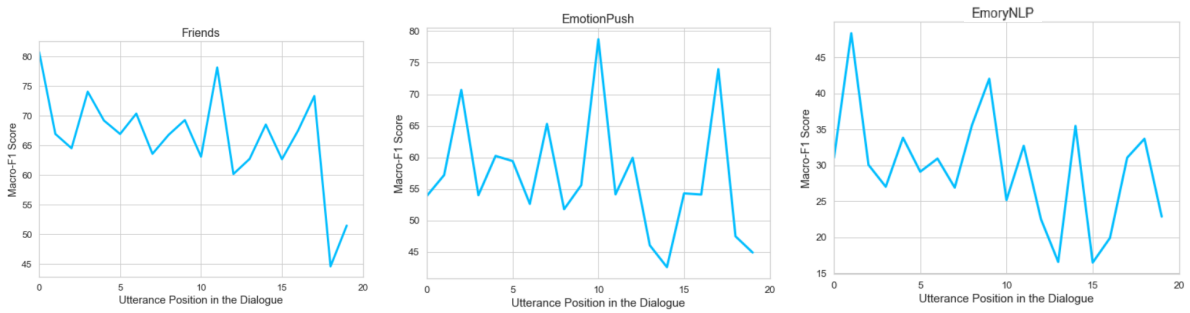


**Figure 5.2:** Impact of utterances' position on the the classifier $\text{F1}_{macro}$ score

1. For datasets Friends and EmoryNLP, there is a decreasing trend of performance w.r.t position of the utterance. At the beginning of the dialogue, utterances are more self-dependent and less context dependent. However, as the conversation progresses, ut-

terances tend to become more dependent on the context and have lesser self-sufficient information to be classified independently.

2. Another possible reason for this decreasing trend could be the lack of training data. There are only a few dialogues with a large number of utterances.

3. For dataset EmotionPush, there is no underlying trend between model performance and utterance position.

## 5.3 Effect of the Speaker, Dialogue Length

In this section, we investigate the effect of the number of speakers and dialogue length on ERC model performance (Li et al., 2020a). Figures 5.3 and 5.4, illustrate the $F1_{macro}$ score trend as the number of speakers and the number of utterances in the dialogue increases. We can draw the following conclusions from these figures:

