CLASSIFICATION IN TEXTUAL CONVERSATIONS: A STUDY OF EMOTION PREDICTION AND DERAILMENT FORECASTING

ENAS ALTARAWNEH

A DISSERTATION SUBMITTED TO THE FACULTY OF GRADUATE STUDIES
IN PARTIAL FULFILMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

GRADUATE PROGRAM IN ELECTRICAL ENGINEERING
AND COMPUTER SCIENCE
YORK UNIVERSITY
TORONTO, ONTARIO

FEBRUARY 25,2025

Abstract

Emotion is fundamental to human communication, shaping not just the content but the very essence of our interactions with others. In the realm of Natural Language Processing (NLP), particularly for applications that bridge human-machine communication such as health-care, education, and social networks, understanding and emulating emotional nuances becomes paramount. While it may be straightforward for humans to perceive and reason about the feelings of others in conversations, it is a challenge for machines, mainly due to context. Conversation models in the literature that incorporate context vary in the type of contextual information they incorporate (e.g., temporal structure, speaker identification, commonsense knowledge). However, studies to date have not explicitly quantified the impact of the type(s) of information incorporated within the critical conversation classification tasks of *future emotion prediction* and *emotional derailment forecasting*, nor the structure of the model architectures and encoding structures used for these tasks. These issues are addressed in this work.

This thesis approaches this problem by developing AI models that can capture different design choices for these tasks. Critically, the models developed here are designed to capture three properties inherently connected to the emotional predictive problem in dialogues; *sequence modeling*, *self-dependency modeling*, and *recency*. These modeling dimensions are

then incorporated into one of two deep neural network architectures, a *sequence model* and a *graph convolutional network model*. The former is designed to capture the sequence of utterances in a dialogue, while the latter captures the sequence of utterances and the formation of multi-party dialogues. Through an empirical evaluation of these model architectures, data type and data encoding choices, this work demonstrates (i) the importance of the self-dependency and recency model dimensions for the prediction tasks, (ii) the effectiveness of graph neural models in improving the predictions obtained by sequence-only models, (iii) the impact of fusing multi-source information of each utterance into utterance capsules, specifically emotion labels and common sense knowledge and, (iv) that using a transformer-based forecaster for the conversation predictive task also improves performance. Optimal design choices within these structures provides near best in class performance for next emotion prediction in conversations and best in class performance for conversation derailment prediction. This thesis also shows that simple fine-tuning of large language models is not an effective classification method for these tasks.

Evaluations are performed using standard conversational datasets and current state of the art network models. Results from this work will help inform future dataset structure and the development of advanced sentiment analysis systems.

Acknowledgements

I had a dream, a dream of wisdom shared. I first saw it in my father, a professor and inspiring mentor, whose steadfast belief in perseverance became the rhythm of my own journey, teaching me that dreams do come true. My mother, with her boundless love and remarkable ability to bring light into darkness, showed me that even in the toughest moments, there is always a way to create something extraordinary.

To my supervisor, Michael, my compass throughout this journey. His guidance illuminated my path and provided unwavering support through both triumphs and challenges. He made me feel understood, capable, and empowered to achieve more than I imagined. To Manos, whose inspiration, wisdom, and optimism taught me to embrace my PhD journey, and whose fearless encouragement motivated me to push beyond my limits. To Ameeta, for your warm guidance, friendship, and the invaluable conversations we shared.

To my husband and children, you are my foundation, my strength, my heart. My husband, who always believed in me, whispering, "You can do it, and I'm here for you," with patience, care, and constant support. To my children, whose laughter brightened the darkest moments and whose love gave me the strength to keep moving forward. This journey was mine, but I never walked it alone.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Emotion is intrinsic to humans and consequently emotion understanding is a key part of human-like artificial intelligence (AI). Unlike simple query-response interactions, in many modern systems interactions between a human and a machine take the form of a conversation, in which information and emotional state is communicated over multiple interactions between the two parties. Humans are exceptionally gifted at recognizing the emotional state presented by individuals with whom they are communicating, and adapting their conversational interactions in response to this. Enabling the HAL robot in the movie 2001 [73] to be able to say "Look Dave, I can see you're really upset about this. I honestly think you ought to sit down calmly, take a stress pill, and think things over." requires enabling a robot with the ability to infer sentiment from an agent's actions. Here we look at this problem in the context of textual-based conversations, such as those encountered in online social media platforms. Inferring the *emotional trajectory* in *multi-turn conversations* is becoming increasingly important as a research frontier in natural language processing (NLP) as it can be used to enhance the naturalness of the interaction [147, 58, 148, 30, 12, 11, 4]. It has potential application in health-care systems (as a tool for psychological analysis), education (understanding student frustration) and more. Additionally, understanding emotions in conversations is also extremely

important for generating emotion-aware dialogues and to inform models of pre-emptive toxicity detection that require an understanding of the speaker's emotions. Understating emotions in conversations enables online conversation moderation in publicly available conversational platforms such as Facebook, Youtube, Reddit, Twitter, and others, to maintain the integrity of these and other platforms.

Emotion AI or Sentiment Analysis (also known as opinion mining) is one of many areas of computational studies that deals with opinion oriented natural language processing [49]. Such opinion-oriented studies include among others, genre distinctions, emotion and mood recognition, ranking, relevance computations, perspectives in text, text source identification and opinion oriented summarization [96]. Traditionally, sentiment analysis dealt with opinion polarity, i.e., whether someone has a positive, neutral, or negative opinion towards something. This might explain why sentiment analysis and opinion mining are often used as synonyms. Some think it is more accurate to view sentiments as emotionally loaded opinions [87]. Sentiment analysis is widely applied to materials such as reviews and survey responses, online and social media, and healthcare materials for applications that range from recommender systems to clinical medicine [49, 74].

Before diving to deeply into the technology of emotion understanding we begin by defining emotion. Emotion can be defined (see [32, 67]) as an individual's mental state associated with thoughts, feelings and behaviour. The action and mood variation of individuals has been of great importance to many philosophers leading to theories that sought to explain them. For example, In Stoic theories, normal emotions (like delight and fear) are described as irrational impulses and 'good emotions' (like joy and caution) are experienced by those that are wise [8]. Aristotle believed that emotions were an essential component of virtue [37]. In the Aristotelian view, all emotions (called passions) correspond to appetites or capacities. Cicero organized emotions into four categories; fear, pain, lust and pleasure [28]. European history and philosophy are not the only source of such models. For example, in the early 11th century,

Ibn Sina, an Arabic philosopher, theorized about the influence of emotions on health and behaviors, suggesting the need to manage emotions [48]. Early Chinese philosophy, especially as presented in the Liji (Book of Rites) [117], an ancient text outlining rituals, customs, and moral principles foundational to Confucian thought, classifies emotions such as joy, anger, and love as inherent feelings crucial to human experience. It underscores the significance of attaining equilibrium and harmony in emotional expression, connecting emotions to moral behavior and social cohesion, while also acknowledging that *qing*, a unique Chinese concept of affectivity encompassing essence, nature, reality, disposition, desire, and feelings, includes a wider array of experiences beyond simple psychological states [117].

The evolutionary theory of emotion was initiated in the late 19th century by Charles Darwin [97]. He hypothesized that emotions evolved through natural selection and, hence, have cross-culturally universal counterparts. Modern models of emotions can be traced to Ekman. In the early 2000's Ekman proposed six basic emotions: fear, anger, joy, sadness, disgust, and surprise [67]. More recent cross-cultural studies led by Daniel Cordaro and Dacher Keltner, both former students of Ekman, extended Ekman's list of universal emotions [32]. In addition to the original six emotions defined in [67], these studies provided evidence for amusement, awe, contentment, desire, embarrassment, pain, relief, and sympathy in both facial and vocal expressions. They also found evidence for boredom, confusion, interest, pride, and shame facial expressions, as well as contempt, relief, and triumph vocal expressions. Plutchik [99] categorized emotion into eight primary types, visualized by the wheel of emotions shown in Figure 1.1. Jaak Panksepp carved out seven biologically inherited primary affective systems called seeking (expectancy), fear (anxiety), rage (anger), lust (sexual excitement), care (nurturance), panic/grief (sadness), and play (social joy). He proposed what is known as the "core-SELF" to be generating these affects [71].

Ekman argued that there exists a correlation between emotion and facial expression [67]. Seeing the evolution and sophistication of emotions by scientists and philosophers informs

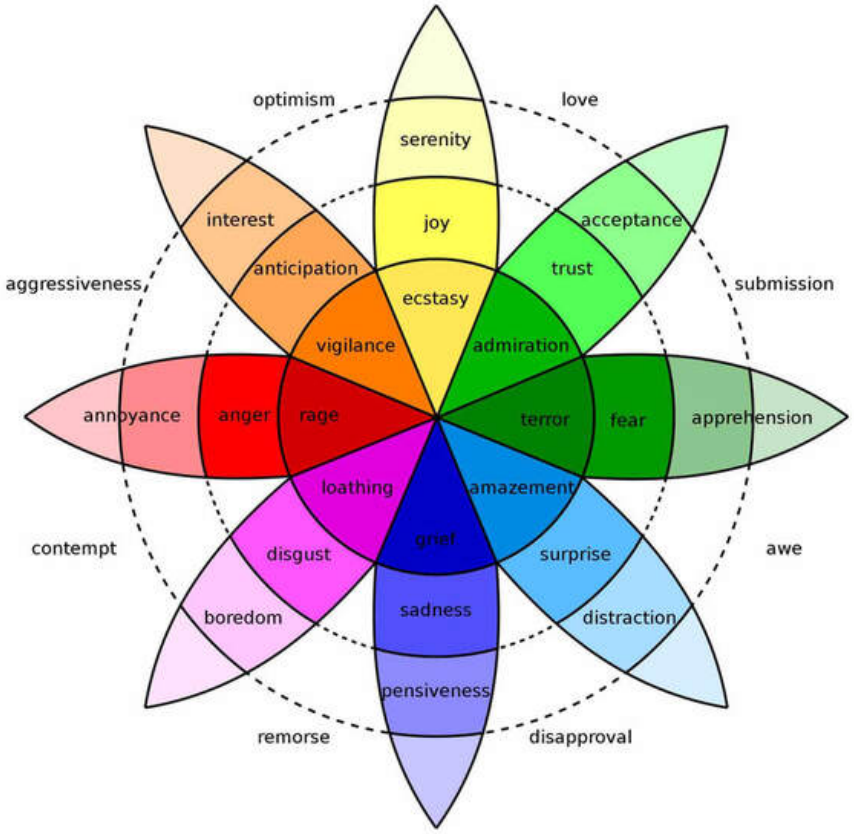


Figure 1.1: Dr. Robert Plutchik's Wheel of Emotions categorizing emotion into eight primary types. Redrawn from [99].

our understanding that the emotion of an utterance is difficult to estimate and detect, and that feeling and expressing that feeling relates to factors other than the specific utterance presented in isolation.

To differentiate between emotion and sentiment, one can look at sentiment as distinctly social. Gordon identifies sentiment as a socially constructed combination of autonomic responses, expressive behaviors, and shared meanings usually organized around another person [120]. Sentiments are the expression of emotions tied to a social object, whereas emotions are primarily confined to the psychological dimensions. For example, love is a

sentiment that is characterized by autonomic symptoms such as the flow of adrenaline and increased heart rate and may be expressed in gazing and smiling at the other. Through socialization we learn the vocabulary to name our internal sensations given objects or events that we encounter, and we learn how to express them accordingly [122]. Therefore sentiment is a mental attitude; a thought that has been influenced by emotions which in turn are complex psychological states. Emotions are very raw and natural. Sentiments are highly organized.

For machine-based systems, and for the purpose of this work, it is perhaps easiest to take a very operational view of the difference between emotion and sentiment. Sentiment being the appearance of the emotional state. And in a gross generalization, assume that appearance is a true and complete reflection of internal state.

While it is straightforward for humans to perceive and reason about the feelings of others in conversations, it is a challenge for machines, mainly due to context. Traditionally, studies on emotion analysis for machines focused on recognizing the emotion of an individual utterance isolated from other utterances in the same conversation (see [123] for example). More recent studies (e.g., [86, 55, 72, 42, 125]) have studied emotion analysis of utterances within the context of a conversation, where conversation modules accumulate and incorporate information from the conversation history to assist in the task of emotion classification. When a given utterance or turn is taken "out of context" so to speak, then the context is lost. But when the turn is considered within the context in which it occurred it becomes necessary to identify explicitly what context is being considered. Conversation models in the literature range in the type of information they are able to incorporate and include temporal [86, 55, 72], speaker [42, 125, 145, 85, 81], and commonsense knowledge [121, 113, 41]. However, studies to date have not explicitly explored the impact of the types of information incorporated. For example, What is the relative value of including speaker identification in sentiment analysis? Understanding the impact of different contexts allows for better decisions on the features used

as input, the structure of conversations, and the architecture of the machine learning models used. One of the key contributions of this thesis is the incorporation of emotion-related insights into the edge formation of the graph within the conversation model. This work studies the behavior of emotions in sequence models and graph models, extracting features that relate to both the user of interest and other users to understand the influence of emotional shifts. By analyzing the predictive model's behavior with respect to emotion, the model adapts its graph structure by introducing self-user edges to incorporate self-dependency and applying a sparsification technique to connect utterances to more recent edges. This allows the model to better capture emotional transitions and predict shifts in a more contextually aware manner.

Persistence and *contagion* are two basic characteristics of emotion in conversations [102, 77]. A speaker may maintain their existing emotional state or be affected by others as the conversation proceeds. *Persistence* refers to the tendency of the speaker to maintain a consistent emotional state from one utterance or turn to another. *Contagion* refers to how the public and communicated perception by others of the emotional intent of an utterance or textual turn in a conversation or online dialogue influences a speaker's emotions. The public up/down vote on utterances in online conversations is an example of such information. A conversation model that incorporates such information when modeling the relationship and emotional dynamics between speakers in a conversation would be able to account for these characteristics and improve on the models' ability to infer emotions. Many current studies on conversation emotion predication and derailment forecasting (e.g., [145, 85, 81]) also neglect other important features in the conversation such as *recency* (encoding the temporal localization of emotions) and *self-dependency* (encoding the relationship between same speaker data) that could improve predication performance. Other than *recency* and *self-dependency*, studies have shown that conversation classification architectures that incorporate design decisions to track emotion shifts which directly relate to the emotion propagation

characteristics *persistence* and *contagion* results in improved performance [77, 135]. The novality in this thesis is the through the study of the predictive model behaviour on the emotion, the model incorporates this infomation in a way that provides the best predictive benifits.

Compared with humans, a computer's reasoning ability is still very limited, in part due to the lack of commonsense knowledge. While humans rely on a vast store of intuitive knowledge, shaped by experience, culture, and social interactions, computers struggle to replicate this natural understanding. Commonsense knowledge, which includes basic facts about the world, human behavior, and everyday reasoning, is essential for enhancing a machine's ability to interpret and respond in more nuanced ways. There exist many commonsense knowledge bases (e.g. [41, 113, 121]) that can potentially be used to help with the conversation classification task. These databases contain relational common sense knowledge related to text that can help converational models make more informed decisions. Conversation classification models often encounter difficulties in complex context propagation, emotion shift detection, and differentiating between related emotion classes. In addition to the previously mentioned direct modeling of *recency*, *self dependency* and *emotion shifts*, learning distinct commonsense representations for a conversation can addresses these challenges [41, 113, 121].

Interactive applications can benefit from a range of different conversation emotional classification tasks, including: emotion recognition, future emotion prediction, and forecasting emotional derailment. *Emotion recognition* estimates the emotional intent of utterances within a complete conversation, *future emotion prediction* aims to predict the emotional intent of the next future utterance, and *forecasting emotional derailment* involves predicting whether and when a conversation emotional trajectory will go off-course and derail due to hateful or offensive remarks. These inter-related topics are addressed in this work and are summarized in Table 1.1. This work explores the use of similar architectural designs across these different yet related conversational problems, offering new insights into how these models can be

Abbreviation	Problem	This study
nEMC	Next Emotion in Conversation - predict the next emotion in a conversation given emotion and other information from earlier turns in the conversation	Yes
FConvD	Forecasting Conversational Derailment - given a conversation that has not derailed earlier in the conversation, predict if it will derail at the next turn	Yes
ERC	Emotion Recognition in Conversations - estimate the emotion label of all utterances in a conversation given an entire conversation context	No

Table 1.1: List of problems that relate to modeling conversation architecture. This work addresses the nEMC and FConvD problems. ERC is a well studied problem.

adapted and optimized for diverse tasks within the conversation classification domain. This approach also opens the door for multi-task research, enabling the development of models that can be applied to a broader range of similar conversational challenges moving forward.

When it comes to supervised machine learning models, a range of common AI architectures have been used in research studies to address these emotional classification tasks. Since these tasks are essentially emotional conversation classification tasks, a model that can better represent the conversation and incorporate more information related to the relationships between the utterances and speakers will likely perform better for all of these tasks. In the literature, conversations for the emotion recognition and prediction classification tasks are typically structured as sequences (e.g., [86, 52, 63, 56, 132, 80]), graphs (e.g., [42, 125, 145, 85, 81, 60, 119, 118]) or as knowledge-graphs (e.g., [121, 106, 17, 113, 146, 41, 149]). A brief review of these architectures is provided in Chapter 2.

The existing literature has not explicitly shown the benefits of conversation structures coupled with different machine learning model architectures for emotion classification. For example, one could extend on GCN (Graph Convolotional Networks) architectures and/or

commonsense knowledge graph models by modeling emotion recency and user dependency in the architecture. Graph models can be used to represent complex dialogue dynamics. And so on. One novelty of the graph model used in this work lies in its adaptation of the graph structure by introducing self-user edges to capture self-dependency, employing a sparsification technique to link utterances to more recent edges, and incorporating user-to-user relational edges to capture complex user-to-user dynamics and emotion propagation. However, there exist cases where conversations are too short to form effective graph models, and the complexity of a graph model may be unnecessary and the conversation may be better structured using a sequence model. Furthermore a more complex architecture may hallucinate some incorrect solution given the lack of data to train or to draw conclusions from. Studying the relationship between conversation length, structure and classification can provide insights to creating an adaptable emotional machine learning model, that adapts to features as presented to it. This thesis also explores the impact of conversation structure with conversations of varying length on classification.

Models with extensible architectures that allow for multi-modal or multi-data designs result in enhanced performance for down stream classification problems [52, 51, 64]. By incorporating various types of data into a model with a similar architecture, one can discern the individual and collective impact of these data inputs. This thesis introduces a generalizable forecasting model, The Forecasting Graph Convlotional Neural network (FGCN) that is layered with diverse data types, allowing for a comprehensive analysis of how each data type affects the model's predictions. By providing a range of different information capsules into the classification model, this thesis investigates the influence of each data type both individually and in combination. The findings presented later in this work underscore the importance of data variation in enhancing model performance, with emotional and common sense knowledge data emerging as significant contributors. This suggests that models equipped with the capability to process such data can more effectively capture the nuances of complex

characteristics, such as the dynamics of multi-party dialogue, context propagation, and mood shifts in conversations. The novel extensible nature of the FGCN model's architecture is advantageous, as it facilitates the inclusion of impactful data types, thereby enriching the model's understanding and interpretation of intricate conversational elements. This approach not only improves performance but also provides a framework for future research to further refine and expand upon the model's capabilities, potentially leading to more nuanced and human-like interactions in conversational artificial intelligence systems.