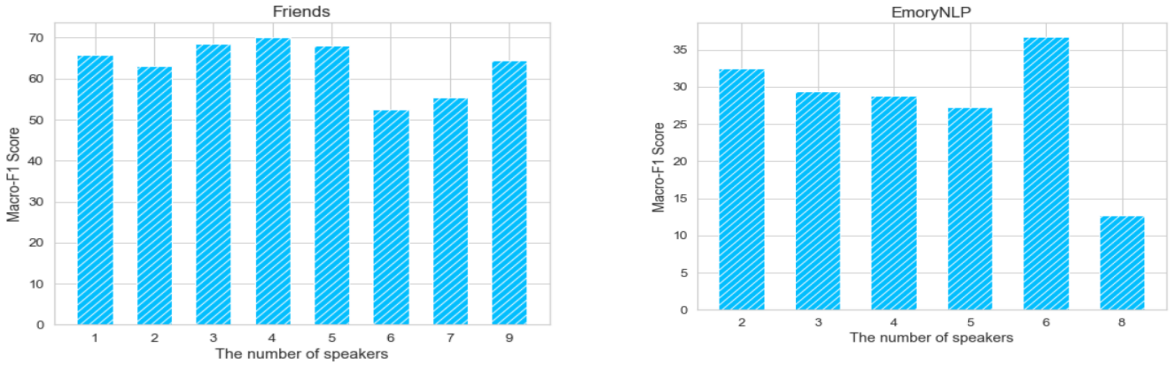


**Figure 5.3:** Effect of speaker count on the classifier $F1_{macro}$ score.

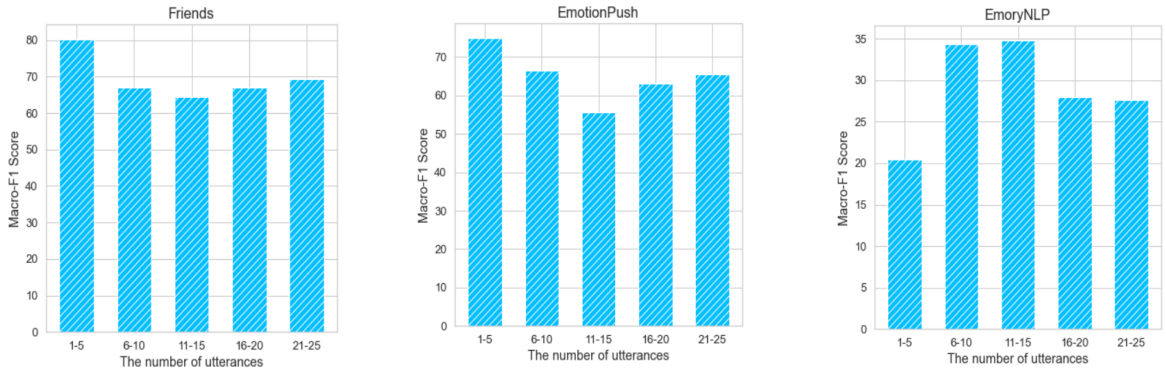


**Figure 5.4:** Effect of dialogue length (Number of utterances in the dialogue) on the classifier $F1_{macro}$ score.

1. Impact of speaker count on the classifier $F1_{macro}$ score:

a) For EmoryNLP dataset, as the number of speakers increases, the $F1_{macro}$ generally decreases, which demonstrates that it becomes increasingly more difficult for the model to recognize emotions as the number of speakers in the conversation increases. This observation is consistent with the human intuition that greater the numbers of speakers in a dialogue, harder it becomes to detect emotion class.

b) However, this trend is not visible in the Friends dataset. It my be caused by the fact that we consider only four emotion classes for $F1_{macro}$ calculation.

c) As the number of speakers per dialogue is fixed at 2, this experiment is not valid for EmotionPush dataset.

2. Impact of utterance count on the classifier $F1_{macro}$ score:

a) For Friends and EmotionPush datasets, initially, the $F1_{macro}$ decreases as the number of utterances in a dialogue goes up, which demonstrates that it is more difficult for the model to detect emotions correctly as the number of utterances in a conversation increases. However, this trend reverses around the median and performance of the model increases as more context is available for the model to make the prediction.

b) However, for EmoryNLP dataset, this trend is completely reversed. For this dataset, our model makes more errors as the number of utterances in a conversation increases. One possible reason for this trend could be the lack of training data. There are only a few dialogues with a large number of utterances and the model has to recognize 7 different emotions in comparison to 4 distinct emotions for the above two datasets.

## 5.4 Classification in Shuffled Context

To understand the importance of context in ERC modeling, we tried to randomly shuffle the order of utterance in a dialog and analyze model performance on shuffled utterance sequence (Ghosal et al., 2021). In ERC modelling, the most crucial contextual information for classifying an utterance comes from the neighbouring utterances. In case of shuffled context, the model capability to predict the correct emotion would go down, as the original neighboring utterances may not be in immediate context after shuffling. This kind of perturbation would make the context modeling less efficient. For example, a dialogue having utterance sequence of $\{u_1; u_2; u_3; u_4\}$ is randomly shuffled to $\{u_4; u_1; u_3; u_2\}$. We use the following shuffling approaches:

1. Dialogues in train and validation sets are shuffled, dialogues in test set are kept unchanged.

2. Dialogues in test set are shuffled, whereas dialogues in train and validation sets set are kept unchanged.

3. Dialogues in all three train, validation and test sets are shuffled.

Experiment results summarized in table 5.1, show that any perturbation in train, validation or test set causes the performance of the model to drop a few points in datasets Friends and EmoryNLP. However, for dataset EmotionPush, model performance goes up for all 3 shuffling strategies. This experiment is not valid for Semeval EmoContext dataset as context is limited to 2 utterances per dialogue. Notably, the highest performance drop is seen for $2^{nd}$ shuffling approach where dialogues in train and validation sets are kept unchanged and dialogues in test set are shuffled.

| **Context Shuffling Strategy** | | | **Friends** | | **EmotionPush** | | **EmoryNLP** | |
|---|---|---|---|---|---|---|---|---|
| Train | Val | Test | $\text{F1}_{macro}$ | $\text{F1}_{micro}$ | $\text{F1}_{macro}$ | $\text{F1}_{micro}$ | $\text{F1}_{macro}$ | $\text{F1}_{micro}$ |
| ✓ | ✓ | ✗ | $-1.68\%$ | $-2.01\%$ | $+2.87\%$ | $+1.59\%$ | $-1.84\%$ | $-1.62\%$ |
| ✗ | ✗ | ✓ | $\mathbf{-5.03\%}$ | $\mathbf{-1.43\%}$ | $+2.82\%$ | $+0.89\%$ | $\mathbf{-3.74\%}$ | $\mathbf{-1.40\%}$ |
| ✓ | ✓ | ✓ | $-4.23\%$ | $-2.26\%$ | $+0.95\%$ | $+1.26\%$ | $-2.67\%$ | $-1.91\%$ |

**Table 5.1:** Deviation in model performance for various shuffling strategies. In Train, Val and Test columns ✓denotes shuffled context and ✗denotes unchanged context.

## 5.5 Controlled Context Dropping

In this section, we try to examine the impact of context length on classification performance. To understand this effect, we design an experimental study with controlled context dropping (Ghosal et al., 2021). In the default setting of HiTransformer, the context of full dialogue is available to the model. We tried a number of different experimental settings, in which we vary and limit the contextual information that is available to the model and study how the results are affected by it. For each dialogue, we control the contextual information available to the model in the following ways:

1. Treat each conversation as an individual sentence without considering the context. In this setting, each dialogue contains only 1 utterance.

2. Realign dialogue boundaries to limit the context to a maximum of 5 utterances per dialogue. In this setting, each dialogue contains 5 utterances.

3. While classifying a target utterance from speaker A, we drop the utterances of all other speakers from the dialogue. In this setting, each dialogue contains utterances from a single speaker.

We report the results for controlled context dropping experiments in Table 5.2. We observe a dip in performance for all three datasets. However, long distance context is much more important for emotion detection in EmoryNLP as compared to the other two datasets. Notably, the highest performance drop is seen for the $1^{st}$ approach which treat each conversation as an individual sentence without considering the context.

| Context Dropping Strategy | Friends | | EmotionPush | | EmoryNLP | |
|---|---|---|---|---|---|---|
| | $F1_{macro}$ | $F1_{micro}$ | $F1_{macro}$ | $F1_{micro}$ | $F1_{macro}$ | $F1_{micro}$ |
| One utterance per dialogue | **−2.75%** | **−1.84%** | **−2.94%** | **−0.42%** | **−10.05%** | **−4.69%** |
| Five utterances per dialogue | −1.19% | −1.05% | −0.34% | −0.19% | −5.62% | +4.69% |
| Remove Interspeaker Context | −0.61% | −0.47% | −0.28% | −0.51% | −6.54% | +3.06% |

**Table 5.2:** Deviation in model performance for various context dropping strategies.

## 5.6 Performance for Label and sentiment Shift

### Label Shift

As illustrated in point 6 of section 2.1 , benchmarked datasets used for this work, exhibit the label copying property which means that there is a very high probability for consecutive utterances to have the same emotion label. Contextual models can end up learning trivial representations to predict either the majority emotion or emotion label of the previous utterance. To understand this phenomenon in more detail, we examine the following two different kinds of shifts that could happen in the course of a dialogue:

1. **Intra-Speaker Shift**: The label of the utterance is different from the label of the previous utterance from the same speaker. Please refer to Fig. 5.5.

2. **Inter-Speaker Shift**: The label of the utterance is different from the label of the previous utterance from the non-target speaker. Please refer to Fig. 5.5.

### Sentiment Shift

In this section, we analyze the results for sentiment shift in intra- and inter-speaker level and examine the deviation in model performance whenever there is a change in sentiment. We divide the emotion classes into three broad categories:

1. Positive sentiment group with emotions like surprise, excitement, happy, peaceful.

2. Negative sentiment group with emotions like sad, angry, disgust.

3. Neutral sentiment group with emotion Neutral.

We are interested to see how our model performs at the utterances were the label or sentiment shift takes place. Table 5.3 summarizes the deviation in model performance corresponding to label and sentiment shift. For datasets Friends and EmotionPush, we see an improvement in $F1_{macro}$ score for all 4 settings, confirming the intuition that our model is indeed learning useful representations rather than just predicting either the majority emotion or emotion label of the previous utterance.

| Experiment Setup | Mode | Friends | | EmotionPush | | EmoryNLP | |
|---|---|---|---|---|---|---|---|
| | | $F1_{macro}$ | $F1_{micro}$ | $F1_{macro}$ | $F1_{micro}$ | $F1_{macro}$ | $F1_{micro}$ |
| Intra-Speaker Shift | Emotion Shift | +4.29% | −5.14% | +8.54% | −4.87% | −5.65% | −12.96% |
| Inter-Speaker Shift | Emotion Shift | +1.60% | −5.14% | +4.91% | −7.69% | −4.38% | −8.33% |
| Intra-Speaker Shift | Sentiment Shift | +6.71% | −6.22% | +7.97% | −5.58% | −4.29% | −13.03% |
| Inter-Speaker Shift | Sentiment Shift | +2.19% | −4.76% | +4.24% | −8.59% | −7.89% | −10.26% |

**Table 5.3:** Deviation in model performance for utterances which exhibits Emotion and Sentiment Shift in test data.

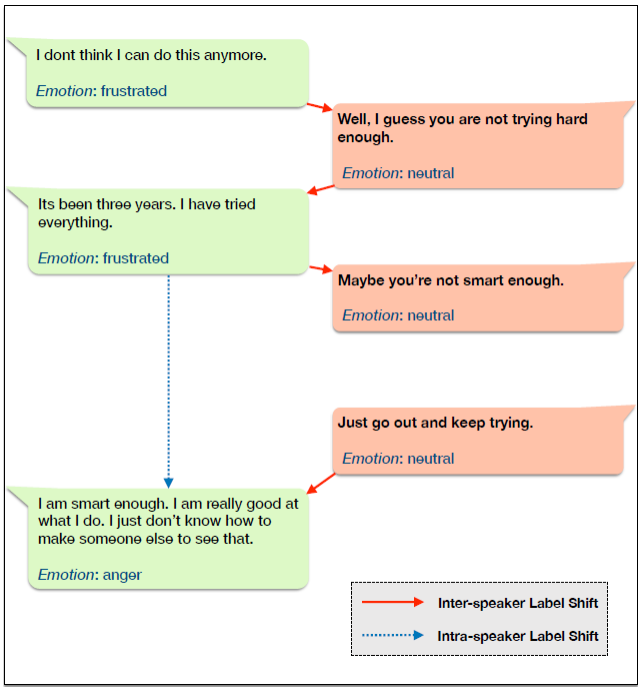


**Figure 5.5:** Example of Inter-speaker and Intra-speaker label shift (Ghosal et al., 2021).

# 6 Case Studies

We did a further investigation into a number of successful and failed predictions, to better understand the results of our model.