The problems of emotion prediction and emotional derailment forecasting involve predicting the future trajectory of a conversation. Large language models (LLM) such and Bert [35], GPT3 [36], GPT4 [94], Llama [128], Claude [21] and Gemini [126], in addition to being classification models, are generative models that can be used to generate next most likely utterances given a conversation. Standard practice in the literature assumes the text of the upcoming utterance in a conversation is omitted. That is, prediction takes place without foreknowledge of the next utterance. One approach that has not yet been fully investigated in the literature is to generate a prediction of this utterance using a LLM then to use this additional information to inform these tasks. This thesis explores using LLM's generated utterance information for emotion prediction tasks, and explores the use of curriculum based learning within the classification model. This work shows that including an estimated last utterance in the FGCN model's data input seems to have an interesting but negative effect on performance: variants that do not include a LLM-generated last utterance outperforms those that do. This suggests that the generation capabilities of LLMs, while powerful, introduce noise or irrelevant information that detracts from the model's forecasting accuracy. LLMs are indeed sensitive to prompt engineering; their generalized nature means that they require careful crafting of prompts to produce the desired output. The straightforward prompt engineering employed in this study may not have leveraged the full potential of LLMs. Sophisticated prompts that incorporate more contextual instructions

or utilize techniques like Retrieval-Augmented Generation (RAG) could potentially yield better results.

This thesis also explores the use of a variation of text embeddings in the forcaesting model. It appears that smaller language models (SLMs) like GPT and BERT, when used to create embeddings for populating the nodes in a GNN, show superior performance over larger language models (LLMs). This could be due to the more focused training and data representation capabilities of SLMs, which may align better with the specific features of derailment forecasting tasks. Fine-tuning these models before embedding extraction further enhances their performance, likely because it allows the model to adjust more closely to the nuances of the task-specific data.

The main contributions of this thesis are:

- A novel conversation forecasting/prediction classification model is developed based on a graph convolutional neural network that captures emotion shifts, dialogue user dynamics, and public perception of conversation utterances enriched with commonsense knowledge and LLM generated data. Preliminary work in this regard is published in [5, 7, 22].

- This forecasting model is used in a novel analysis of conversation derailment and next emotion predication in dyadic and multi speaker conversations. Preliminary work in this regard is published in [5, 7].

- Various types of data are incorporated into the same forecasting model, through its novel extensible architecture, to study the individual and collective impact of these data inputs. Preliminary work in this regard is published in [5, 7, 6].

- Curriculum-based learning is used to study the impact of different training paradigms on the classification tasks. Preliminary work in this regard is published in [5].

- An extensive empirical evaluation of the structurally informed models on benchmark datasets is performed.

The rest of this document is organized as follows: Chapter 2 provides a review of conversation models in the literature. Chapter 3 describes the conversion forecasting models used in this work. Chapter 4 describes the problem of predicting emotions in conversations using the conversation forecasting model described in this work. Chapter 5 describes the problem of conversion derailment forecasting using the forecasting model described in this work. Chapter 6 provides a summary of this work and suggests directions for future research. Finally, Appendix A lists and describes the available datasets for the task of emotional classification in conversations.

Chapter 2

Literature Review

This thesis deals with modeling sentiment state in a conversation. This computational model includes mechanisms that model *recency* in sentiment state throughout a conversation. It also includes mechanisms to represent *persistence* and *contagion* which are two basic characteristics of emotion in conversations [102, 77]. Representing such information when modeling a conversation enhances the model's ability to infer the emotions presented by the individuals involved. This thesis also deals with more complex conversational characteristics such as context propagation, emotional shifts and emotional derailment. Conversational emotional derailment is heavily impacted by the underlying commonsense knowledge shared between speakers. Studies on conversational emotional classification [41, 149] have shown that incorporating common sense knowledge can capture such conversational characteristics and enables the use of this information in emotion classification.

A key issue in modeling conversations is the development of an appropriate computational model. This chapter reviews relevant AI models, with a particular focus on graph-based representations and concludes with a brief introduction to generative AI models and the learning approaches that are relevant to this work. This chapter also reviews the structure of a conversation and the various tasks that might be brought to bear on turns in a given

conversation, including sentiment prediction and conversational derailment. This chapter also provides a formal definition of these problems. The chapter concludes with a summary of open problems and a description of the problems that this work seeks to address. It begins with a formal definition of a conversation and the prediction tasks of interest.

2.1 What is a conversation

Imagine a group of identifiable individuals engaged in a conversation. For simplicity, suppose that only one individual speaks at a time and that we concatenate the uninterrupted utterances of a given individual into a single group. For simplicity we will refer to this group of utterances as a *turn*, as this work deals with textual conversations. This results in a simple model of a conversation which consists of a sequence of turns, in which a given turn is associated with an identifiable individual. If we have n turns in a conversation, we can identify the user u_i of turn i, and the text t_i associated with that turn. Here we assume that the context is textual but this representation can be easily generalized to other forms of communication.

The problems addressed in this work involve estimating or reasoning about the emotional content e_i associated with the conversation at turn i. We assume that there exists some discrete set of pre-defined emotional labels. L, such as those proposed by [103], and $e_i \in L$. We recognize two different versions of the problem of estimating e_i, a discrete version in which the goal is to identify the single emotional label e_i associated with turn i, and a continuous version in which we seek a probability distribution over the possible labels. That is some distribution $p(e_i)$ such that $\forall^i p(e_i) \geq 0$, and $\sum p(e_i) = 1$.

To assist in understanding this representation, consider a conversation between three individuals (A: Monica, B: Ross, C: Phoebe) engaged in a conversation of six turns extracted from the MELD emotion conversation dataset [103], as shown in Figure 2.1. The dataset uses emotional labels L={Anger, Disgust, Sadness, Joy, Neutral, Surprise, Fear} and assigns a

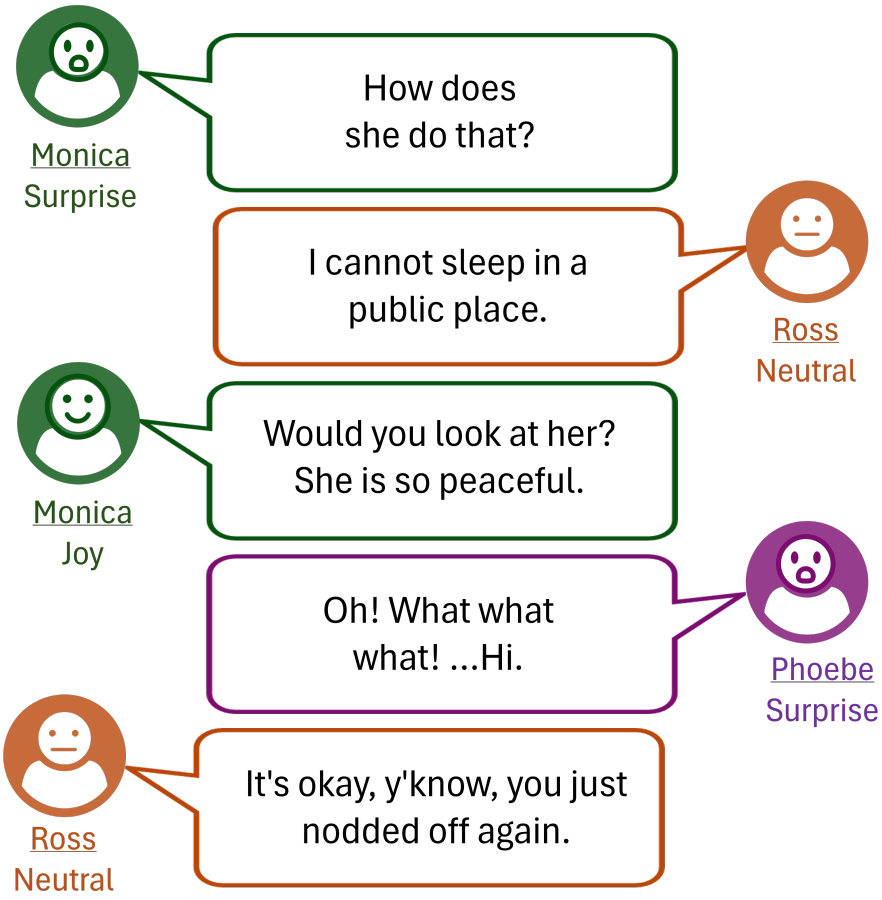


Figure 2.1: A conversation between three individuals (A: Monica, B: Ross, C: Phoebe) engaged in a conversation of six turns from the MELD emotion conversation dataset [103]. The dataset uses emotional labels L = {Anger, Disgust, Sadness, Joy, Neutral, Surprise, Fear}.

single emotion label for each utterance in the conversation. Figure 2.1 shows the conversation and associated hand-assigned emotional labels from the dataset. Given this representation, we now move on to formalizing the problems of interest in this work; emotion recognition, next emotion prediction, and conversation derailment forecasting.

2.2 Emotional conversation classification tasks

This section reviews three related emotional classification tasks related to conversations; emotion recognition, next emotion prediction and forecasting emotional derailment.

2.2.1 Emotion recognition in conversations

Emotion recognition is typically formulated as a structured classification problem in which sentences or utterances are classified in isolation, and emotion classification is performed on properties of the turn or utterance in isolation. This labelling problem may use the choice of words used (e.g., [3, 69, 138]), or sentence structure (e.g., [69, 138]), or for audio-based utterances the tone of the voice used (e.g., [83, 88]).

Emotion recognition in conversations (ERC) is the problem of estimating the emotions of turns within the context of a conversation. For text-based conversations this context typically includes the transcript of the conversation along with the speaker information associated with each constituent turn. ERC aims to identify the emotion label of each turn or a distribution of certainty over a set of predefined emotion labels over the entire conversation. Table 2.1 illustrates a sample conversation between two people from the conversational dataset DAILYDIALOG [79], where each utterance is labeled by the underlying emotion from the set of emotions in DAILYDIALOG.

Definition 2.2.1 (Emotion recognition in conversations (ERC))**.**
Let $\mathcal{C} = ((t_1, e_1, u_1), ..., (t_n, e_n, u_n))$ be a sequence denoting a conversation of n turns, where $t_i = (w_{i,1}, ..., w_{i,m})$ is the sequence of $k = 1, ..., m$ words $w_{i,k}$ at turn i and used by speaker or user u_i with emotion label $e_i \in \mathcal{E}$ where $\mathcal{E}$ is a finite set of emotion categories. Note that each turn has an independent m. The discrete version of the task is to estimate the emotion label $\hat{e_i}$ of each turn t_i in the conversation $\mathcal{C}$. The probabilistic version involves estimating the probability distribution $\hat{p}(e_i)$ associated with each turn i.

Turn	User	Text	Emotion
1	$\mathcal{A}_1$	*"Is everything alright?"*	**neutral**
2	$\mathcal{A}_2$	*"No, the steak is not very fresh."*	**anger**
3	$\mathcal{A}_1$	*"Oh! Sorry to hear that."*	**sadness**

Table 2.1: A sample conversation taken from the DAILYDIALOG dataset [79] showing text utterances and their emotion labels. The emotion recognition task in conversation is to estimate the emotion of the text of each utterance, here e_1=neutral, e_2=anger, e_3=sadness

Emotion recognition in conversations (ERC) has been widely studied due to its potential for use in applications. An interesting application of ERC involves analyzing conversations that take place on social media for pre-emptive toxicity detection [141, 18, 13] or the problem of detecting early indicators of antisocial discourse, which has become a pertinent research topic. ERC can also aid in analyzing conversations in real time, which can be instrumental in legal trials, interviews, providing e-health services and more. ERC ideally requires context modeling of the individual turns.

Recent work in ERC generally relies on deep learning methods to capture contextual characteristics, which can be divided into sequence-based (e.g., [86, 55, 72]) and graph-based methods (e.g., [42]). Another research direction involves improving the performance of existing models by incorporating external knowledge (e.g., [78]), which are classified here as knowledge-based methods. Compared to graph- and knowldge-based works on ERC, vanilla emotion recognition approaches, which use Recurrent Neural Networks (RNN), often fail to work well on ERC datasets in part because they ignore conversation-specific factors such as the presence of contextual cues, temporality in speakers' turns, and speaker-specific information.

Turn	User	Text	Emotion
$n-3$	$\mathcal{A}_1$	*"Is everything alright?"*	neutral
$n-2$	$\mathcal{A}_2$	*"No, the steak is not very fresh."*	anger
$n-1$	$\mathcal{A}_1$	*"Oh! Sorry to hear that."*	sadness
n	$\mathcal{A}_2$		???

Table 2.2: A sample conversation from DAILYDIALOG showing text associated with a given turn and their emotion labels. Given $n-1$ turns, the next emotion prediction task in conversations is to predict the emotion at turn n (anger, in this case) for a given user without knowledge of t_n

2.2.2 Next emotion prediction in conversations

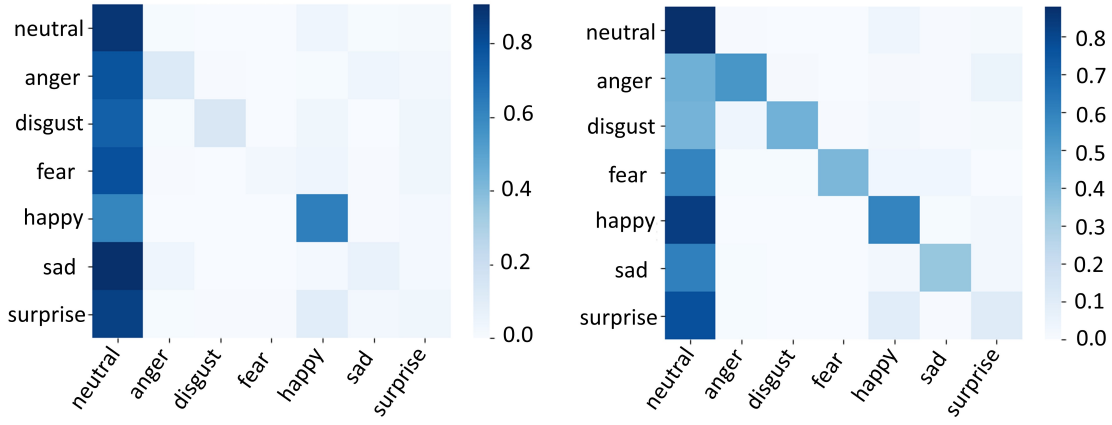
Next emotion prediction aims to predict how the content of other parties in a conversation affect the emotion of users in an ongoing dialogue. This task demonstrates a conversational model's ability to predict the emotion trajectory of a conversation before the actual (future) turn's text becomes available. This task explores the emotional influence from the users' perspective. In this task, it is critical to integrate the turn information of the users and model the emotional interaction between the users. Given a sequence of turn text in a turn-taking conversation, the next emotion prediction problem is to predict the likelihood of certain emotion(s) being expressed by a user when they next generate text. An example of this task in a conversation coming from a two-speaker conversation dataset, DAILYDIALOG, is presented in Table 2.2.

Definition 2.2.2 (Next Emotion prediction in conversation (nEMC))**.**
Let $\mathcal{C} = ((t_1, e_1, u_1), ..., (t_n, e_n, u_n))$ be a sequence denoting a conversation of n turns, where $t_i = (w_{i,1}, ..., w_{i,m})$ represents the sequence of $k = 1, ..., m$ words $w_{i,k}$ at turn i and used by user u_i. Note that each turn has an independent m. In the discrete version of this problem we have emotion labels $e_i \in \mathcal{E}$ where $\mathcal{E}$ is a finite set of emotion categories. Given some

conversation $\mathcal{C}$ consisting of a specific sequence of labeled turns, up to and including turn $n \geq 1$, the goal is to predict the emotion $\hat{e}_{n+1}$ at the *next* turn for a given user without access to the actual content text or emotion label of $n+1$. In the continuous version of the problem with the probability distributions $\forall^i \leq n\ p(e_i)$, for each turn and the goal is to estimate the probability distribution $\hat{p}(e_{n+1})$.

Figure 2.2 shows the conditional probability of transition of emotions from the DAILYDIALOG dataset. In Figure 2.2a, we observe that for the majority of the cases, there is a high probability of transitioning to a *neutral* state when the two utterances pertain to *two* different users, except in the case of *happy* which is more likely to be expressed by the other party. Figure 2.2b, on the other hand, suggests that emotion consistency is typically maintained when the two utterances pertain to the *same* user (i.e., there is a high level of self-dependency). Interestingly, we note that *surprise* is likely to transition to *happy* on the next turn. These observations motivates further exploration of the emotion data and using feature selection to extract relevant parts of the emotion history. While current models are designed to extract this type of relevant information (see [42], for example), in existing approaches, the data is not processed or filtered to enhance the results by avoiding noise, or excluding older or non-influencing utterances. In addition, many architecture designs do not take other data insights into consideration (e.g., speaker information and the emotion of past utterances).

Predicting the next or future emotion in a conversation has not been studied extensively. Rashkin et al. [107] and Gaonkar et al. [40] introduced an emotion inference task to predict how events influence emotions in the characters of a story. Although this task is more closely associated with multi-party multi-turn conversational datasets, they studied the emotion inference task using a news and stories corpus. Hasegawa et al. [50] predicted and elicited the addressee's emotion in online dialogue, where they used two turns as context and used a



(a) $p(e_{i+1}|e_i)$ (b) $p(e_{i+1}|e_{i-1})$

Figure 2.2: Transition matrix based on our work in [7] showing the transition probabilities (a) of one emotion at turn i (vertical) to another at turn $i+1$ (horizontal). (b) of one emotion at turn $i-1$ (vertical) to another at turn $i+1$ (horizontal). For this dyadic conversation e_{i+1} and e_{i-1} come from the same user; e_i pertains to another user.

non-neural-network method. Predicting emotions has also been presented as an auxiliary task in a number of emotion response generation studies (e.g., [145, 85, 81]). These studies do not explicitly show the accuracy of the predictions. Other models such as knowledge graphs and curriculum based learning have not yet been investigated for this task.

There is a wide range of applications for which such a forecasting model can be useful, including forecasting the emotional trajectory of a conversation between a machine and human agent or pre-emptively detecting hate speech in social forums [54, 33, 89, 109, 100, 2]. A machine's ability to infer emotional consequences in conversations is critical to achieve conversational intelligence and is important for various real-life tasks, including dialogue planning, dialogue generation and interpretability, empathetic machines, and customer service, among others. For example, emotion inference is a key component for human-like chatbots. One natural application of a predictive model is in the area of empathetic response generation

where existing strategies either mimic the previous emotion, require pre-determined emotion signals [147, 58, 148, 30], or jointly model emotion prediction and response generation [84, 9, 10, 27, 47]. Although these studies show the possibility of generating a response capable of conveying an emotion, the approach is limited in that the emotion of the response is determined manually by the user.

2.2.3 Emotional derailment in conversations

Emotional derailment in conversations refers to the process by which a conversation or discussion is redirected away from its original topic or purpose, typically as a result of inappropriate or off-topic comments or actions by one or more participants. In online conversations, derailment can be exacerbated by the lack of nonverbal cues and the anonymity that can be provided by the internet causing escalation with anger and frustration. Table 2.3 shows an example conversation, coming from the popular Conversation Gone Awry (CGA) benchmark dataset [141]. One can observe that there is offensive language used by one of the participants and this is associated with conversational derailment. The severity of the verbal attack may indicate a prior history between the two participants in previous conversations. Conversation derailment can lead to confusion, anger, frustration, and a lack of productivity or progress in the conversation.