## 6.1 Successful Cases

As shown in table 6.1, we investigate three scenes related to the word "Yeah" that expresses three distinct emotions. All three scenes come from the testing subset of dataset Friends.

1. **Scene1**: In Scene-1, 'Yeah' with 'full stop' usually exhibits an absence of emotion and our model correctly classifies the utterance's emotion as 'Neutral'.

2. **Scene2**: In Scene-2, 'Yeah' with "!" expresses strong emotion and our model correctly recognizes it as 'Joy'.

3. **Scene3**: In Scene-3, 'Yeah' with 'full stop' now indicates "Sadness" and is correctly predicted even though in the training set, 'Yeah' with 'full stop' is labelled with emotion 'Neutral' more than $90\%$ of times. These results demonstrate that Hi-Transformer learns from both individual utterances and as well as the context while making the prediction.

| Dialogue | Speaker | Utterance | Ground Truth | Prediction |
|---|---|---|---|---|
| **Scene1** | | | | |
| 11 | Monica | Hey, Joey, could you pass the cheese? | Neu | Neu |
| **11** | **Joey** | **Yeah.** | **Neu** | **Neu** |
| 11 | Joey | Listen uh, I'd prefer it if you didn't call me Joey. | Neu | Neu |
| **Scene2** | | | | |
| 144 | Bonnie | Because I think about shaving it all off again sometime. | Neu | Neu |
| 144 | Rachel | Really?! | Surprise | - |
| **144** | **Bonnie** | **Yeah!** | **Joy** | **Joy** |
| **Scene3** | | | | |
| 141 | Monica | Ohh, sweetie! Hey, I bet you anything that he's gonna call you again. | Non-Neu | - |
| 141 | Rachel | Yeah, maybe, but I don't think I even care. | Sadness | Sadness |
| 141 | Rachel | I don't think he's the one I'm sad about. | Sadness | Sadness |
| **141** | **Phoebe** | **Yeah.** | **Sadness** | **Sadness** |

**Table 6.1:** Three conversations from dataset Friends, related to the word "Yeah" expressing three distinct emotions that are correctly classified by our Emotion Model.

## 6.2 Failure Cases

Table 6.2 illustrates some examples from the testing set of Friends and EmotionPush that are incorrectly classified by Hi-Transformer, Scene-1 from Friends, and Scene-2 from Emotion-Push.

1. **Scene1**: In Scene-1, our model makes wrong prediction for the $5^{th}$ utterance. Rachel is insisting on Ross to be friendly with his father even though Ross had some reservations about it. However, the text transcripts do not reveal very strong emotions compared to how the characters might act in the TV show. These kinds of scenes may be addressed by incorporating additional information like audio and video.

| Dialogue | Speaker | Utterance | Ground Truth | Prediction |
|---|---|---|---|---|
| **Scene1** | | | | |
| 13 | Rachel | Okay, look, Ross, I realise that my Father is difficult, but that's why you have got to be the bigger man here. | Neu | Neu |
| 13 | Ross | Look sweetie, I could be the bigger man, I could be the biggest man, I could be a big, huge, giant man, and it still wouldn't make any difference, except that I could pick your Father up and say 'Like me! | Non-Neu | - |
| 13 | Rachel | Okay, well can't you just try it one more time Ross? For me? For me? | Non-Neu | - |
| 13 | Ross | Rachel one brunch is not gonna solve anything. You gotta face it, okay we're never gonna get along. | Non-Neu | - |
| **13** | **Rachel** | **Okay, well you are just gonna have too, okay!** | **Anger** | **Neutral** |
| **Scene2** | | | | |
| 120 | Speaker1 | i dont think i'm cool i know i'm cool. | Neu | Neu |
| 120 | Speaker2 | so confident. so sexy. | Joy | Joy |
| 120 | Speaker1 | when we see each other, i will show u my confidence and sexiness? | Joy | Joy |
| 120 | Speaker2 | Are you sure you have that | Neu | Neu |
| **120** | **Speaker2** | **How are you going to show me?** | **Joy** | **Neu** |
| **120** | **Speaker1** | **I have many things. How I'm going to show them is a secret.** | **Joy** | **Neu** |
| **120** | **Speaker1** | **I don't want to ruin the experience.** | **Joy** | **Neu** |

**Table 6.2:** Examples from dataset Friends, illustrating the shortcomings of Hi-Transformer Emotion Model.

2. **Scene2**: In Scene-2, Speaker-1 and Speaker-2 are fooling around and Speaker-1 is boasting about his confidence. In this scene, utterances fifth, sixth an seventh utterances are misclassified. It can be noticed that the emotions indicated from the two utterances are very subtle even for humans. There is a high possibility that all three utterances may

have been labeled as 'Neutral' even by Human annotators. This example demonstrate that even after taking into the context of the conversation, the models' capability of understanding subtle emotions is still limited and more exploration is required.

# 7 Conclusion and Future Work

## 7.1 Conclusion

In the current work, we have reviewed and implemented HiTransformer, a transformer based context- and speaker-sensitive model for emotion recognition in conversations (ERC). We utilize BERT as the low-level transformer to generate individual utterance embeddings, and feed them into another high-level transformer to capture the global context of the conversation. To model the emotional interaction between speakers, we introduce speaker embedding into our model. The weighted loss avoids our model to only predict on majority class.

We evaluate our model on four benchmark datasets and demonstrate its effectiveness in comparison to the previous state-of-the-art models. Subsequently, we evaluated the model performance on the German translation of benchmarked datasets. Experimental results demonstrate that our model can effectively capture the context and speaker information in textual conversations. Furthermore, we experimented with several different strategies designed to understand the role of context in recognizing emotions. These experimental settings provide interesting insights about utterance-level dialogue understanding and help us in understanding the impact of position of an utterance, number of utterances in the dialogue, number of speakers in the dialogue, label and sentiment shift on the classifier performance.

## 7.2 Future work

In the current work, we have implemented HiTransformer, a transformer based model for emotion recognition in conversations(ERC). This work opens up several directions for future research on the task of multi-utterance emotion recognition in textual conversations. We are listing some of them here.

1. Our results show that embedding speaker information with the utterance context improves the model performance. It will be interesting to further develop speaker's personality embedding and investigate how this additional information can contribute towards further improvement in model performance.

2. ERC modelling approach in this work can be identified as offline as it is allowed to look at future utterances in a conversation. One interesting idea would be to further develop the modelling approach to support real-time dialogue systems, in which the model will have no information regarding the future context.

3. One of the major challenges in the emotion recognition task is finding a good labeled dataset. It will be interesting to explore semi-supervised learning methods to address the problem of data scarcity in this task.

4. Another interesting direction would be to collect specific types of labels to enrich the minor emotion categories, e.g., trying horror movies scripts to get more fear utterances and tragedies for sadness utterances.

# Acknowledgment

I would like to express my sincere gratitude to Prof. Dr. Martin Volk, and Dr. Annette Rios for their commitment to supervising this thesis.

# Appendices

# A Appendix

| From State / To State | Neutral | Sad | Angry | Joy | Non-Neu | Surprise | Fear | Disgust |
|---|---|---|---|---|---|---|---|---|
| Neutral | 0.54 | 0.03 | 0.03 | 0.10 | 0.16 | 0.11 | 0.02 | 0.02 |
| Sad | 0.35 | 0.21 | 0.04 | 0.07 | 0.17 | 0.12 | 0.01 | 0.02 |
| Anger | 0.30 | 0.03 | 0.28 | 0.04 | 0.22 | 0.09 | 0.02 | 0.03 |
| Joy | 0.36 | 0.02 | 0.02 | 0.32 | 0.14 | 0.10 | 0.01 | 0.02 |
| Non-Neutral | 0.39 | 0.04 | 0.06 | 0.09 | 0.28 | 0.11 | 0.02 | 0.02 |
| Surprise | 0.43 | 0.03 | 0.04 | 0.11 | 0.20 | 0.16 | 0.02 | 0.02 |
| Fear | 0.41 | 0.02 | 0.05 | 0.07 | 0.22 | 0.11 | 0.11 | 0.02 |
| Disgust | 0.34 | 0.03 | 0.08 | 0.07 | 0.21 | 0.12 | 0.10 | 0.13 |

**Table A.1:** Friends Dataset: Label Transition Probabilities Table

| From State / To State | Neutral | Sad | Angry | Joy | Non-Neu | Surprise | Fear | Disgust |
|---|---|---|---|---|---|---|---|---|
| Neutral | 0.74 | 0.02 | 0.01 | 0.11 | 0.08 | 0.04 | 0.00 | 0.01 |
| Sad | 0.49 | 0.21 | 0.01 | 0.10 | 0.14 | 0.04 | 0.00 | 0.01 |
| Anger | 0.46 | 0.06 | 0.16 | 0.11 | 0.16 | 0.02 | 0.00 | 0.01 |
| Joy | 0.51 | 0.03 | 0.01 | 0.33 | 0.09 | 0.03 | 0.00 | 0.01 |
| Non-Neutral | 0.52 | 0.07 | 0.02 | 0.14 | 0.19 | 0.04 | 0.01 | 0.01 |
| Surprise | 0.58 | 0.04 | 0.01 | 0.11 | 0.14 | 0.12 | 0.00 | 0.01 |
| Fear | 0.57 | 0.05 | 0.00 | 0.12 | 0.19 | 0.05 | 0.02 | 0.00 |
| Disgust | 0.49 | 0.04 | 0.01 | 0.09 | 0.19 | 0.08 | 0.00 | 0.09 |