Definition 2.2.3 (forecasting conversation derailment (fConvD))**.** Let $\mathcal{C} = ((t_1, v_1, u_1), ..., (t_n, v_n, u_n))$ be a sequence denoting a conversation of n turns, the last turn (i.e., the n^{th} turn) is the potential site of derailment where $l \in \{civil, derailed\}$ denotes the label of this turn. For the i^{th} turn, $t_i = (w_{i,1}, ..., w_{i,m})$ represents the sequence of words w presented in turn i, u_i denotes the i^{th} user, and v_i denotes an optional score, (e.g., number of votes, up-vote/down-vote, associated with this turn in an online forum). Each turn has a separate m. A positive value of v_i denotes a positive public perspective of the turn and a

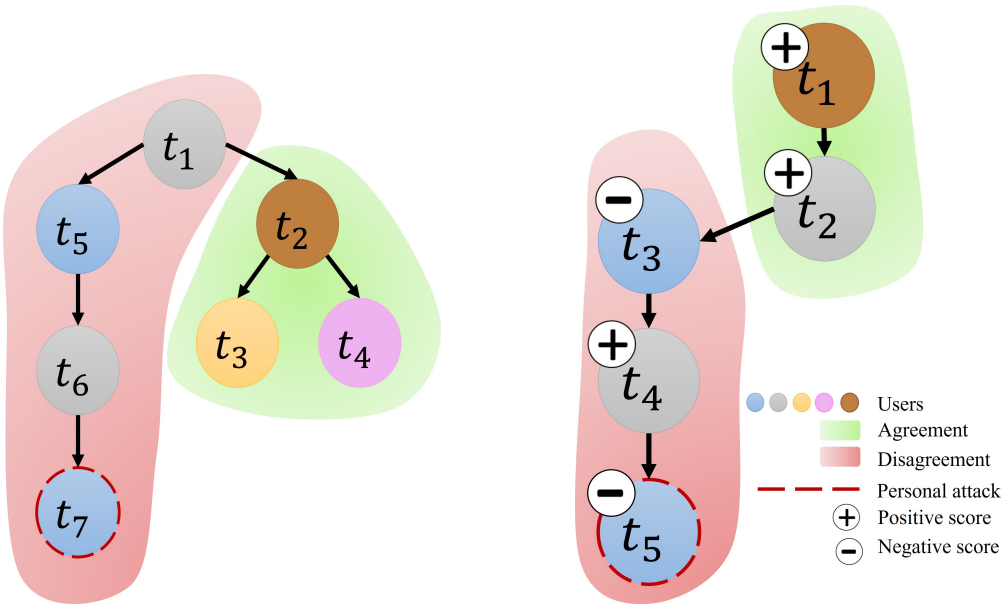
Utterance	User	Text	Label
$n-3$	$\mathcal{A}_1$	*Proper use of an editor's history includes fixing errors or violations of Wikipedia policy or correcting related problems on multiple articles.*	
$n-2$	$\mathcal{A}_2$	*All you are doing is following me round the biography articles it's very clear that you just go to my contributions list and look to see what biography articles I've worked on, then you go and look to see if you can find something wrong with them.*	
$n-1$	$\mathcal{A}_1$	*So, what is wrong with fixing things? At the top of my talk page, it says to keep it on your watchlist.*	
n	$\mathcal{A}_2$		**?**

Table 2.3: A sample conversation coming from the Conversation Gone Awry (CGA) dataset showing a sequence of text utterances that end with verbal abuse. Given the conversation context up to and including $N-1$ utterances which has not yet derailed, the task is to predict whether utterance N will be a respectful or offensive statement leading to derailment.

negative value denotes negative public perspective. The goal is to forecast the derailment label l_i of the n^{th} utterance given a civil conversation $\mathcal{C}$ from turn 1 up to and including turn $n-1$.

This work assumes that the conversation did not derail in turns $1 \ldots n-1$ but may derail at turn n. In practice, a forecast could be made at each turn in the conversation assuming derailment has not yet occurred. In addition to forecasting a possible derailment, for the set of conversations that do derail and *their derailment is positively detected*, the task can be further refined to forecast the number of utterances into the future at which derailment will occur. That is to predict $\min_k l_k == \textit{derailed}$. In the continuous version of the problem if $(1-p(l_i))$ is the probability of non-derailment of turn i then the probability of non-derailment up to and including turn n is $(1-p(l_1)) \ldots (1-p(l_n))$ and if this ever becomes less than some threshold then one could predict that the conversation will have derailed by turn n.

Being able to predict conversation derailment early has multifold benefits: (i) it is *more*



(a) Conversation C_{CGA} (b) Conversation C_{CMV}

Figure 2.3: Conversation examples drawn from two benchmark datasets, CGA and CMV. The turns are encoded with the letter t and a number describing the turns' sequential occurrence in time, produced by users encoded with different colors. The structure of the conversation is based on a parent-child relationship, where, for example, the child text of turn t_2 is a direct reaction to parent turn text t_1. Participants are in either agreement (encoded within a green region) or disagreement (encoded within a red region). The last utterance is the site of derailment encoded in a red dashed line. Both parts of the figure show information neglected in previous work on derailment forecasting; (a) illustrates graph dynamics in a conversation, and (b) illustrates public perception through votes in a conversation encoded with positive and negative icons.

timely, as it allows for proactive moderation of conversations (before they cause any harm) due to early warning, (ii) it is *more scalable*, as it allows for automatically monitoring large active online communities, a task that is otherwise time-consuming, and (iii) it is *more cost-effective*, as it reduces and possibly eliminates the need for human moderators.

Early efforts towards automatic moderation in a conversation (e.g., [133, 100]) focused on detecting inappropriate comments once they have occurred. But the utility of such an approach is limited as participants have already been exposed to abusive behavior and any potential harm has already been caused. The current state-of-the-art approach to predicting conversation derailment relies on sequence models that treat dialogues as text streams [24, 68]. However, this approach has limitations, as it ignores the semantics and dynamites of a multi-party dialogue involving individuals with their own intrinsic characteristics and history of interactions with other individuals.

Conversations Gone Awry (CGA) [141] and Reddit ChangeMyView (CMV) [23] are two popular benchmark datasets employed for this problem. These datasesets are described in detail in Section 5.3 and Appendix A. Both datasets contain multi-party conversations, including the text and the user ID of each utterance, along with a label annotation on whether the conversation will or will not derail at the last turn. CMV also includes a public vote on each of the utterances that provides the public perception of people listening in on the conversation towards that utterance. These are the number of up votes (community support for the text at that turn) or down votes (community disagreement with the text at that turn). The number of up votes minus the number of down votes is recorded. Figure 2.3a shows an example multi-party conversation from CGA that is not sequential in nature. Graph models are more suitable to represent such dialogue dynamics. Figure 2.3b shows an example conversation from CMV that shows the voting scores on each turn in the conversation that could be related to derailment.

2.3 Modeling a conversation

2.3.1 Sequential conversation modeling

Perhaps the simplest approach to modeling a conversation context is to model the conversation as a sequence of turns. One can use Recurrent Neural Networks (RNN) (e.g., [86, 52]) or transformers (e.g., [35, 132, 68]) among other sequence models (e.g., [63, 51, 24]) to represent a conversation in this manner. Extensive research has been performed leveraging such neural network architectures for emotion recognition in conversational data of different modalities such as text, speech, and vision [86, 55, 72, 51, 52]. Individuals evoke different responses in others as a function of their tendency to express particular emotions. ICON [52] is a multimodal emotion detection framework that extracts multimodal features from conversational videos and hierarchically models the self- and inter-speaker emotional influences into global memories. Such memories generate contextual summaries which aid in predicting the emotional orientation of turn-videos. CMN [51] uses a multimodal approach comprising audio, visual and textual features with gated recurrent units to model past utterances of each speaker into memories. These memories are then merged using attention-based hops to capture inter-speaker dependencies. [86] proposed a method for emotion detection in conversations in which the individual person state and global context state is tracked throughout the conversation. The model, named DialogueRNN, is based on a recurrent neural network that uses two Gated Recurrent Units (GRU's) which takes into account that emotion occurrence in an conversation depends on three factors: the speaker, the preceding context in the conversation and the preceding emotion behind the conversation. Figure 2.4 illustrates the abstract structure of DialogueRNN and demonstrates the sequential nature of the architecture.

HiGRU [63] address three challenges in utterance-level emotion recognition in dialog systems: (1) the same word can deliver different emotions in different contexts; (2) some

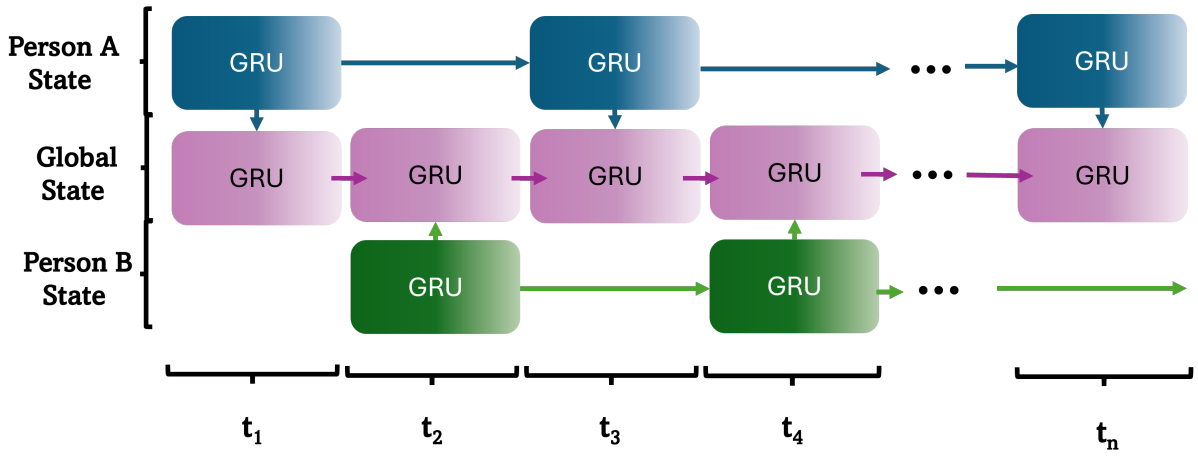


Figure 2.4: The sequential structure of a conversation in DialougeRNN [86]. The dyadic model assumes two speakers and maintains a state for each speaker and a global conversation state at each point of time in the sequential timeline. The network consists of a collection of gaited recurrent units (GRU) connected in a linear fashion.

emotions are rarely seen in general dialogues; and (3) long-range contextual information is hard to capture effectively. This hierarchical Gated Recurrent Unit (HiGRU) framework addresses these issues with a lower-level GRU to model the word-level inputs and an upper-level GRU to capture the contexts of utterance-level embeddings. This framework has two variants, Hi-GRU with individual features fusion (HiGRU-f) and HiGRU with self-attention and features fusion (HiGRU-sf), so that the word/utterance-level individual inputs and the long-range contextual information can be sufficiently utilized.

DialogueCRN [56] was introduced to understand the conversational context from a cognitive perspective. Inspired by the Cognitive Theory of Emotion [115], it uses multi-turn reasoning modules to extract and integrate emotional clues. The reasoning module iteratively performs a retrieval process and a conscious reasoning process, which is based on a model of human cognitive thinking. CESTa [132] uses sequence tagging where a Conditional Random Field (CRF) layer is leveraged to learn the emotional consistency in the conversation. It

employs LSTM-based encoders that capture self and inter-speaker dependency of interlocutors to generate contextualized utterance representations which are fed into the CRF layer. In order to capture long-range global context, CESTa uses a multi-layer Transformer encoder to enhance the LSTM-based encoder. EmoCaps [80] extracts multi-modal emotion vectors from different modalities and fuses them with a sentence vector to create an emotion capsule. Emocaps uses these emotion vectors to obtain the emotion classification results from a context analysis model.

The CRAFT models [24] were among the first models to go beyond classification of hate speech to addressing the problem of forecasting emotional conversation derailment. The CRAFT models integrate two components: (a) a generative dialog model that learns to represent conversational dynamics in an unsupervised fashion, and (b) a supervised component that fine-tunes this representation to forecast future events. As a proof of concept, a mixed methods approach combining surveys with randomized controlled experiments investigated how conversation forecasting using the CRAFT model can help users [24] and moderators [116] proactively assess and deescalate tension in online discussions. Extending the hierarchical recurrent neural network architecture with three task-specific loss functions as proposed in [62] was shown to improve the CRAFT models. After pretrained transformer language encoders such as BERT [35] proved to be successful at various NLP tasks, [68] explored how BERT can be used for forecasting derailment. Similarly, [34] evaluated feature-based and neural models to predict whether disagreements in Wikipedia Talk page conversations would be escalated to mediation by a moderator.

2.3.2 Graph conversation modeling

Graph convolutional neural networks have also been used for conversation classification. This approach improves the context understanding demonstrated in sequence models by taking

into account inter-speaker dependency and self-dependency (the emotional influence a person has on themself during conversation). In one popular application, emotion estimation, the graph model is used to account for speaker-related information [42, 125, 145, 85, 81].

DialogueGCN [42] is a graph neural network-based approach for emotion recognition in conversations. This model improves context understanding by taking into account inter-speaker dependency and self-dependency between different individuals involved in an conversation. The directed graph for this model is constructed such that each statement in the conversation is represented by a vertex, while edges are modeled on the basis of the context and edge weights are computed using a similarity based module.

DialogInfer [77] is an speaker-aware framework; it models the speakers's emotional state. By leveraging information on contagion and persistence of emotions in conversations, it tracks if the speaker maintains the current emotional state or if the speaker's emotional state is affected by others as the conversation proceeds. A sequence-based and a graph-based model are used to accumulate short-term and long-term conversation history information respectively. An addressee-aware module is designed for both models to learn whether the addressee keeps her/his own historical emotional state or is affected by others. [60] provides relational position encoding to a relational graph attention networks (RGAT) with sequential information reflecting the relational graph structure. The RGAT model can capture both the speaker dependency and the sequential information. DaG [119] encodes the utterances as a directed acyclic graph to model the intrinsic structure within a conversation. DaG combines the strengths of conventional graph-based neural models and recurrence-based neural models, DAG-ERC provides a more intuitive way to model the information flow between long-distance conversation background and nearby context. Conversational hierarchical structures [118] are not conducive to the application of pre-trained language models such as XLNet. DialogXL was proposed as an all-in-one XLNet model with enhanced memory to represent longer historical contexts and dialog-aware self-attention to deal with the multi-party structure in

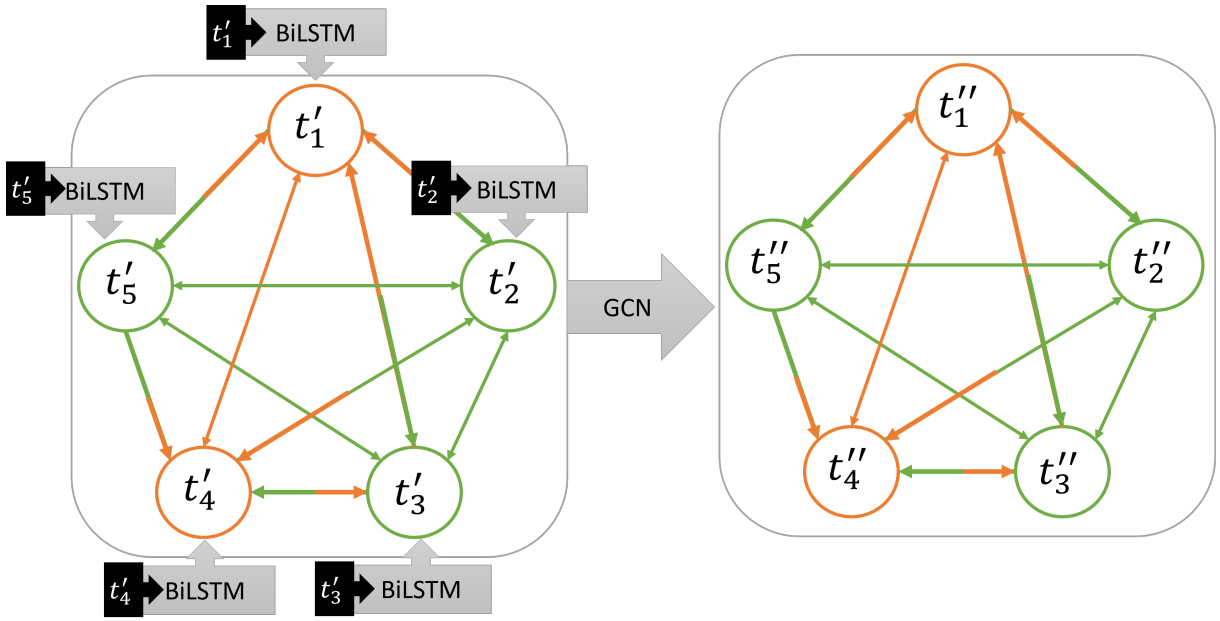


Figure 2.5: The graph structure of a conversation in DialougeGCN. The relational graph model on the left populates the nodes in the graph with the sequential encoding of the text from the turns in a conversation. Each node represents the text in a given turn with BilSTM sequential encoding. The edges between the nodes encode user information. User nodes are represented with different colors (orange and green) representing the two users. The edges encode the users relationship. For example, a green and orange edge represents that these two turn's text were generated by two different users and weights the possibility of "other user" influence on the text. On the other hand, a full orange or green edge represents a "same user" relationship between the two turn's text and weights the possible *persistence* of user state. The model uses graph neural networks to transform the features and encode the dynamics between users for classification as shown on the right side of the figure.

conversations. DialogXL modifies the recurrence mechanism of XLNet from segment-level to utterance-level in order to better model the conversational data.