**Table A.2:** EmotionPush Dataset: Label Transition Probabilities Table

| From State / To State | Neutral | Sad | Mad | Joyful | Peaceful | Powerful | Scared |
|---|---|---|---|---|---|---|---|
| Neutral | 0.45 | 0.06 | 0.09 | 0.19 | 0.06 | 0.05 | 0.11 |
| Sad | 0.29 | 0.18 | 0.08 | 0.13 | 0.11 | 0.8 | 0.13 |
| Mad | 0.25 | 0.05 | 0.23 | 0.16 | 0.09 | 0.08 | 0.15 |
| Joyful | 0.25 | 0.05 | 0.07 | 0.37 | 0.08 | 0.07 | 0.11 |
| Peaceful | 0.20 | 0.08 | 0.09 | 0.21 | 0.18 | 0.11 | 0.13 |
| Powerful | 0.18 | 0.06 | 0.10 | 0.20 | 0.13 | 0.19 | 0.13 |
| Scared | 0.24 | 0.06 | 0.13 | 0.16 | 0.10 | 0.11 | 0.20 |

**Table A.3:** EmoryNLP Dataset: Label Transition Probabilities Table

| From State / To State | Others | Sad | Angry | Happy |
|---|---|---|---|---|
| Others | 0.61 | 0.14 | 0.14 | 0.11 |
| Sad | 0.52 | 0.18 | 0.17 | 0.13 |
| Angry | 0.52 | 0.17 | 0.17 | 0.14 |
| Happy | 0.52 | 0.16 | 0.18 | 0.13 |

**Table A.4:** Semeval Dataset: Label Transition Probabilities Table

| **TrialId** | **Friends** | | | **EmotionPush** | | | **EmoryNLP** | | |
|---|---|---|---|---|---|---|---|---|---|
| | **F1**$_{macro}$ | **WA** | **UWA** | **F1**$_{macro}$ | **WA** | **UWA** | **F1**$_{macro}$ | **WA** | **UWA** |
| 1 | 69.12 | 83.23 | 65.12 | 68.32 | 88.42 | 61.82 | 25.54 | 39.83 | 27.91 |
| 2 | 67.54 | 81.94 | 64.09 | 65.39 | 87.16 | 65.58 | 27.14 | 41.58 | 28.27 |
| 3 | 68.65 | 82.58 | 64.92 | 66.31 | 87.81 | 64.34 | 26.61 | 40.19 | 29.69 |
| 4 | 68.99 | 82.99 | 65.07 | 65.94 | 87.01 | 62.25 | 25.59 | 39.96 | 27.63 |
| 5 | 66.32 | 81.15 | 63.54 | 66.16 | 86.44 | 63.32 | 27.84 | 40.38 | 29.98 |
| 6 | 68.08 | 82.31 | 63.93 | 67.12 | 87.31 | 64.47 | 26.68 | 40.39 | 29.49 |
| 7 | 67.52 | 82.01 | 63.61 | 66.11 | 86.98 | 61.19 | 25.91 | 38.72 | 26.53 |
| 8 | 67.89 | 82.24 | 63.64 | 66.64 | 87.01 | 63.41 | 27.15 | 39.14 | 29.22 |
| 9 | 68.73 | 83.01 | 65.04 | 67.39 | 87.12 | 64.19 | 27.28 | 40.76 | 28.72 |
| 10 | 68.89 | 83.11 | 64.93 | 67.57 | 87.33 | 64.14 | 26.37 | 39.07 | 28.88 |

**Table A.5:** Test Results on benchmarked datasets across 10 trials with speaker embedding support enabled and weighted balanced warming support disabled (WA and UWA represent weighted and unweighted accuracy respectively).

| TrialId | Friends | | | EmotionPush | | | EmoryNLP | | | Semeval | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | **F1**$_{macro}$ | **WA** | **UWA** | **F1**$_{macro}$ | **WA** | **UWA** | **F1**$_{macro}$ | **WA** | **UWA** | **F1**$_{micro}$ | **WA** | **UWA** |
| 1 | 66.41 | 82.19 | 64.44 | 63.32 | 88.67 | 57.74 | 26.46 | 40.33 | 27.98 | 72.34 | 83.52 | 82.29 |
| 2 | 66.92 | 83.24 | 64.84 | 64.03 | 89.03 | 59.33 | 26.09 | 40.19 | 27.58 | 69.37 | 79.83 | 79.04 |
| 3 | 65.65 | 81.18 | 62.92 | 63.55 | 87.13 | 58.11 | 26.96 | 41.68 | 27.02 | 73.54 | 83.81 | 82.76 |
| 4 | 65.99 | 81.99 | 63.07 | 62.99 | 85.17 | 61.05 | 26.59 | 40.72 | 27.88 | 71.28 | 81.36 | 80.87 |
| 5 | 67.22 | 83.15 | 64.94 | 63.56 | 87.39 | 60.66 | 25.48 | 38.29 | 25.28 | 72.67 | 82.93 | 82.04 |
| 6 | 66.08 | 82.31 | 64.13 | 61.84 | 85.01 | 62.13 | 26.39 | 40.29 | 25.34 | 71.17 | 81.61 | 80.29 |
| 7 | 67.01 | 82.94 | 64.61 | 63.72 | 87.86 | 58.49 | 26.43 | 40.44 | 27.33 | 73.18 | 83.74 | 82.47 |
| 8 | 67.91 | 83.24 | 64.97 | 62.44 | 86.14 | 59.44 | 25.22 | 38.34 | 24.98 | 71.29 | 81.48 | 80.88 |
| 9 | 68.19 | 83.41 | 65.39 | 61.99 | 85.38 | 60.12 | 25.11 | 38.12 | 24.76 | 70.61 | 81.24 | 80.57 |
| 10 | 66.89 | 82.01 | 63.22 | 62.59 | 86.14 | 59.94 | 26.77 | 40.80 | 28.10 | 69.97 | 81.11 | 80.25 |

**Table A.6:** Test Results on benchmarked datasets across 10 trials with speaker embedding and weighted balanced warming support disabled.

# Bibliography


Cecilia Ovesdotter Alm, Dan Roth, and Richard Sproat. Emotions from text: Machine learning for text-based emotion prediction. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 579–586, Vancouver, British Columbia, Canada, October 2005. Association for Computational Linguistics. URL https://aclanthology.org/H05-1073.

Christoph Alt, Marc Hübner, and Leonhard Hennig. Fine-tuning pre-trained transformer language models to distantly supervised relation extraction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1388–1398, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1134. URL https://aclanthology.org/P19-1134.

Alexandra Balahur, Jesús M. Hermida, and Andrés Montoyo. Detecting implicit expressions of sentiment in text based on commonsense knowledge. In *Proceedings of the 2nd Workshop on Computational Approaches to Subjectivity and Sentiment Analysis (WASSA 2.011)*, pages 53–60, Portland, Oregon, June 2011. Association for Computational Linguistics. URL https://aclanthology.org/W11-1707.

Yonatan Belinkov and James Glass. Analysis methods in neural language processing: A survey. *Transactions of the Association for Computational Linguistics*, 7:49–72, March 2019. doi: 10.1162/tacl_a_00254. URL https://aclanthology.org/Q19-1004.

Kaj Bostrom and Greg Durrett. Byte pair encoding is suboptimal for language model pretraining. In *Findings of the Association for Computational Linguistics: Empirical Methods in Natural Language Processing (EMNLP) 2020*, pages 4617–4624, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.findings-emnlp.414. URL https://aclanthology.org/2020.findings-emnlp.414.

Ankush Chatterjee, Kedhar Nath Narahari, Meghana Joshi, and Puneet Agrawal. SemEval-2019 task 3: EmoContext contextual emotion detection in text. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 39–48, Minneapolis, Minnesota, USA, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/S19-2005. URL https://aclanthology.org/S19-2005.

Junyoung Chung, Caglar Gulcehre, Kyunghyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. In *Advances in Neural Information Processing Systems (NeurIPS) 2014 Workshop on Deep Learning, December 2014*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://aclanthology.org/N19-1423.

Paul Ekman. Facial expression and emotion. *The American psychologist*, 48 4:384–92, 1993. doi: 10.1037//0003-066x.48.4.384.

Deepanway Ghosal, Navonil Majumder, Rada Mihalcea, and Soujanya Poria. Exploring the role of context in utterance-level emotion, act and intent classification in conversations: An empirical study. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1435–1449, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.findings-acl.124. URL https://aclanthology.org/2021.findings-acl.124.

Hatice Gunes, J. Vallverdú, and Maja Pantic. Automatic, dimensional and continuous emotion recognition. *International journal of synthetic emotions*, 1(1):68–99, January 2010. ISSN 1947-9093. doi: 10.4018/jse.2010101605. 10.4018/jse.2010101605.

Boris Hanin. Which neural net architectures give rise to exploding and vanishing gradients? In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper/2018/file/13f9896df61279c928f19721878fac41-Paper.pdf.

Devamanyu Hazarika, Soujanya Poria, Amir Zadeh, Erik Cambria, Louis-Philippe Morency, and Roger Zimmermann. Conversational memory network for emotion recognition in dyadic dialogue videos. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2122–2132, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1193. URL https://aclanthology.org/N18-1193.

Sepp Hochreiter and Jürgen Schmidhuber. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780, 11 1997. ISSN 0899-7667. doi: 10.1162/neco.1997.9.8.1735. URL https://doi.org/10.1162/neco.1997.9.8.1735.

Chao-Chun Hsu and Lun-Wei Ku. SocialNLP 2018 EmotionX challenge overview: Recognizing emotions in dialogues. In *Proceedings of the Sixth International Workshop on Natural Language Processing for Social Media*, pages 27–31, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-3505. URL https://aclanthology.org/W18-3505.