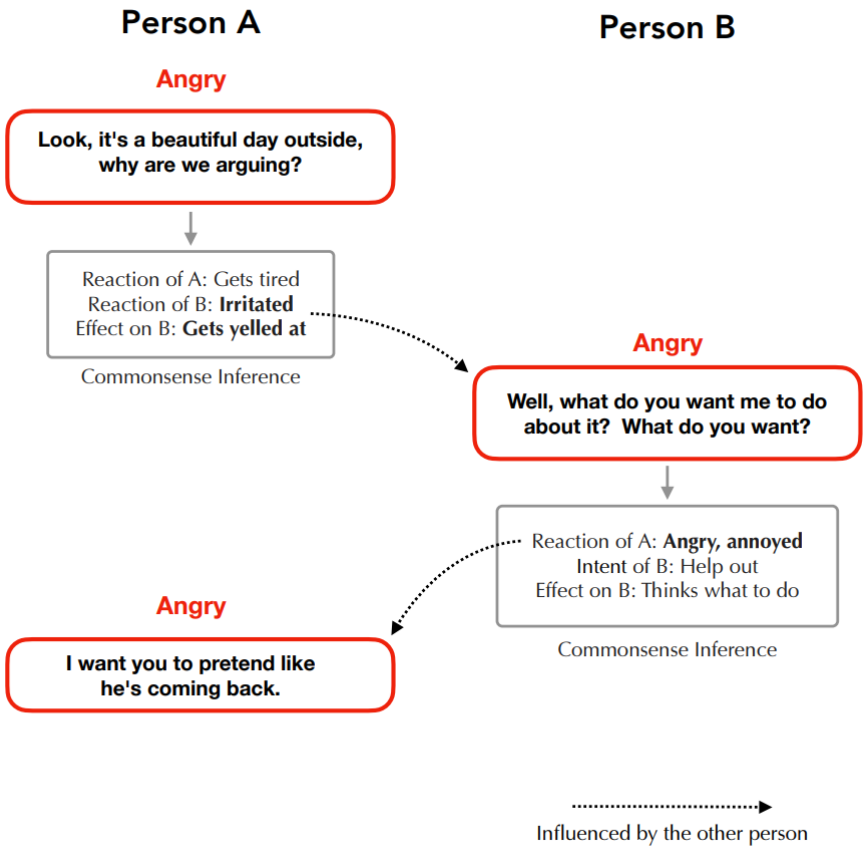


Figure 2.6: Commonsense knowledge can lead to explainable dialoge understanding. It can help models understand, reason, and explain events and situations. In this particular example, commonsense inference is applied to a sequence of turn's text in a two party conversation. Person A's first text indicates that he/she is tired of arguing with person B. The tone of the text also implies that person B is being yelled at by person A, which invokes a reaction of irritation in person B. Person B then asks what he/she can do to help and says this while being angry. This again makes person A annoyed and influences him/her to respond with anger. This kind of inferred commonsense knowledge about the reaction, effect, and intent of the speaker and the listener helps in predicting the emotional dynamics of the participants. Reprinted from [41].

2.3.3 Knowledge graphs and conversation modeling

Compared with humans, a computer's reasoning ability is still very limited, in part due to the lack of commonsense knowledge. There are many commonsense knowledge bases that can potentially be used to help with the emotion inference task.

ConceptNet [121] was the first ERC model to integrate common-sense knowledge. ConceptNet is a semantic network that contains concept-level relational commonsense knowledge. SenticNet [36] is a sentiment lexicon that is widely used for sentiment analysis tasks. Event2Mind [106] and ATOMIC [113] focus on inferential knowledge organized as typed if-then relations with variables. COMmonsEnse Transformers (COMET) [17] is a generative model trained on ConceptNet and ATOMIC, which is able to generate novel, rich, and diverse knowledge that is not explicitly represented in the original knowledge bases. KET [146] is a knowledge-enriched transformer that interprets contextual utterances using hierarchical self-attention and dynamically leverages external commonsense knowledge using a context-aware effective graph attention mechanism.

S-BERT (Sentence-BERT) [108] serves as an effective method for extracting common sense knowledge for a given utterance by generating semantically meaningful sentence embeddings and computing their relationships relative to external knowledge sources. It first converts both the input text, such as a conversational utterance, and statements from a common sense knowledge base into high-dimensional vector representations using a transformer-based model. By computing the cosine similarity between these embeddings, S-BERT determines how closely the input text aligns with known facts or logical inferences. When integrated into a retrieval-based system, precomputed embeddings of common sense knowledge (e.g., ConceptNet or ATOMIC) allow for efficient matching, enabling the system to retrieve the most relevant knowledge statements based on similarity scores. Additionally, fine-tuning S-BERT on datasets that explicitly link conversational data to common sense knowledge

improves its ability to recognize nuanced semantic relationships, making it particularly useful for tasks such as conversational emotion prediction, derailment forecasting, and response generation by grounding predictions in real-world knowledge.

The integration of commonsense knowledge into dialogue systems is a significant advancement in the field. It enables a more nuanced understanding of conversations by considering the underlying implications and emotional context of exchanges between individuals (see Figure 2.6). By analyzing the sequence of utterances, the system can infer the emotional state and intentions of the speakers, leading to more accurate predictions of the conversation's direction. This approach not only enhances the interpretability of dialogue systems but also contributes to the development of more empathetic and responsive AIs. As AI-based systems continue to evolve, incorporating such human-like reasoning capabilities will be crucial in creating systems that can interact with users in more natural and meaningful way. For example, COSMIC [41], shown in Figure 2.7 incorporates different elements of commonsense information such as mental state, events, and causal relations, extracted using COMET, and builds upon these elements to learn interactions between interlocutors participating in a conversation. By learning distinct commonsense representations, COSMIC adopts a network structure very close to DialogRNN and adds external commonsense knowledge from ATOMIC [113] to improve its performance. TODKAT [149] is a Topic-Driven Knowledge-Aware Transformer that utilizes a topic-augmented language model (LM) with an additional layer specialized for topic detection. The topic-augmented LM is then combined with commonsense statements derived from a knowledge base based on the dialogue contextual information. The transformer-based encoder-decoder architecture fuses the topical and commonsense information in ATOMIC.

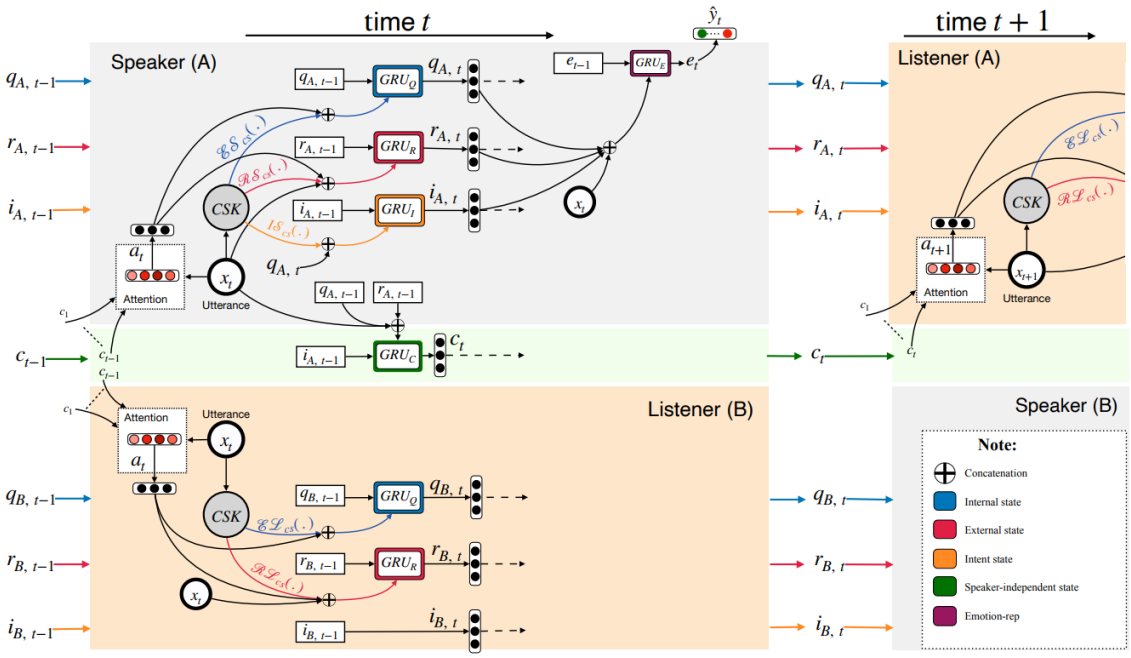


Figure 2.7: The COSMIC framework. In the framework CSK represents COMET, the commonsense knowledge generation that feeds into a number of maintained states. The nature of the overall model structure is sequential and utilizes COMET for utterance feature extraction from its knowledge graphs. Reprinted from [41].

2.4 Learning sentiment trajectories in conversations

Curriculum learning [15] is a training strategy which imitates the meaningful learning order found in human curricula. The core idea of curriculum learning is to train the machine learning model with easier data subsets at first, and then to gradually increase the difficulty level of data until the entire training dataset is learned. Recent studies (e.g., [130]) have demonstrated that providing training examples in a meaningful order rather than presenting them in random order can boost model performance.

A curriculum learning strategy has demonstrated its power in improving the overall performance of various models in a wide range of scenarios [130]. In particular, curriculum

learning has shown promise in NLP tasks [146, 82, 124]. Starting from [15] a variety of curriculum learning approaches has been studied in the field of NLP. For example, curriculum learning has been used for tasks such as neural machine translation [140, 82], relation extraction [59] and natural answer generation [82].

In the context of this thesis, curriculum learning is used to improve traditional ERC methods. [135] introduced a hybrid curriculum learning framework for conversation emotion classification. Due to the hierarchical structure of the ERC datasets, the curricula was constructed from two granularities: (1) conversation-level curriculum (CC); and (2) utterance-level curriculum (UC). In CC, the author constructed a difficulty measurer based on "emotion shift" frequency within a conversation. Conversations are scheduled in an "easy to hard" schema according to the difficulty score returned by the difficulty measurer.

Given this approach the question of how to measure the difficulty of conversations and utterances arises. Previous studies (e.g.[86, 119]) have reported that many ERC methods suffer from two issues: the "emotion shift" problem, methods cannot efficiently handle scenarios in which emotions of two consecutive utterances are different [43]; and the "confusing label" problem. The "confusing label" problem occurs when a model fails to distinguish between similar emotions. Previous methods [42, 119] usually fail to distinguish between similar emotions very well. This is due in part to the subtle semantic difference between certain emotion labels such as happy and exciting.

2.5 Large language models

Large language models (LLMs) have revolutionized the field of natural language processing (NLP). Their ability to understand and generate human-like text has opened up new possibilities in AI, from automating customer service to creating more interactive and responsive virtual assistants. The advancements in these models' capabilities, such as in-context learning

and few-shot learning, have significantly reduced the amount of data needed to train effective AI systems, making them more efficient and accessible for a wider range of applications.

The evolution of language models represents a significant leap in the field of natural language processing. The transition from small transformer language models (SLMs) (e.g., GPT [104], BERT [35], RoBERTa [150]), to large language models (LLMs) (e.g., ChatGPT [93], GPT4 [94], Claude2 [21], LLaMA [128]), marks a paradigm shift in our approach to understanding and generating human language. LLMs, with their vast number of parameters, are not just larger versions of their predecessors; they exhibit emergent abilities that were not apparent in smaller models. This includes a remarkable capacity for generalization, allowing them to perform a wide array of tasks without task-specific training. Moreover, their deep learning architectures enable them to handle complex tasks that were previously out of reach, making them invaluable tools for a variety of applications in technology and communication.

The landscape of large language models is diverse, reflecting a broad spectrum of applications and capabilities. General LLMs, such as GPT-4, Claude, ChatGPT, and LLaMA, are engineered to handle a variety of natural language processing (NLP) tasks without being tailored to any specific one. This versatility makes them invaluable tools for a range of applications, from conversational AI to text summarization. However, the trade-off is that they may not reach the peak performance of specialized models in certain tasks. On the other hand, specialized LLMs, which are fine-tuned for specific tasks via task-specific architectures and knowledge, are optimized for particular functions, which can lead to more efficient processing and potentially superior results in those areas. 'Phoenix' by Wang et al. [26] is an example of specialized LLMs providing a more targeted solution that can cater to the nuanced demands of language processing across different languages and contexts.

LLMs have transformative potential, but their limitations are an active area of research and development. The closed-source nature of many LLMs can limit collaboration and innovation within the community, potentially slowing progress. Additionally, while LLMs offer broad

domain knowledge, they may lack the depth required for specialized fields, necessitating further refinement and training on niche datasets to enhance their expertise and utility in such areas.

In the context of this thesis, large general purpose LLMs lack distinct focus on the domain task of emotion understanding [143]. As a result, using LLMs for emotion recognition can lead to suboptimal and inadequate precision. [142] showed LLMs' unsatisfactory performance in many emotion recognition tasks without fine-tunning on emotional knowledge. [75] presented a retreival-based framework to improve the adaptability of LLMs for emotion recognition. The potential of LLMs in understanding emotional communication needs to be further explored. One recent study [131], introduced DialogueLLM, an emotion-tailored LLMs that can better understand human-human conversation and takes a further step towards emotion intelligence. DialogueLLM, is an emotion and context knowledge enhanced language model, which is specifically designed for ERC based on the open-source base models, namely LLaMA. DialogueLLM collects diverse instruction conversational data based on emotional knowledge from five open-source benchmarking datasets (i.e., MELD [103], IEMOCAP [19], EmoryNLP [53]).

2.6 Response generation

In the digital age, Information Retrieval (IR) systems have become indispensable tools for sifting through the immense volumes of data available online. They enable users to find relevant information efficiently, whether through search engines like Google [44], Bing [16], and Baidu [14], or via interactive platforms such as ChatGPT [93] and Bing Chat [29]. Moreover, recommendation engines like those used by Amazon and YouTube further tailor the user experience by suggesting content aligned with individual preferences. Collectively, these IR technologies are not just facilitators of convenience; they are vital conduits for the

global exchange of knowledge and ideas.

The evolution of Information Retrieval (IR) systems has been marked by a significant shift from traditional sparse retrieval methods to advanced dense retrieval techniques. Sparse retrieval, relying on word-level matching, has been the backbone of IR for decades, with methods like Boolean Retrieval [112], BM25 [110] and SPLADE [38] providing efficient and robust performance. These methods excel in connecting a specific vocabulary to relevant documents, thus ensuring high retrieval efficiency. However, the advent of deep learning has introduced dense retrieval methods such as DPR [65] and ANCE [134], which leverage the BERT model's bidirectional encoding representations to grasp deeper semantic meanings within documents [35]. This has led to a notable improvement in retrieval precision. Despite these advancements, dense retrieval methods are not without their limitations. Dense retrieval methods depend on extensive document indices and lack the capability for end-to-end optimization. Furthermore, the ultimate goal of IR systems is to deliver precise and reliable answers to users, who still need to sift through document ranking lists to find the information they seek. This highlights an ongoing challenge in the field: to develop IR systems that not only improve accuracy but also streamline the user experience by reducing the time and effort required to obtain relevant information.

The transformative impact of LLMs on the field of artificial intelligence is undeniable. With roots in Transformer-based architectures, models like T5 [105], BART [76], and GPT [94] have redefined the boundaries of text generation, showcasing remarkable proficiency in producing coherent and contextually relevant content. This proficiency stems from extensive pre-training on diverse corpora, coupled with sophisticated techniques such as Reinforcement Learning from Human Feedback (RLHF) [66], enabling these models to excel in natural language tasks. The evolution of LLMs has been particularly influential in the realm of IR, birthing the concept of Generative Information Retrieval (GenIR). This new paradigm leverages the generative capabilities of LLMs to fulfill IR objectives, marking a significant departure from

traditional retrieval methods.

In this thesis, the capabilities of LLMs are harnessed to predict subsequent utterances, which can be instrumental in estimating the future emotional state of a speaker or the likelihood of a conversation going off track. The utilization of machine learning models to augment or fill in missing data is a well-established practice in data preprocessing. This approach not only enhances the quality of the data but also improves the efficiency of the data analysis process, leading to more accurate predictions and insights.

2.7 Summary

The task of emotion recognition has been researched extensively but often in the context of utterances in isolation. Emotion recognition in conversations generalizes the task to include the context of the conversation, and this provides the for enhanced performance as a result. A related but more challenging task, but desirable in many applications, is predicting the emotion trajectory of a conversation before the actual (future) utterances become available to the model.

There is a wide range of applications for which an emotional forecasting model can be useful, including forecasting the emotional trajectory of a conversation between a machine and a human agent or pre-emptively detecting hate speech in social forums. Psychological studies show that humans create mental models of emotion transitions and can predict others' emotions up to two transitions into the future with an above-chance accuracy [127]. The feasibility of computational models for predicting emotion trajectories in conversations have been barely explored. One natural application of such a predictive model is in the area of empathetic response generation where existing strategies either mimic previous emotion, require pre-determined emotion signals or jointly model emotion prediction and response generation. Although these studies show the possibility of generating a response capable

of conveying an emotion, the approach is limited in that the emotion of the response is typically determined manually by the user or through training examples provided by the developer. Another interesting application lies in the area of pre-emptive toxicity detection or the problem of detecting early indicators of antisocial discourse, which has become a pertinent research topic. [114] studied the relationship between structure and toxicity in conversations on Twitter at the individual, dyad, and group level, and found that social relationships among users influence their behaviors. [111] also studied the differences between toxic and non-toxic conversations on Twitter, highlighting important differences between user engagement and toxicity. While these recent works stress the importance of user characteristics in conversation modeling, to our knowledge, models that incorporate such signals for the task of predicting derailment remain unexplored.

One can extend on the classification gains of network formation in GCN models by modeling emotion and utterence recency and self user dependency in the graph and model architecture. Predicting emotions has only been introduced as an auxiliary task in studies [145, 85, 81]. These studies do not explicitly show the accuracy of the predictions. There may exist important features in the data such as recency and self-dependency that improve predication performance when taken into consideration that have been neglected in previous studies.

LLMs, in addition to being classification models, are generative models that can be used to generate next most likely utterances given a conversation. For the emotion and derailment predication problem we assume the text of the last utterance in the conversation to be omitted. An approach that has not yet been fully investigated is to generate this utterance, estimate the emotion of this utterance using some ERC model, and then to use this added information to inform these tasks.

Chapter 3

Conversation forecasting

This chapter describes the development a generalizable GNN-based representation to support conversation forecasting which is then expanded on in subsequent chapters to deal with future emotion estimation and conversation derailment. As identified in Chapter 1, the emotional trajectory structure in a conversation is at least dependent upon (i) self-speaker dependency, (ii) other-speaker dependency, and (iii) general knowledge. Thus for an AI architecture to be effective in this space it must provide mechanisms to encode such information. This chapter describes the Forecasting Graph Convolutional Network (FGCN), a graph-based architecture to capture these critical properties in a conversation. To achieve this, a generalized framework for the conversation forecasting problem is defined. Performance of the model on the next emotion prediction (nEMC) and the forecasting conversation derailment (FConvD) and tasks are presented and discussed in Chapter 4 and Chapter 5 respectively. In its most base form FGCN is not expected to perform as well as state of the art algorithms for the tasks considered here. Rather, FGCN is envisioned as a framework within which complex conversational properties can be embedded in a structured manner to explore the value of different information types in problems like conversational derailment.