Yen-Hao Huang, Ssu-Rui Lee, Mau-Yun Ma, Yi-Hsin Chen, Ya-Wen Yu, and Yi-Shin Chen. Emotionx-idea: Emotion BERT - an affectional model for conversation. *Computing Re-*

*search Repository(CoRR)*, abs/1908.06264, 2019. URL http://arxiv.org/abs/1908.06264.

Wenxiang Jiao, Haiqin Yang, Irwin King, and Michael R. Lyu. HiGRU: Hierarchical gated recurrent units for utterance-level emotion recognition. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 397–406, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1037. URL https://aclanthology.org/N19-1037.

Sopan Khosla. EmotionX-AR: CNN-DCNN autoencoder based emotion classifier. In *Proceedings of the Sixth International Workshop on Natural Language Processing for Social Media*, pages 37–44, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-3507. URL https://aclanthology.org/W18-3507.

Agata Kołakowska, Agnieszka Landowska, Mariusz Szwoch, Wioleta Szwoch, and Michał Wróbel. *Modeling emotions for affect-aware applications*, pages 55–67. 01 2015. ISBN 978-83-64669-06-4.

Jingye Li, Donghong Ji, Fei Li, Meishan Zhang, and Yijiang Liu. HiTrans: A transformer-based context- and speaker-sensitive model for emotion detection in conversations. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 4190–4200, Barcelona, Spain (Online), December 2020a. International Committee on Computational Linguistics. doi: 10.18653/v1/2020.coling-main.370. URL https://aclanthology.org/2020.coling-main.370.

Qingbiao Li, Chunhua Wu, Zhe Wang, and Kangfeng Zheng. Hierarchical transformer network for utterance-level emotion recognition. *Applied Sciences*, 10(13), 2020b. ISSN 2076-3417. doi: 10.3390/app10134447. URL https://www.mdpi.com/2076-3417/10/13/4447.

Michael W. Morris and Dacher Keltner. How emotions work: The social functions of emotional expression in negotiations. *Research in Organizational Behavior*, 22:1–50, 2000. ISSN 0191-3085. doi: https://doi.org/10.1016/S0191-3085(00)22002-9. URL https://www.sciencedirect.com/science/article/pii/S0191308500220029.

Costanza Navarretta. Mirroring facial expressions and emotions in dyadic conversations. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 469–474, Portorož, Slovenia, May 2016. European Language Resources Association (ELRA). URL https://aclanthology.org/L16-1075.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237,

New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1202. URL https://aclanthology.org/N18-1202.

Soujanya Poria, Erik Cambria, and Alexander Gelbukh. Deep convolutional neural network textual features and multiple kernel learning for utterance-level multimodal sentiment analysis. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2539–2544, Lisbon, Portugal, September 2015. Association for Computational Linguistics. doi: 10.18653/v1/D15-1303. URL https://aclanthology.org/D15-1303.

Soujanya Poria, Navonil Majumder, Rada Mihalcea, and Eduard Hovy. Emotion recognition in conversation: Research challenges, datasets, and recent advances. *IEEE Access*, 7:100943–100953, 2019. doi: 10.1109/ACCESS.2019.2929050.

Sebastian Ruder, Matthew E. Peters, Swabha Swayamdipta, and Thomas Wolf. Transfer learning in natural language processing. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Tutorials*, pages 15–18, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-5004. URL https://aclanthology.org/N19-5004.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *ArXiv*, abs/1910.01108, 2019.

Carlo Strapparava and Alessandro Valitutti. WordNet affect: an affective extension of WordNet. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04)*, Lisbon, Portugal, May 2004. European Language Resources Association (ELRA). URL http://www.lrec-conf.org/proceedings/lrec2004/pdf/369.pdf.

Gongbo Tang, Mathias Müller, Annette Rios, and Rico Sennrich. Why self-attention? a targeted evaluation of neural machine translation architectures. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4263–4272, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1458. URL https://aclanthology.org/D18-1458.

Ashish Vaswani, Noam Shazeer, Nikki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, and Łukasz Kaiser. Attention is all you need. In *Proceedings of the Neural Information Processing Systems, Long Beach, CA, USA*, page 6000–6010, 2017.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP) Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium, November 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-5446. URL https://aclanthology.org/W18-5446.

Wenxiang Jiao, Michael R. Lyu, and Irwin King. Pt-code: Pre-trained context-dependent encoder for utterance-level emotion recognition. *Computing Research Repository (CoRR)*, 2019.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online, October 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-demos.6. URL https://aclanthology.org/2020.emnlp-demos.6.

Chung-Hsien Wu, Ze-Jing Chuang, and Yu-Chung Lin. Emotion recognition from text using semantic labels and separable mixture models. *ACM Transactions on Asian Language Information Processing*, 5:165–183, 2006.

Sayyed Zahiri and Jinho D. Choi. Emotion Detection on TV Show Transcripts with Sequence-based Convolutional Neural Networks. In *Proceedings of the Association for the Advancement of Artificial Intelligence (AAAI) Workshop on Affective Content Analysis*, AFFCON'18, pages 44–51, New Orleans, LA, August 2018. URL https://sites.google.com/view/affcon18.