3.1 Generalizing the conversation forecasting problem

In Chapter 2 the problems of conversation derailment and next emotion prediction were formally defined. Here a framework is defined for their solution. For a conversation of n turns, $\mathcal{C} = (T, U[, D_1, D_2, ...D_m])$ is a $m + 2$ tuple and $[\,] \equiv$ optional data. $T = (t_1, t_2, ..., t_n)$ is the sequence of text associated with the n turns in the conversation, $U = (u_1, u_2, ..., u_n)$ is the sequence of user IDs associated with the n turns in the conversation and D_j represents optional data sequences when and if available. Specific examples of this optional data include $CN = (cn_1, cn_2, ..., cn_n)$ the common sense knowledge of the n turns, $S = (s_1, s_2, ..., s_n)$ the public perception score associated with the n turns and $E = (e_1, e_2, ..., e_n)$ the labeled emotion associated with the n turns. For the i'th turn, t_i denotes its text, u_i denotes its user, cn_i denotes its estimated common sense knowledge, s_i denotes an optional public perception score (only available in one conversation derailment dataset), and e_i denotes an optional emotion label of the utterance (only available in emotion prediction datasets). The representations of these components are described in further detail below. For the problems of conversation derailment and next emotion prediction the goal is to estimate the label of the n'th turn given a conversation C up to and including the $n-1$'th turn including the actual values and optionally use an LLM estimate of the text sentence t_n for a given user ID u_n. A future investigation of the problem could also estimate the speaker ID u_n as well. Given this information the two tasks previously defined in Chapter 2 can be expressed as:

- **Next Emotion predication in conversation (nEMC):** Predict the emotion label l_{ep} at turn N of a given user u_n. Here $l_{ep} \in E$ denotes the label and E is the set of emotion labels available in the dataset.
- **Forecasting Conversation derailment (fConvD):** Predict if a conversation will derail or not at turn n. That is to predict the most likely label of l_{fd}. Here $l_{fd} =$

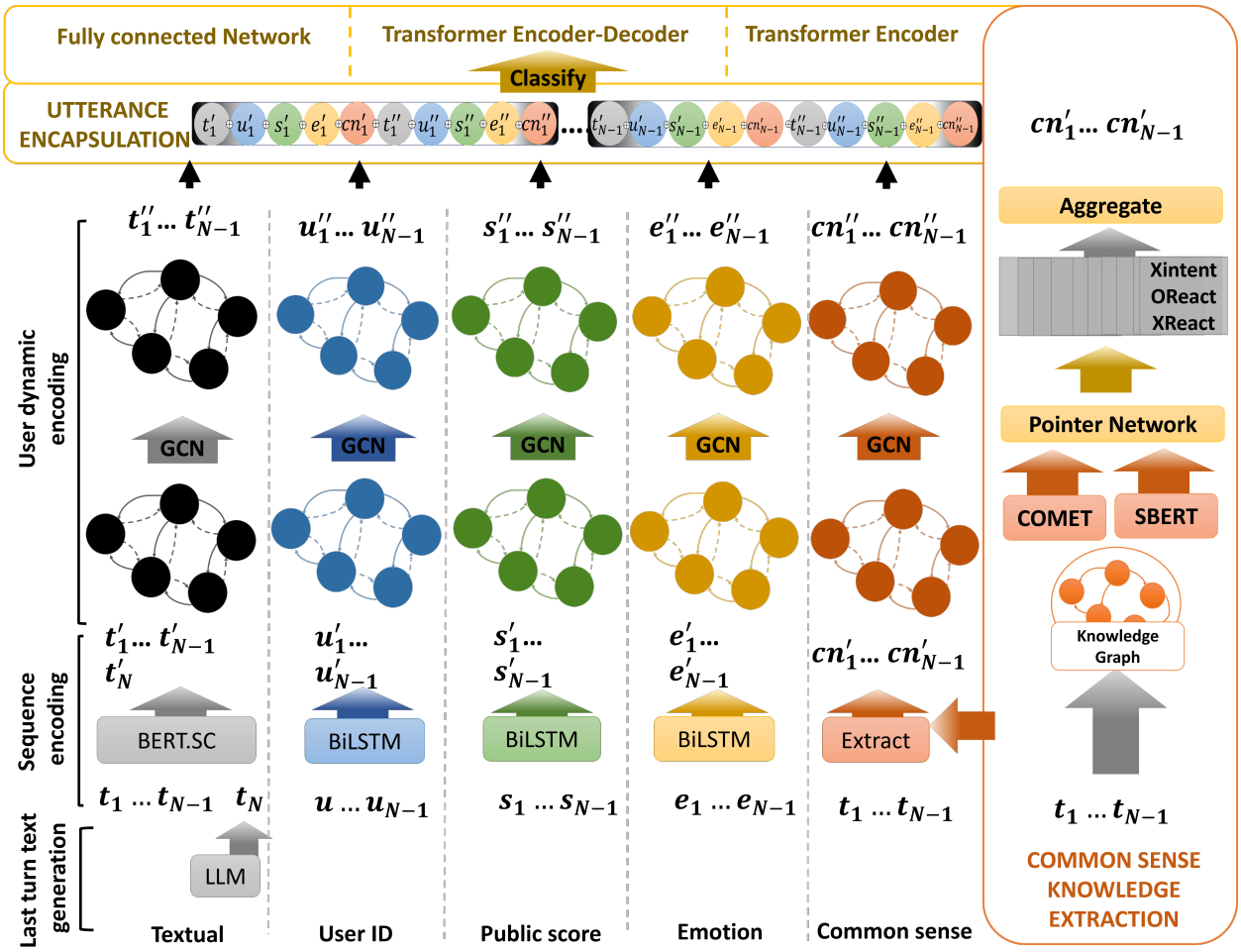


Figure 3.1: The FGCN model architecture. A sample encoding of a conversation of length 6 with three classes of optional data (S, CN and E) is shown. For each type of data the model separately (shown in the data layers) uses sequential encoding to populate the vertices in a graph where it undergoes GCN transformation. The results of the sequential encoder and user dynamics encoder are concatenated for classification. The details of graph construction in the user dynamics encoding are shown in Figure 3.2.

$\{civil, personal\ attack\}$ denotes the label of turn n. Further detail on how these labels are obtained can be found in the derailment forecasting data set description.

For both problems, nEMC and fConvD, the classification site is located at turn n. These

classifications are investigated using the entire context using information associated with turns $\{1, 2, ..., n-1\}$.

This work also investigates this classification with multiple window sizes starting from turn $\{1\}$, then turn $\{1, 2\}$ until $\{1, 2...n-1\}$ to classify turn n. That is to compare with previous work that investigates the effect of training on early subsets of the conversation on detection and earlier forecasting.

When studying the problem of nEMC and to take advantage of available emotion labels associated with each turn in the conversation, this work also investigates a number of additional classifications with different classification sites. Starting from turn $\{1\}$ to classify turn $\{2\}$, then turn $\{1, 2\}$ to classify turn $\{3\}$, until $\{1, 2, ..., n-1\}$ to classify turn n. This can not be investigated for the derailment forecasting fConvD problem as there do not exist labels for turns $\{1, 2, ..., n-1\}$.

For both the nEMC and fConvD problems, variants of the forecasting model that eliminate aspects of the full representation are created to identify the impact of different data sources and data structures on the models' ability to forecast. That is, to identify the level of model complexity that is most suitable for this problem and the parts that are most effective in addressing the problem. This analysis provides insights into future development of models for these problems in terms of data and data structure inclusion.

3.2 Forecasting model

This section describes the Forecasting Graph Convolutional Neural Network (FGCN) model for emotion predication and derailment forecasting visualized in Figure 3.1. As described earlier, there exist approaches (e.g. [42, 81, 125]) that use GNNs to address the emotion recognition problem. However the use of GNNs remains generally unexplored for the emotion prediction and derailment forecasting problems. This is explored here. Information used

in graph construction such as the parent-child structure of the conversation, localization of emotions (recency) and emotion persistence in speakers (self dependency) have not been directly encoded in the graphs reported in the literature. This information is leveraged in the graph model described here. Other than this, the main novelty in the model developed here is its extensible architecture that allows for additional related data to be integrated into the architecture and to investigate the relative performance of the information encoded in these various extensions.

The FGCN model consists of three levels, a sequential extraction level, a dynamics level, and a classification level. The first level in the model uses sequential encoding to extract related sequential features from the various conversation data. These features are then used to initialize the vertices in parallel graphs organized in layers for each type of data. These graphs go through graph neural network transformations at the user dynamics encoding level. In each data layer a method inspired by [42] is used to incorporate the data type in the model performance. The outcome of the sequential encoding and the GCN transformations of these encodings are concatenated and then passed to a classifier. In contrast to [42], for FGCN in each layer state-of-the-art sequential encoders, a more informed graph construction method and a more sophisticated classifier are used. The detailed structures are described below.

3.2.1 Sequential encoding

The input to the model consists of the text $T = (t_i)$, user ID $U = (u_i)$, and a combination of optional data, that includes the public perception score, $S = (s_i)$, the common sense knowledge $CN = (s_i)$ and the emotion label $E = (e_i)$ for each turn in the conversation $i \in (1, 2, ..., n-1)$, as introduced earlier and described in detail below:

- **Textual input** — This input consists of the encoding of the turn text t_i using BERT embeddings extracted after fine tuning as described in [68], resulting in the sequential

encoding of the text as the vector $t_i^{'}$. In addition, a LLM such as BERT, GPT, Gemini or Llama is used to generate the likely t_n utterance for a given user u_n for turn n and the encoding of this is included in the textual input for investigating the case of including an estimated textual value for t_n.

- **User ID input** — This input consists of the encoding of the user ID u_i, where each user has a unique vector; BiLSTM [136] sequential encoding is used to encode the user ID vectors $u_i^{'}$.

- **Public perception input** — This input is specific to data available in the conversation derailment dataset (see Appendix A). This consists of a public popularity score where the up-votes on a turn is subtracted by the down-votes on the same. To obtain the score vector s_i equal depth binning is used to capture discrete levels of popularity for positive scores and levels of unpopularity for negative scores. In practice seven levels are used; zero, three positive and three negative levels. BiLSTM [136] is used to encode utterance public perception vectors $s_i^{'}$.

- **Emotion input** — The input consists of an encoding of the utterance emotion label e_i, where each emotion class has a unique one hot vector; A one-hot vector is a $1 \times |\mathcal{E}|$ matrix (a vector of length $|\mathcal{E}|$) used to distinguish each $e \in \mathcal{E}$ from every $e \in \mathcal{E}$. The vector consists of 0's in all cells with the exception of a single 1 in a cell used uniquely to identify the class; e.g. emotion neutral could be represented by vector (1,0,0,0,0,0,0) in a set of 7 emotion classes. A BiLSTM [136] is used as the sequential encoder to encode the utterance emotion vectors $e_i^{'}$.

- **Common sense input** — ATOMIC [113] is used as the source of external knowledge. In ATOMIC, each node is a phrase describing an event. Edges are relation types linking one event to another. ATOMIC encodes triples of the form <event, relation type,

Event	Inference example	Inference
Person X pays person Y a compliment	PersonX wanted to be nice	xIntent
	PersonX will feel good	xReact
	PersonY will feel flattered	oReact

Table 3.1: An example of common sense inferences that can be extracted from a given utterance event that can be added as features to the utterance representation.

event>. There are a total of nine relation types in ATOMIC, of which three are used here: xIntent, the intention of the subject (e.g., 'to get a raise'), xReact, the reaction of the subject (e.g., 'be tired'), and oReact, the reaction of the object (e.g., 'be worried'), since they are defined as the mental states of an event.

Each input utterance text t_i, is compared with the text representing every node in the ATOMIC knowledge graph, and the K most similar ones are retrieved. A K value of 10 is used. SBERT [108] is used to compute the similarity score between an utterance and events. The top-K events are extracted and their intentions and reactions obtained, which are denoted as $\{e^{sI}_{i,k}, e^{sR}_{i,k}, e^{oR}_{i,k}\}, k = 1, ..., K$.

The knowledge generation model COMET [17], which is trained on ATOMIC is used to retrieve common sense knowledge. COMET takes t_i as input and extracts the knowledge with the desired event relation types specified (e.g., xIntent, xReact or oReact). The generated knowledge can be unseen in ATOMIC since COMET is essentially a finetuned language model. COMET is used to generate the K most likely events, each with respect to the three event relation types. The produced events are denoted as $\{g^{sI}_{i,k}, g^{sR}_{i,k}, g^{oR}_{i,k}\}, k = 1, ..., K$.

With the knowledge retrieved from ATOMIC, a pointer network [129] is constructed to select the commonsense knowledge either from SBERT or COMET. The pointer network calculates the probability of choosing the candidate knowledge source as:

$P(\Pi(t_i, e_i, g_i) = 1) = \sigma([t_i, e_i, g_i]W_\sigma)$, where $\Pi(t_i, e_i, g_i)$ is an indicator function with value 1 or 0, and $\sigma(t) = 1/(1 + exp(-t))$. σ is enveloped with the Gumbel Softmax [61] to generate a one-hot distribution. The integrated commonsense knowledge is expressed as: $cn_i = \Pi(t_i, e_i, g_i)e_i + (1 - \Pi(t_i, e_i, g_i))g_i$, where $cn_i = \{c^{sI}_{n,k}, c^{sR}_{n,k}, c^{oR}_{n,k}\}^K_k = 1$.

With the knowledge source selected, we proceed to select and aggerate the most informative knowledge using the method described in [149] to obtain cn_i.

In this work, we use single utterances to generate common sense knowledge for each utterance, leveraging frameworks like ATOMIC to incorporate common-sense reasoning and relational knowledge. While single utterances are effective for providing relevant replies, integrating prior context would further enhance the model's ability to maintain coherence, adapt to the user's emotional tone, and personalize interactions. ATOMIC aids in understanding potential consequences, beliefs, and intentions, even from isolated statements. However, future work could explore the integration of context with single utterances, allowing for a more dynamic and nuanced conversational experience while still maintaining the efficiency of single-response generation.

3.2.2 Embedding extraction

Text embeddings are a natural language processing (NLP) technique that converts text into numerical coordinates (called vectors) that can be plotted in an n-dimensional space. This approach enables pieces of text to be treated as bits of relational data, which we can then be used to train models. The text embedding used in this work are described below.

GloVe textual embeddings

GloVe [98] is an unsupervised learning algorithm for obtaining vector representations for words. Training is performed on aggregated global word-word co-occurrence statistics from

a corpus, and the resulting representations showcase interesting linear substructures of the word vector space. This work uses the Wikipedia 2014 + Gigaword 5 version with 6B tokens, 400K vocab, uncased with a 300 dimension vector to obtain embeddings.

BERT textual embedding

For extracting the word embedding with BERT only the last layer is used to obtain embeddings. The PyTorch framework is used to extract these embeddings. For the emotion prediction problem the "bertbase-uncased" pre-trained model [35] is used. For the forecasting derailment the BERT.SC [68] a model trained on the forecasting derailment dataset is used. The word embeddings have 768 dimensions.

RoBERTa textual embedding

For extracting the word embedding with RoBERTa only the last layer is used to obtain embeddings. The PyTorch framework is used to extract these embeddings. For the emotion prediction problem the 'roberta-base' pre-trained model [150] is used. The word embeddings have 768 dimensions.

GPT textual embedding

For extracting the word embedding with GPT the last layer only of the GPT model is used. The PyTorch framework is used to extract these embedding. The "GPT2" model [36] in the Transformers package is used. The word embeddings have 768 dimensions.

Gemini textual embedding

For extracting the word embedding with Gemini the Google platform API is used. The embeddings are extracted using "models/text-embedding-004". The "Text Embedding" are used rather than the "Embedding" model offered by Gemini API. This is an updated version

of the Embedding model that offers elastic embedding sizes under 768 dimensions. Elastic embedding generate smaller output dimensions and potentially save computing and storage costs with only minor performance loss [45].

Llama textual embedding

For extracting the word embedding with Llama the PyTorch framework is used. This work Uses "meta-llama/Llama-2-7b-hf" from the "transformers" package. The initial embedding dimension is 4096.

3.2.3 Response generation models

GPT response generation

For GPT response generation results, this work uses DialogGPT [144]. The DialogGPT repository is based on the huggingface pytorch-transformer and OpenAI GPT-2. the package contains a data extraction script, model training code and pretrained small (117M) medium (345M) and large (762M) model checkpoints. The GPT model in this work is fine tuned using the small pretrained model to reduce the chances of over fitting due to the limited size of our dataset. The model is fine tuned on the two datasets for 2 epoches with a max sequence length of 200. The AdamW optimizer, a learning rate of 1e-5, and a batch size of 4 is used for training and validation. In this thesis adapter layers are used for parameter-efficient fine-tuning (PEFT) of GPT. The adapter layers are inserted into each of GPT's transformer blocks, specifically between the attention and feed-forward layers.

BERT response generation

For the BERT response generation results, this work uses DialogBERT [46]. This work uses use "bertbase-uncased" which is fine tuned using the DialogBERT approach. Since the

base-size model is vulnerable to over-fitting in small datasets, The response generation model is trained with small-sized models. The small-size model reduces the 'bertbase-uncased' configuration to 6 transformer layers, has a hidden size of 256, and contains 2 attention heads (L=6, H=256, A=2). All of the experiments use the default BERT tokenizer. The model is fine tuned on the two datasets for 2 epochs with a max sequence length of 200. The AdamW optimizer, a learning rate of 5e-5, and a batch size of 4 is used for training and validation. In this thesis adapter layers are used for parameter-efficient fine-tuning (PEFT) of BERT. The adapter layers are inserted into each of BERT's transformer blocks, specifically between the attention and feed-forward layers.

Gemini response generation

For the Gemini response generation results, this work uses the Geminin API with gemini-1.0-pro [126]. This work uses the default supervised learning fine tuning setting for Gemini API. The model is fine tuned on the datasets for 2 epochs with a learning rate of 1.

Llama response generation

For the Llama response generation results this work uses Hugging Face fine-tuning with PEFT QLoRA with "Llama-2-7b". QLoRA pre-trained models are loaded to the GPU as quantized 4-bit weights. QLoRA is required since a single GPU is used to fine-tune the model. The model is fine tuned on the two datasets for 2 epochs with a max sequence length of 200. The AdamW optimizer, a learning rate of 2e-5, and a batch size of 4 is used for training and validation.

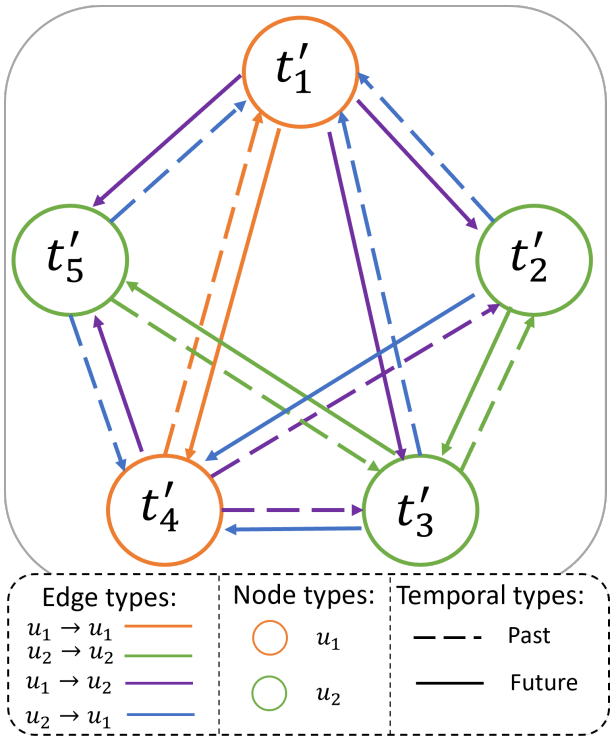


Figure 3.2: The set of edge labels has the relation types shown under user-to-user dependency. The graph in its original form is a fully connected graph, two vertices can have edges in both directions with different relations. This is used to represent the past (preceding) and future (succeeding) temporal dependency between the vertices, shown as temporal user dependency. Graph sparsification is used to enhance the graph construction based on the requirements of each problem (the derailment forecasting and next emotion prediction problem).

3.2.4 FGCN graph construction

After sequentially encoding T, U, and optionally CN, S and/or E for a given conversation, the output of the sequential encoder for each one of these input types $t_i^{'}$, $u_i^{'}$, $s_i^{'}$, $e_i^{'}$ and $cn_i^{'}$ as displayed horizontally in the bottom Figure 3.1 is used to initialize the vertices in the homogeneous graphs shown in the user dynamic encoding vertical layer in Figure 3.1. The vertices in the graphs represent turns in the conversation. Each parallel graph $G_x = (V, E, R, W)$, for each type of input $x \in \{t, u[, D_i]\}$, is constructed with vertices $x_i \in X$.

$r_{ij} \in E$ are the labeled edges between x_i and x_j, the edge labels (relations) $\in R$ and α_{ij} is the weight of the labeled edge r_{ij}. $0 \leq \alpha_{ij} \leq 1$, where $W = \{\alpha_{ij}\}$ is the set of all edge weights, $i \in \{1, 2, ..., n-1\}$ ($i \in \{1, 2, ..., n\}$ for T and U where t_n is the LLM estimated text informed by u_n) and j is the set of all neighboring vertices to x_i. The neighbouring vertices class varies by the relationship under consideration and is defined below.

For each conversation two or more graphs are constructed; including a text-based graph G_t and a user-based graph G_u. Optional additional graphs for each of the data types used are constructed if and when the data is available. Specifically, the public perception score-based graph G_s, the emotion vector graph G_e, and the common sense-based graph G_{cn} are constructed if the data is available. In G_t, each conversation turn is represented as a vertex x_i and is initialized with the textual sequentially encoded feature vector $t_i^{'}$, for all $i \in \{1, 2, ..., n-1\}$ and $t_n^{'}$ for the LLM estimated t_n informed by the value of u_n. In G_u each vertex is initialized with a utterance user ID vector $u_i^{'}$, for all $i \in \{1, 2, ..., N-1\}$ provided by the sequential encoder. u_n is given by the assumed user at turn n. Similarly, for G_s, G_e and G_{cn} the vertices are initialized with the corresponding data encoding. These are shown in the center part of Figure 3.1.

User to user relationship edge construction

The user-to-user relation R of an edge r_{ij} is set based on the user-to-user dependency between user u_i (of turn i) and user u_j (of turn j). For example, in Figure 3.2, there are two users, so the set of edge labels R has the relation types shown under user-to-user dependency. The graph in its original form is a fully connected graph, two vertices can have edges in both directions with different relations. This is used to represent the past (preceding) and future (succeeding) temporal dependencies between the vertices, shown as temporal user dependency. Graph sparsification is used to enhance the graph construction based on the requirements of each problem (the derailment forecasting and next emotion prediction problem). Sparsification

is based on the actual child parent relationship and/or to encode recency and self-dependency in the graphs. These are described in detail in Chapter 4 and Chapter 5 for the Next emotion prediction and conversational derailment problems. Similar to [42], the edge weights are initialized using a similarity based attention module. The attention function is computed such that, for each vertex, the incoming set of edges has a sum total weight of 1. The weights are calculated as $\alpha_{ij} = \text{Softmax}(\hat{x}_i \odot W_e)$ where $\odot$ is the Hadamard product and W_e is the vector of weights associated with the edges connecting vertex i to vertex j. The Hadamard product $\odot$ ensures that the element-wise multiplication is performed between the corresponding components of $\hat{x}_i$ and W_e, producing a vector whose values are passed through the softmax function to generate a normalized output that lies between 0 and 1, indicating the strength or relevance of the relationship between x_i and x_j.

3.2.5 Feature transformation

The sequentially encoded text features $t_i^{'}$, the user features $u_i^{'}$, and the optional features such as the public perception features $s_i^{'}$, the emotion features $e_i^{'}$ and the common sense features $cn_i^{'}$ of the graph network are transformed from the user dynamic independent feature vector into a user dynamic dependent feature vectors using the two-step graph convolution process employed by Ghosal et al. [42]. In the first step, a new feature vector is computed for each vertex in all created graphs by aggregating local neighbourhood information:

$$t_i^{''} \leftarrow \sigma(\sum_{r \in R} \sum_{j \in n_i^r} (\frac{\alpha_{ij}}{c_{i,r}} W_r \, t_j^{'} + \alpha_{ii} W_0 \, t_i^{'})),$$

$$u_i^{''} \leftarrow \sigma(\sum_{r \in R} \sum_{j \in n_i^r} (\frac{\alpha_{ij}}{c_{i,r}} W_r \, u_j^{'} + \alpha_{ii} W_0 \, u_i^{'})),$$

$$s_i^{''} \leftarrow \sigma(\sum_{r \in R} \sum_{j \in n_i^r} (\frac{\alpha_{ij}}{c_{i,r}} W_r \, s_j^{'} + \alpha_{ii} W_0 \, s_i^{'})),$$

$$e_i^{''} \leftarrow \sigma(\sum_{r \in R} \sum_{j \in n_i^r} (\frac{\alpha_{ij}}{c_{i,r}} W_r \, e_j^{'} + \alpha_{ii} W_0 \, e_i^{'})),$$

$$cn_i^{''} \leftarrow \sigma(\sum_{r \in R} \sum_{j \in n_i^r} (\frac{\alpha_{ij}}{c_{i,r}} W_r \, cn_j^{'} + \alpha_{ii} W_0 \, cn_i^{'})),$$

for $i = 1, 2, ..., n-1$, and $i = n$ for the textual input augmented by the LLM, since for $i = n$ we only the generated text is included. α_{ii} and α_{ij} are the edge weights, and n_i^r is the neighbouring indices of vertex i under relation $r \in R$. In the work reported here R is the set of user-to-user relationships. R is shown as edge types in Figure 3.2. $c_{i,r}$ is a problem specific normalization constant learned through gradient based learning. σ is an activation function such as ReLU, and W_r and W_0 are learnable parameters of the transformation. In the second step, a second local neighbourhood based transformation is applied to update the outputs of the first step, as follows:

$$t_i^{''} \leftarrow \sigma(\sum_{j \in n_i^r} W \, t_j^{''} + \alpha_{ii} W_0 \, t_i^{''})$$

$$u_i^{''} \leftarrow \sigma(\sum_{j \in n_i^r} W \, u_j^{''} + \alpha_{ii} W_0 \, u_i^{''})$$

$$s_i^{''} \leftarrow \sigma(\sum_{j \in n_i^r} W \, s_j^{''} + \alpha_{ii} W_0 \, s_i^{''}),$$

$$e_i^{''} \leftarrow \sigma(\sum_{j \in n_i^r} W \, e_j^{''} + \alpha_{ii} W_0 \, e_i^{''}),$$

$$cn_i^{''} \leftarrow \sigma(\sum_{j \in n_i^r} W\, cn_j^{''} + \alpha_{ii} W_0\, cn_i^{''}),$$

for $i = 1, 2, ..., n-1$ and $i = n$ for the estimated LLM textual input. W and W_0 are transformation parameters, and σ is the activation function. The transformations terminate after two turns. This two step transformation accumulates the normalized sum of the local neighbourhood.

3.2.6 Forecasting classifier

The sequential encoded vectors $t_i^{'}$, $u_i^{'}$, $s_i^{'}$ and $cn_i^{'}$, and the user dynamic encoded vectors $t_i^{''}$, $u_i^{''}$, $s_i^{''}$ and $cn_i^{''}$ are concatenated for each turn i in a conversation to form a turn information capsule d_i. This represents the turn and its associated sources of information. The information in the turn capsules is varied to understand the impact of each data type on forecasting derailment, these variants are described in Table 5.3.

The turn capsules d_i for $i \in \{1, 2...., n-1\}$ are concatenated to form a representation of the conversation $C^{'}$:

$$C^{'} = [d_1, d_2..., d_{n-1}]$$

In the cases where the use of an LLM augmented textual utterance is included as an estimated last utterance t_n in the conversation to investigate the impact of such an addition. The representation of the conversation $C^{'}$ including t_n is as follows:

$$C^{'} = [d_1, d_2..., d_{n-1}, t_n]$$

Finally, $C^{'}$ is fed to a classifier. The type of classifier architecture is varied in this work to explore the impact of different types of classifier. These classifiers are:

Simple classifier (S) – C' is fed to a classifier with a linear layer, a full connected network and a sigmoid activation function, as described by Ghosal et al. [42], to obtain the label $\hat{l}$ of each conversation C.

Transformer encoder only (E) – this is a transformer encoder classifier. For this classifier we preserve information about the boundaries between turns in C by inserting a [SEP] token between them. One [CLS] token is further added to the start of the input and one [SEP] token to its end. C' is fed to this encoder classifier to obtain the label $\hat{l}$ of each conversation C.

Transformer encoder-decoder (ED) – prediction labels $\hat{l}$ are obtained for each conversation C at each iteration turn i. A Transformer encoder-decoder is used to map the conversation capsule sequence to a sequence of prediction labels. Optimally this sequence would have the same prediction label at each turn iteration, that is the true conversation derailment or next emotion label. Each turn representation is converted to the [CLS] representation described above. We enforce a masking scheme in the self-attention layer of the encoder to make the classifier predict the label in an auto-regressive way, entailing that only the past utterances are visible to the encoder. This masking strategy, preventing the query from attending to future keys, better suits a real-world scenario in which the subsequent turns are unseen when predicting the label at a point of time. As for the decoder, the output of the previous decoder block is input as a query to the self-attention layer. C' is fed to this encoder-decoder classifier. Since we obtain a sequence of predictions at each turn i, we use the most predicated label in the generated label sequence as the classification result.

Variant	t	cn	u	s	e	**Capsule** d_i
T	✓	✗	✗	✗	✗	$[t_i^{'}, t_i^{''}]$
TU	✓	✗	✓	✗	✗	$[t_i^{'}, u_i^{'}, t_i^{''}, u_i^{''}]$
TE	✓	✗	✗	✗	✓	$[t_i^{'}, e_i^{'}, t_i^{''}, e_i^{''}]$
TS	✓	✗	✗	✓	✗	$[t_i^{'}, cn_i^{'}, t_i^{''}, cn_i^{''}]$
TCN	✓	✓	✗	✗	✗	$[t_i^{'}, cn_i^{'}, t_i^{''}, cn_i^{''}]$
TSU	✓	✗	✓	✓	✗	$[t_i^{'}, u_i^{'}, s_i^{'}, t_i^{''}, u_i^{''}, s_i^{''}]$
TUE	✓	✗	✓	✗	✓	$[t_i^{'}, u_i^{'}, e_i^{'}, t_i^{''}, u_i^{''}, e_i^{''}]$
TCNU	✓	✓	✓	✗	✗	$[t_i^{'}, u_i^{'}, cn_i^{'}, t_i^{''}, u_i^{''}, cn_i^{''}]$
TCNS	✓	✓	✗	✓	✗	$[t_i^{'}, s_i^{'}, cn_i^{'}, t_i^{''}, s_i^{''}, cn_i^{''}]$
TCNE	✓	✓	✗	✗	✓	$[t_i^{'}, e_i^{'}, cn_i^{'}, t_i^{''}, e_i^{''}, cn_i^{''}]$
TCNSU	✓	✓	✓	✓	✗	$[t_i^{'}, u_i^{'}, s_i^{'}, cn_i^{'}, t_i^{''}, u_i^{''}, s_i^{''}, cn_i^{''}]$
TCNEU	✓	✓	✓	✗	✓	$[t_i^{'}, u_i^{'}, e_i^{'}, cn_i^{'}, t_i^{''}, u_i^{''}, e_i^{''}, cn_i^{''}]$

Table 3.2: Variants of the turn information capsules at classification time. The sequential encoded vectors $t_i^{'}$, $u_i^{'}$, $s_i^{'}$ and $cn_i^{'}$, and the user dynamic encoded vectors $t_i^{''}$, $u_i^{''}$, $s_i^{''}$ and $cn_i^{''}$ are concatenated for each turn i in a conversation to form an information capsule.

3.2.7 Model variants

Input variants

To understand the effect of each type of input on emotion prediction and forecasting derailment, variants of the model are created where different combinations of the various types of input are included. These inclusions depend on their availability within the data being used. The following is a more detailed description of the variants. Each one of these variants are also evaluated with the optional LLM estimated t_N informed by u_N. Table 3.2 summarizes the variants.

FGCN-T — This variant uses conversations in the form $C = (T, U = \varnothing)$. FGCN-T is a degraded baseline to permit an ablation study of the importance of U in the baseline model. This variant of the model constructs only one graph using the output of the sequential encoder on the textual data $t_i^{'}$. It is created for all datasets, as all datasets contain the necessary

textual data. FGCN-T uses only the textual layer shown in Figure 3.1. At classification it uses $g_i = [t_i^{'}, t_i^{''}]$.

FCGN-TU — This variant uses conversations in the form $C = (T, U)$. FCGN-TU is the baseline of the study. This variant of the model constructs two graphs using the output of the sequential encoder; one for the textual output $t_i^{'}$ and the other for the user ID output $u_i^{'}$. FCGN-TU is created for all datasets, as all datasets contain the textual and user ID data. FGCN-TU uses the textual and user ID layer shown in Figure 3.1. At classification it uses $g_i = [t_i^{'}, u_i^{'}, t_i^{''} u_i^{''}]$.

FCGN-TE — This variant uses conversations in the form $C = (T, U = \varnothing, E)$. FCGN-TE uses the degraded baseline to permit an ablation study of the importance of the optional data E in the degraded baseline. This variant of the model constructs two graphs using the output of the sequential encoder, one for the textual output $t_i^{'}$ and the other for the emotion output $e_i^{'}$. FCGN-TE is created for emotion prediction datasets only, as only they contain the necessary textual and optional emotion data. FGCN-TE uses the textual and emotion layer shown in Figure 3.1. At classification it uses $g_i = [t_i^{'}, e_i^{'}, t_i^{''} e_i^{''}]$.

FGCN-TS — This variant uses conversations in the form $C = (T, U = \varnothing, S)$. It uses the degraded baseline to permit an ablation study of the importance of the optional data S. This variant of the model constructs two graphs using the output of the sequential encoder, one for the textual output $t_i^{'}$ and the other for the public perception score output $s_i^{'}$. FGCN-TS is created for CMV only, as only CMV contains the public perception data. FGCN-TS uses the textual and public score layer shown in Figure 3.1. At classification it uses $g_i = [t_i^{'}, s_i^{'}, t_i^{''} s_i^{''}]$.

FCGN-TCN — This variant uses conversations in the form $C = (T, U = \varnothing, CN)$. FCGN-TCN uses the degraded baseline to permit an ablation study of the importance of the optional data CN in the degraded baseline. this variant of the model constructs two graphs using the output of the sequential encoder, one for the textual output $t_i^{'}$ and the other for the common

sense output $cn_i^{'}$. It is created for all datasets, as all contain the textual and the common sense data is estimated. FGCN-TCN uses the textual and knowledge layer shown in Figure 3.1. At classification FCGN-TCN uses $g_i = [t_i^{'}, cn_i^{'}, t_i^{''} cn_i^{''}]$.

FGCN-TUS — This variant uses conversations in the form $C = (T, U, S)$. FGCN-TUS uses the baseline to permit an ablation study of the importance of the optional data S in the baseline. This variant of the model constructs three graphs using the output of the sequential encoder, one for the textual output $t_i^{'}$, the user ID output $u_i^{'}$ and the public perception score output $s_i^{'}$. It is created for CMV only, as only CMV contains the public perception data. FGCN-TSU uses the textual, user and public score layers shown in Figure 3.1. At classification FGCN-TUS uses $g_i = [t_i^{'}, u_i^{'}, s_i^{'}, t_i^{''} u_i^{''}, s_i^{''}]$.

FGCN-TUE — This variant uses conversations in the form $C = (T, U, E)$. FGCN-TUE uses the baseline to permit an ablation study of the importance of the optional data E in the baseline. This variant of the model constructs three graphs using the output of the sequential encoder, one for the textual output $t_i^{'}$, the user ID output $u_i^{'}$ and the emotion output $e_i^{'}$. It is created for emotion prediction datasets only, as only they contains the emotion data. FGCN-TSU uses the textual, user and public score layers shown in Figure 3.1. At classification FGCN-TUE uses $g_i = [t_i^{'}, u_i^{'}, e_i^{'}, t_i^{''} u_i^{''}, e_i^{''}]$.

FGCN-TUCN — This variant uses conversations in the form $C = (T, U, CN)$. FGCN-TUCN uses the baseline to permit an ablation study of the importance of the optional data CN in the baseline. This variant of the model constructs three graphs using the output of the sequential encoder, one for the textual output $t_i^{'}$, one for the user ID output $u_i^{'}$ and one for the common sense output $cn_i^{'}$. FGCN-TUCN uses the textual, user ID and knowledge layers shown in Figure 3.1. At classification FGCN-TUCN uses $g_i = [t_i^{'}, u_i^{'}, cn_i^{'}, t_i^{''} u_i^{''}, cn_i^{''}]$.

FGCN-TCNS — This variant uses conversations in the form $C = (T, U = \varnothing, CN, S)$. FGCN-TCNS uses the degraded baseline to permit an ablation study of the importance of

the combination of the two optional data CN and S in the degraded baseline. This variant of the model constructs three graphs using the output of the sequential encoder, one for the textual output $t_i^{'}$, the common sense output $cn_i^{'}$ and the public perception score output $s_i^{'}$. It is created for CMV, as only CMV contains the public perception data. FGCN-TCNS uses the textual, knowledge and public score layers shown in Figure 4.2. At classification FGCN-TCNS uses $g_i = [t_i^{'}, cn_i^{'}, s_i^{'}, t_i^{''}cn_i^{''}, s_i^{''}]$.

FGCN-TCNE — This variant uses conversations in the form $C = (T, U = \varnothing, CN, E)$. FGCN-TCNE uses the degraded baseline to permit an ablation study of the importance of the combination of the optional data CN and E in the degraded baseline. This variant of the model constructs three graphs using the output of the sequential encoder, one for the textual output $t_i^{'}$, the common sense output $cn_i^{'}$ and the emotion output $e_i^{'}$. It is created for the emotion prediction datasets only, as only they contains the emotion data. FGCN-TCNS uses the textual, knowledge and public score layers shown in Figure 4.2. At classification FGCN-TCNE uses $g_i = [t_i^{'}, cn_i^{'}, e_i^{'}, t_i^{''}cn_i^{''}, e_i^{''}]$.

FGCN-TUSCN — This variant uses conversations in the form $C = (T, U, S, CN)$. FGCN-TUSCN uses the baseline to permit an ablation study of the importance of the combination of both optional data S and CN in the baseline. This variant of the model constructs four graphs using the output of the sequential encoder, one for the textual output $t_i^{'}$, the user ID output $u_i^{'}$, the public perception score output $s_i^{'}$ and the common sense output $cn_i^{'}$. It is created for CMV only, as only CMV contains the public perception data. FGCN-TUSCN uses all of the layers shown in Figure 4.2. At classification FGCN-TUSCN uses $g_i = [t_i^{'}, u_i^{'}, s_i^{'}, cn_i^{'}, t_i^{''}u_i^{''}, s_i^{''}, cn_i^{''}]$.

FGCN-TUECN — This variant uses conversations in the form $C = (T, U, E, CN)$. FGCN-TUECN uses the baseline to permit an ablation study of the importance of the combination of both optional data E and CN in the baseline. This variant of the model constructs four graphs using the output of the sequential encoder, one for the textual output $t_i^{'}$, the

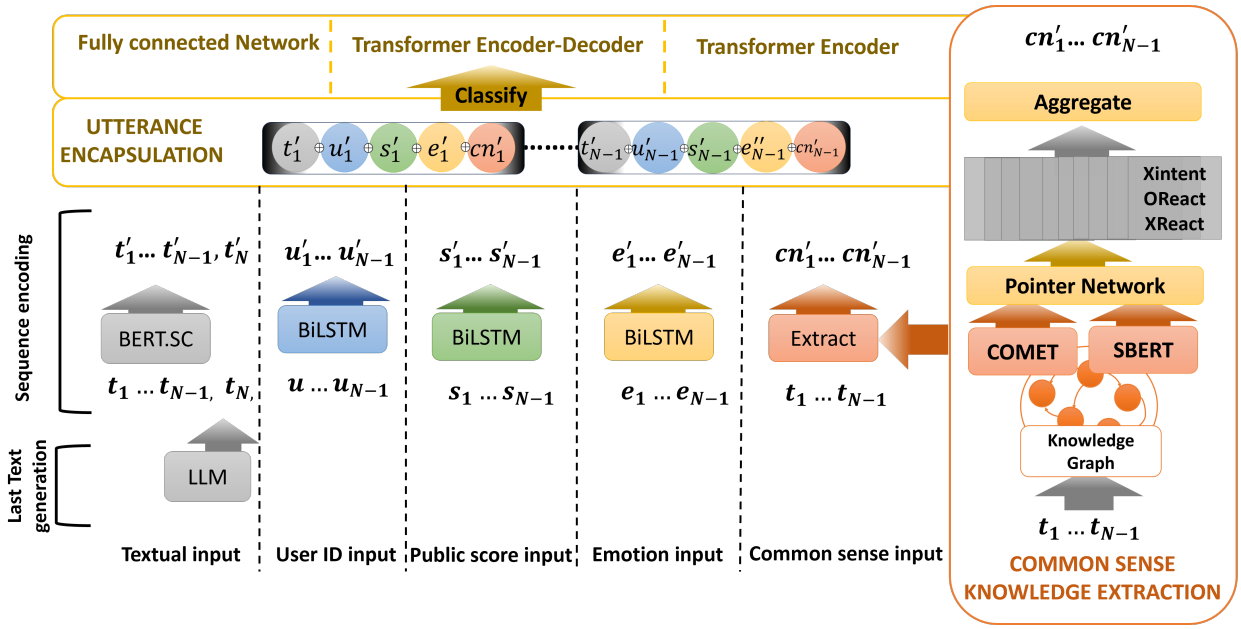


Figure 3.3: A variant of the FGCN model architecture using sequence encoding and no user dynamic encoding.

user ID output $u_i^{'}$, the emotion output $e_i^{'}$ and the common sense output $cn_i^{'}$. It is created for emotion prediction datsets only as they contain the necassery emotion data. FGCN-TUECN uses all of the layers shown in Figure 4.2. At classification FGCN-TUECN uses $g_i = [t_i^{'}, u_i^{'}, e_i^{'}, cn_i^{'}, t_i^{''} u_i^{''}, e_i^{''}, cn_i^{''}]$.

Architecture variants

To understand the effect of each level within the architecture on emotion prediction and forecasting derailment experiments are run with different architecture variants. Results of the above models on the architecture levels listed below are reported. Each is tested with a simple classifier and a transformer classifier.

- Sequence modeling: using only the sequential encoding for classification in the model. Figure 3.3 shows a variant of the model that uses sequential encoding and no user

dynamic encoding.

- Graph modeling: using the sequential and graph encoding for classification in the model.

Model sensitivity study

To understand the effect of recency on emotions and derailment predication the size of the past and future window for the edge construction of the graph model are studied. The model sensitivity to same user dependency is explored by eliminating or retaining same user edges in the graph construction. The length of the conversation being fed into the sequence and graph models is varied to understand how far ahead in the future can an effective prediction be made using the proposed architecture and models.

3.3 Experimental setup

3.3.1 Evaluation metrics

For the derailment task, fConvD, following prior work (e.g., [68, 50]), the performance of the models is reported in terms of accuracy (Acc), precision (P), recall (R), and F1-score (F1). For the derailment task the forecast horizon H introduced by [68] is reported, which is the mean of the number of turns in the future in which the first detection of derailment occurred. This is calculated for only the set of conversations that actually derail. If the model was unable to detect derailment the value is set to the length of the conversation n. This enables a more straightforward comparison with earlier work that use these metrics. A limitation of using the forecast horizon as a measure for early detection is that it does not account for false positives. For instance, the measure would fail if we decide to classify everything as a failure.

For the next emotion prediction task, nEMC, due to the highly imbalanced nature of the

datasets, the results are reported in terms of the **macro-averaged F1 score** that combines the per-class F1-scores into a single number, the classifier's overall F1-score. The F1-score for a single class is defined as F1-score=$\frac{2\times p\times r}{p+r}$, [95] where p and r are the precision and recall, respectively for a given class. Given the F1-scores for the classes individually, the macro-averaged F1 score for all classes combined is defined as:

$$F1_{macro} = \sum_{k \in class} w_k F1(class_k)$$

This is the weighted sum over every $class_K$ in the set of classes, where w_k is the weight given to $class_k$ and F1 ($class_k$) denotes F1-score of $class_k$ [95]. Different w_k strategies are defined in the literature and these are discussed below. For evaluation, the datasets are split into train and test sets using a 80/20 ratio.

3.3.2 Implementation

For both tasks, the following training and testing methods are performed using the training and testing splits reported in prior work that relates to the specific tasks. The FGCN results were ran on an Alienware Aurora R16 Gaming Desktop. The Desktop is equipped with an Intel Core™ i7, a NVIDIA GeForce RTX™ 4070 SUPER 16GB GDDR6X graphics card with 16GB memory.

Training

The training process is end-to-end, starting with sequential encoding, followed by user dynamic encoding, and culminating in classification. This approach utilizes the Adam optimizer and a cross-entropy loss function to optimize the model's performance at each stage. In this thesis static and dynamic training paradigms are used:

- **Static training**, For each conversation a single training instance with all available turns

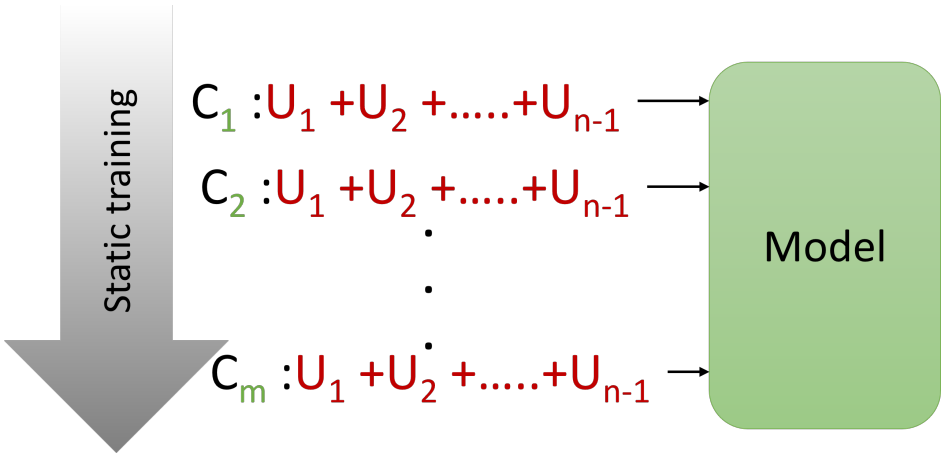


Figure 3.4: An illustration of the static training process. In static training, every utterance capsule U_i for each i from 1 to $n-1$ is used for each conversation instance C_j for every j in the training split provided to the model in the training process. m denotes the number of conversations in the training set.

is used as input. Figure 3.4 illustrates the static training process. In static training every turn capsule U_i for every i from 1 to $n-1$ is used for each conversation instance C_j for every j in the training split provided to the model in the training process.

- **Dynamic training**, Multiple instances are used of each conversation, by varying the last turn used in each training instance. This approach maximizes the use of the available datasets, and enables an examination of the effect of the temporal horizon in conversations. Prefix portions of each conversation is used. For a conversation with a length of n turns, prefix versions of this conversation of lengths shorter than n are used. In addition, by providing a mechanism to maximize this data usage, we use the predicted values of the text and user to explore long-term conversational forecasting, including derailment forecasting to a final iteration including the estimated LLM textual input t_n. We denote all dynamically trained models with an added "+" at the end of the model name. Figure 3.5 illustrates the dynamic training process.

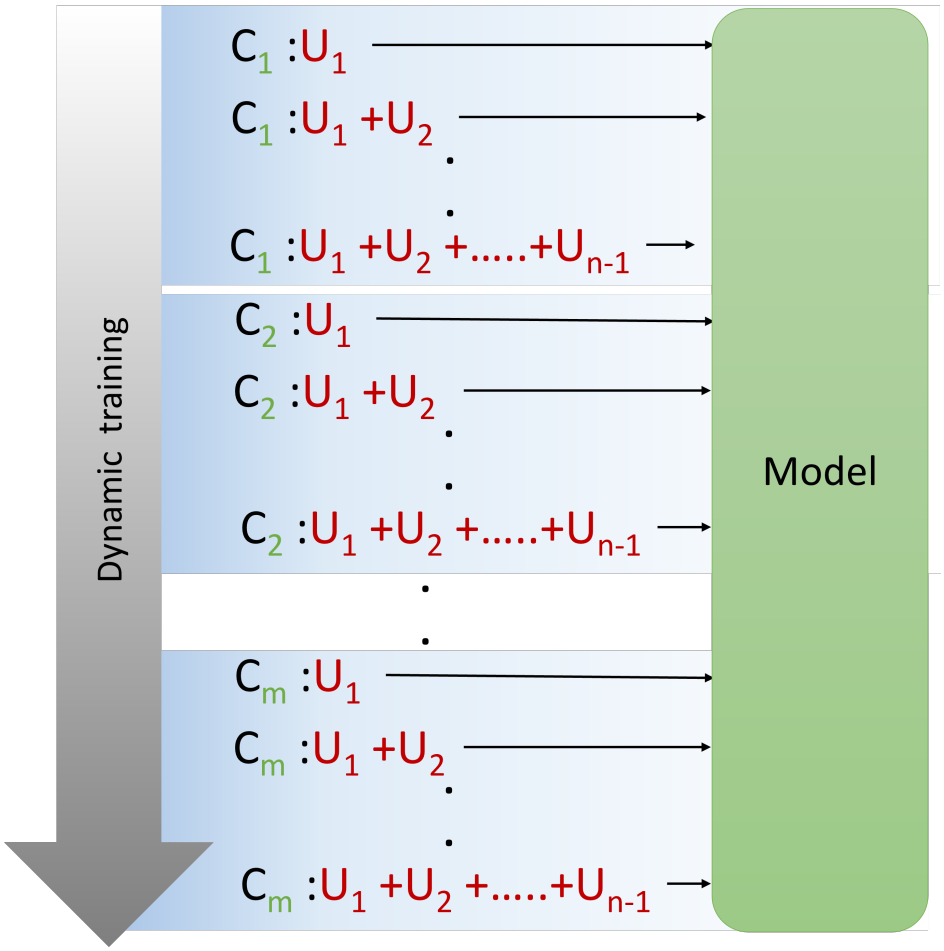


Figure 3.5: An illustration of the dynamic training process. In dynamic training, the utterance capsules U_i included for each conversation instance C_j varies. The first instance for C_i will contain U_1 only, the second instance will contain U_1 and U_2, the third instance will contain U_1, U_2 and U_2, and so on until the last instance that contains the entire context from U_1 to U_{n-1}. This is preformed for each conversation instance in the training set. m denotes the number of conversations in the training set.

Testing

For the derailment forecasting task, fConvD, at inference time, the model is evaluated dynamically following prior work methodology. That is, using turn information up to turn

$k-1$ as input, a prediction $\hat{l}_k$ is made at turn k. Then, using turn information up to turn k, a prediction $\hat{l}_{k+1}$ is made at turn $k+1$, and so on until $n-1$ predictions have been accumulated. In addition to a final prediction including the estimated LLM textual input t_n. The overall predicted label is the label with most votes in the set of predictions for a given conversation.

For the next emotion prediction task, nEMC, at inference time, the model is evaluated statically following prior work methodology. That is, for each conversation a single testing instance with all available turns $1-(n-1)$ as input is used to generate a predicted next emotion.

3.4 Summary

This chapter provides a detailed description of the model, data encapsulation, training and evaluation associated with the tasks at hand. Two different main conversation architectures (sequence and graph) and two different training mechanisms (static and dynamic) are described. The training data is encapsulated in various ways ranging from two or more different types of data, as shown in Table 3.2. Figure 3.6 provides an illustration of the evaluations paths provided in this thesis. As shown by the yellow path in Figure 3.6, after encapsulating the data to be investigated the combined data is either provided to a sequence model or a graph model. The blue path in Figure 3.6 shows that if the data is given to a graph model, it can be constructed as a complete graph or go through sparsification based on the characteristics of the task investigated. The green and orange paths in Figure 3.6 show that the complete and sparsified graph features can be classified using a simple, transformer-encoder or transformer-encoder-decoder classifier. Each one of these evaluation paths can be be trained dynamically or statically.

Given these model variations and evaluation paths, Chapter 4 evaluates the performance of the various architectures for the problem of derailment forecasting and Chapter 5 evaluates

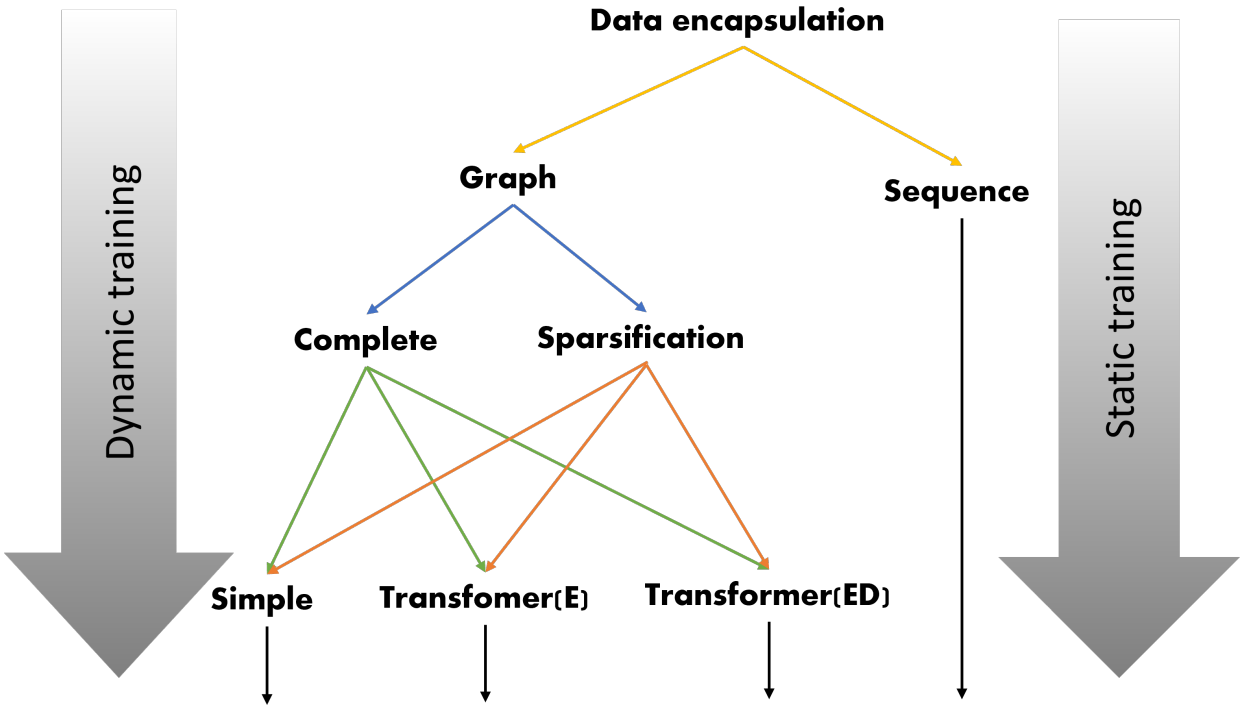


Figure 3.6: An illustration of the evaluation paths. As shown by the yellow path, after encapsulating the data to be investigated, the data is provided to either a sequence model or graph model. The blue path shows that if the data is given to a graph model, it can be constructed as a complete graph or go through graph sparsification. The green and orange paths show that complete and sparsified graph features can be classified using a simple, transformer-encoder or transformer-encoder-decoder classifier. Each one of these evaluation paths can be be trained dynamically or statically.

the performance of the various architectures for the problem of next emotion prediction.

Chapter 4

Next emotion prediction in conversations (nEMC)

4.1 Introduction

Next emotion prediction in a conversation (nEMC) aims to predict how the textual content of other parties in a conversation affect the emotion of speakers in an ongoing dialogue. This task demonstrates a conversational model's ability to predict the emotion trajectory of a conversation before the actual (future) turn text becomes available. This task explores the emotional influence from the speakers' perspective. In this task, it is critical to integrate the turn information of the speakers and model the emotional interaction between the speakers.

Figure 4.1 shows an example conversation, coming from the popular DailyDialogue benchmark dataset. The goal here is to predict the emotion label l of the n^{th} turn's text given a conversation $\mathcal{C}$ from turn 1 up to and including turn $n-1$ including the user, text and emotion information of these turns. This chapter first explores the use of turn emotion data in the conversation on predicting the next emotion through sequence, self dependency and recency modeling of the data using both a sequence-based and a graph-based model. Results

from this exploration informs model architecture decisions such as the graph sparsification used in the graph construction of the predicting/forecasting model used in this work. This Chapter then compares the resulting forecasting model with graph sparsifiaction and with different input data combinations with baselines for the next emotion prediction problem. Section 4.2 reviews various machine architectures and data structures that might be utilized to address the nEMC problem. Critical design questions here include

- What sort of architecture is most appropriate, a GCN- or a sequence-based approach?
- What types of data are most effective for nEMC? Text only, Text+emotion or Emotion only?
- What portion of the conversation is most important in terms of nEMC? Is it necessary to use a large portion of the conversation? Or are smaller, more focussed portions more effective. Is same-speaker information important, or is other-speaker information, or both?
- From a theoretical point of view, would non-causal information be helpful or is it possible to obtain good predictive models using only causal data?
- The impact of different text encoding structures on nEMC performance.

After considering these problems in detail, in Section 4.3 this chapter develops a solution to nEMC informed by the results of Section 4.2. The critical goal here is not so much to build an algorithm that provides the best possible nEMC algorithm but rather to identify the characteristics that are important in the design of algorithms that can predict emotional trajectories in conversations. A problem that is further addressed in Chapter 5.

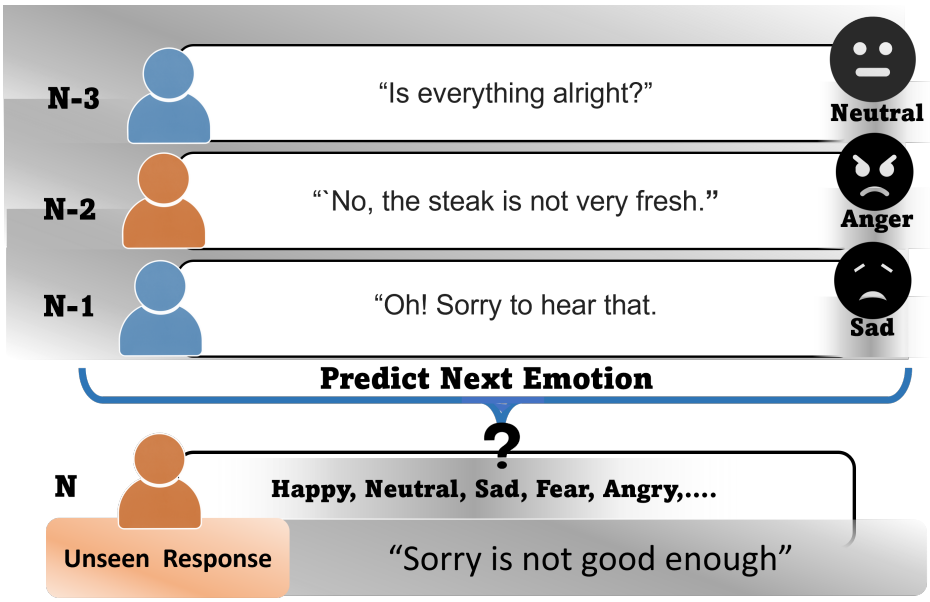


Figure 4.1: The problem of Next Emotion Prediction

4.2 Emotion data analysis

The data available for the next emotion prediction in conversation problem nEMC includes the emotion labels for each previous turn in the conversation, not just the emotion label of the n'th turn in the conversation. This allows for modeling of the problem based on the emotional trajectory of all speakers throughout the conversation. The outcome of this emotional data analysis provides guidelines for the use of emotional information (e.g. recency and user self-dependency) to inform conversation structuring within a general conversation forecasting model.

To extract beneficial information on the relationship between emotion data and conversation classification, the nEMC problem is approached by modeling three dimensions inherently connected to evoked emotions in dialogues; (**i**) *sequence modeling*, (**ii**) *self-dependency modeling*, and (**iii**) *recency modeling*. These dimensions are elaborated on below. The foresting model described in Chapter 3 is applied to nEMC using the information identified from this

analysis. Here the problem is treated in a discrete form given the structure of the available datasets. The problem of a probabilistic version of the problem acting on a distribution of emotional state probabilities is left for future work.

4.2.1 Emotional data modeling

Sequence Modeling

Given a conversation $\mathcal{C}$ of $n-1$ turns, the goal is to predict the emotion associated with the next turn e_n given knowledge of the user at turn n. Three cases that treat the sequence of conversation turns as a *sequence of emotions*, *sequence of texts* or *sequence of (emotion, text) pairs* are considered.

- **Sequence of emotions**: This analysis uses the sequence of emotion labels $\mathcal{C}^{\langle e \rangle} = (e_1, e_2, ..., e_{n-1})$, $e \in \mathcal{E}$ to predict e_n. Each of the e_i is represented as a one-hot encoded vector of size $1 \times |\mathcal{E}|$, with each dimension representing one of the emotion classes $\mathcal{E}$.

- **Sequence of texts**: This analysis utilizes the sequence of text turns $\mathcal{C}^{\langle t \rangle} = (t_1, t_2, ..., t_{n-1})$ to predict e_n.

- **Sequence of (emotion, text) pairs**: This analysis utilizes the sequence of (text, emotion) pairs of turns $\mathcal{C}^{\langle t,e \rangle} = (\langle t_1, e_1 \rangle, \langle t_2, e_2 \rangle, ..., \langle t_{n-1}, e_{n-1} \rangle)$ to predict e_n. The sequence of emotion labels $\mathcal{C}^{\langle e \rangle} = (e_1, e_2, ..., e_{n-1})$ and the sequence of texts $\mathcal{C}^{\langle t \rangle} = (t_1, t_2, ..., t_{n-1})$ are constructed and provided separately into the model, then combined at classification time.

Self-dependency modeling

The sequence models presented above are agnostic to the the identity of the speaker. However, the nature of an evoked emotion might be speaker dependent. Does the model rely on the

text used by of the speaker being modeled (*self-dependency*), or on the text used by other participants in the conversation (*other-dependency*)? To address this question, this analysis designs and trains variants of the sequence models that explore the nature of self-dependency and other-dependency to evoked emotions.

Consider a group of m people $U = \{u_1, u_2, \ldots, u_m\}$ that participate in a group conversation. Given a conversation $\mathcal{C}$ and a specific speaker u representing *self*, we can define sequences of turns $\mathcal{C}_u$ and $\mathcal{C}_{\mathcal{O}}$, such that $\mathcal{C}_u$ represents all turns by u and $\mathcal{C}_{\mathcal{O}}$ represents all turns from any of the other speakers $\mathcal{O} = U \setminus u$. Note that $\mathcal{C} = \mathcal{C}_u \cup \mathcal{C}_{\mathcal{O}}$ and $\mathcal{C}_u \cap \mathcal{C}_{\mathcal{O}} = \phi$. After running the prediction models (see Section 4.2.2) on the input conversation $\mathcal{C}$ and prior to final classification, we obtain a representation of the conversation $\mathcal{C}' = \mathcal{C}'_u \cup \mathcal{C}'_{\mathcal{O}}$, where $\mathcal{C}', \mathcal{C}'_u$, and $\mathcal{C}'_{\mathcal{O}}$ are the representations of $\mathcal{C}, \mathcal{C}_u$, and $\mathcal{C}_{\mathcal{O}}$ after running the prediction and prior to classification. This separation of textual input would allow us to study the impact of the text of the same user, other user and all users in the conversation on the prediction of user's next emotion and draw comparisons on performance.

Recency modeling

The sequence models presented so far assume that all of the $n-1$ turns of a conversation $\mathcal{C}$ inform the sequence model. However, the nature of an evoked emotion might be influenced more directly by more recent turns of the conversation [39]. Does the model rely on all turns of the conversation or focus on the more recent ones? To address this question, this analysis designs and trains variants of sequence models that explore the *temporal dimension* of evoked emotions. Formally, the length w of a *temporal look back window* is defined to control how far the sequence extends into the past upon which estimation relies explicitly. $w = 0$ being no look back, $w = 1$ being one look back and so on. A look back (LB) is of one turn in the entire conversation given same and other user turns. However when studying self look backs SLB, the look back window is defined in term of the number of turns of the same user only.

When studying and other look backs OLB the look back window is defined in term of the number of turns of the other users only.

4.2.2 Neural network architectures for interpretation

To analyze the previously mentioned data modeling, neural network architectures that incorporate the three modeling dimensions, including a *sequence model* and a *graph convolutional network model* are developed to derive insights on conversation modeling from the available emotional data. Each of these models are described in detail below. The choice of model architectures and look back structure enable an investigation into the ability of different data structures and different machine structures to address this problem.

BiLSTM-nEMC

BiLSTM-nEMC is a BiLSTM-based model introduced to capture the sequence of utterances in a dialogue. Bidirectional LSTMs (BiLSTMs) [136] are enhanced versions of LSTM models which are well-suited to classifying and making predictions based on sequence data and are known to outperform traditional recurrent neural networks (RNNs) [136]. BiLSTMs enable additional training by traversing the input data twice (i.e., they exploit future and history context together at once). Typically, BiLSTM-based modeling offers better predictions than regular LSTM-based models [136], making this architecture a sensible choice for the nEMC problem.

Text sequences of each turn are pre-processed (removal of punctuation and stopwords, lower-casing, and lemmatization (reducing a word to its base or root form)) and converted into a vector representation using GloVe embeddings [98]. *Emotion sequences* are converted into a a one-hot vector representation before being provided to the neural network. For *(emotion, text) pair sequences*, emotion and text sequences are treated separately (as if in

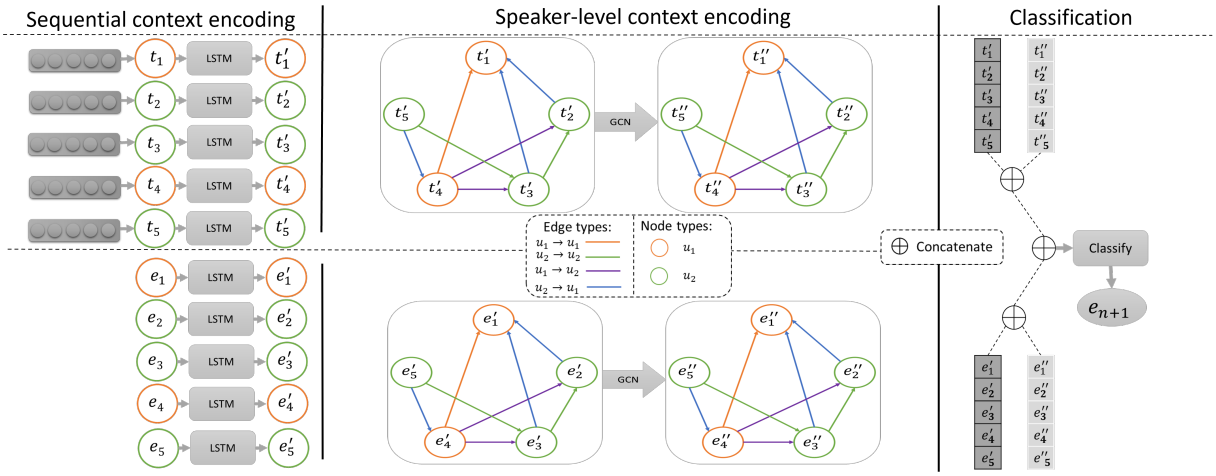


Figure 4.2: Overview of the DGCN-nEMC model architecture.

isolation) and they are concatenated at a later stage, just prior to the final classification layer. To leverage u's self-dependency in predicting the emotion of its next turn a model is trained on the sequence of turns $\mathcal{C}_u$ (self-dependency) and classified using $\mathcal{C}'_u$. To explore the influence of other speakers' information in predicting the emotion of u's next utterance, a model is trained on the sequence of utterances $\mathcal{C}_\mathcal{O}$ (other-dependency) and classified using the emotion labels using $\mathcal{C}'_\mathcal{O}$.

In terms of implementation, vectors (word embeddings) are provided into a Time Distributed Layer (which applies the layer wrapped inside it to each timestep, in this case the turn timestep) followed by a Flatten Layer (which is used to reshape the input data into a one-dimensional vector so that it can be provided to the subsequent layers) before passing them to a BiLSTM with an attention layer. `ReLu` is used as the dense layer activation function and `softmax` as the output layer's activation function.

DGCN-nEMC

DGCN-nEMC is an extension of the DialogueGCN model [42], designed to address the nEMC problem. DialogueGCN is an ERC classifier that given conversation text turns as input,

classifies each text turn into one of a given set of emotion classes. DialogueGCN incorporates speaker information by modeling multi-party conversations as a graph, where nodes represent turns and edges (connecting two turns) represent the speaker type relationship. Details on the specifics of DialogueGCN can be found in [42]. At the time of this data analysis DialougeGCN provided state of the art performance on the ERC problem.

This work introduces DGCN-nEMC to capture the sequence of utterances and the network formation of multi-party dialogues. DGCN-nEMC [7] is an early version of the graph FGCN model (described in Chapter 3), used to study emotion data. DGCN-nEMC takes as input a combination of text and/or emotion signals of conversation utterances, and predicts the emotion class of the next turn, e_n, in a conversation of size $n-1$. The high level architecture of the DGCN-nEMC model is provided in Figure 4.2. The input of the model is a one-hot encoded emotion vector and text vectors (GloVe embeddings) of the conversation. BiLSTM is used as the base model in the sequential context encoding and uses the same speaker-level context encoding as DialogueGCN does. The speaker-level context encoding creates a graph of turns. The nodes are instantiated with features extracted using sequential context encoding. The turns are connected through edges that reflect speaker relationships. The features associated with the text belonging to a turn are transformed using a graph convolutional network. The sequential and speaker level context encoding output features are concatenated prior to classification. In a similar manner to BiLSTM-nEMC, the *(emotion, text)* sequences are processed separately (as a sequence of emotions and a sequence of texts), and their output is concatenated just prior to the final classification layer. To leverage u's self-dependency in predicting the emotion of its next utterance, the model is trained on the sequence of turns in $\mathcal{C}$ (the entire conversation, so as to create a graph of the turns), the $\mathcal{C'}_u$ part of the conversation is extracted from $\mathcal{C'}$. This extracted part contains the transformed representation of turns for same user only taken from the transformed representation of all turns in the conversation. This same user extracted transformed turn features are used to classify on

$\mathcal{C}'_u$ (self-dependency). Similarly, to explore the influence of other speakers' information in predicting the emotion of u's next turn, the model is trained on the sequence of turns $\mathcal{C}$, the $\mathcal{C}'_{\mathcal{O}}$ part of the conversation is extracted from $\mathcal{C}'$. This extracted part contains the transformed representation of turns for all other users taken from the transformed representation of all turns in the conversation. This other user extracted transformed turn features are used for classification using $\mathcal{C}'_{\mathcal{O}}$ (other-dependency).

When DGCN-nEMC constructs the conversation graph in the speaker-level context encoding, the parameters, pw (past window) and fw (future window), control how far in the past or future in the conversation to include when creating the edges between turn nodes. A pw or fw of n corresponds to a window of n turns. A preliminary analysis varying the values of these parameters found that $pw = 3$ and $fw = 0$ provide the best results on average. These values are used here. Results for varying pw are presented in Section 4.2.14.

4.2.3 Combining data modeling and neural network architectures

Based on the aforementioned modeling dimensions and model architectures the following look back (LB) models are defined.

- ***w*LB**: A sequence model that is agnostic to the identity of the speaker and considers a temporal window of length w turns in $\mathcal{C}$.

- ***w*SLB**: A sequence model that considers self-dependency on speaker $u \in U$ (i.e., only utterances of speaker u are used) and a temporal window of length w in $\mathcal{C}_u$ for BiLSTM-nEMC or $\mathcal{C}'_u$ for DGCN-nEMC. Only the turns of the same user are used.

- ***w*OLB**: A sequence model that considers other-dependency (i.e., only utterances of speakers $\mathcal{O} = U \setminus u$ are used) and a temporal window of length w in $\mathcal{C}_{\mathcal{O}}$ for BiLSTM-nEMC or $\mathcal{C}'_{\mathcal{O}}$ for DGCN-nEMC. Only the turns of other users are used.

Dataset	Type	# Classes	# Utterances
DAILYDIALOG	dyadic	7	103.0k
IEMOCAP	dyadic	11	6.8k
MELD	group	7	13.7k
EMOTIONLINES:FRIENDS	group	8	14.5k

Table 4.1: Details of the datasets.

Each model is instantiated for each of the three types of sequence models (*emotion*, *text*, or *(emotion, text)*), and processed through either BiLSTM-nEMC or DGCN-nEMC.

4.2.4 Data analysis: model evaluation

This section evaluates the performance of the models in this data analysis. It also examines the sensitivity of the models to the choice of the classifier, word embeddings, and the use of same/other speaker edges used in the graph based model DGCN-nEMC. Before presenting the results of the data analysis and modeling, details of the datasets employed, the evaluation metric and the evaluation scenarios in this data analysis are provided.

Datasets

The datasets used in these experiments can be categorized as either being *dyadic conversations* (i.e., dialogues involving two speakers) or *group conversations* (i.e., dialogues involving multiple speakers, two or more). The dyadic conversations datasets, are (i) DAILYDIALOG [79], which consists of two-speaker dialogues pertaining to conversations about daily life, and (ii) IEMOCAP [19], which consists of dyadic sessions with actors performing emotional improvisations or scripted scenarios. For group conversations, the datasets are (iii) MELD [103], which consists of multi-speaker dialogues from the comedy show Friends, and (iv) EMOTIONLINES:FRIENDS [55]. The details of the datasets are summarized in Table 4.1. See also Appendix A.

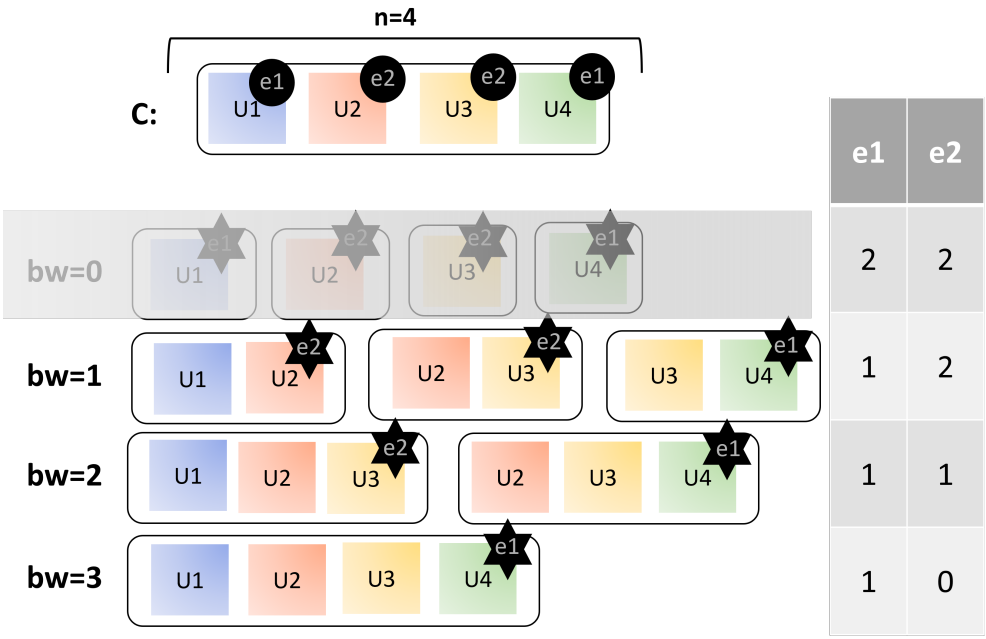


Figure 4.3: An illustration of an example conversation augmentation and the redistribution of the emotion labels for a generic conversation of length $n = 4$. With $bw = 0$ there is no context for the emotion to be predicated, so this case is not used for augmentation. In the case where $bw = 1$, the conversation is augmented into three conversation each conversation has one previous utterance in the context. In the case where $bw = 2$ the conversation is augmented into two conversation each conversation has two previous utterance in the context. In the case where $bw = 3$ the conversation has the original three previous utterance in the context and is not augmented. The table shows that this results in different distributions of classified emotion class labels (shown in the stars) for different bw for different window sizes.

Reconstructed datasets

To train the sequence models that employ a temporal look back window of size w, dialogues should include at least n turns, where $n \geq w$ to fully express turns of length w. From the original datasets, subsets are constructed to meet this requirement. For example, for the back window $bw = 3$, dialogues are extracted from the original dataset that are of at least

	Look back window size (bw)						
	$bw = 0$	$bw = 1$	$bw = 2$	$bw = 3$	$bw = 4$	$bw = 6$	$bw = 8$
neutral	85572	73997	62907	52242	42127	26675	15639
happiness	12885	11829	10464	9212	7836	5590	3623
surprise	1823	1708	1436	1143	894	544	302
sadness	1150	1036	815	698	509	341	191
anger	1022	855	760	607	509	327	186
disgust	353	291	230	184	128	73	37
fear	174	145	131	104	87	57	34

Table 4.2: Emotion class label distributions on the reconstructed DAILYDIALOG, for varying values of bw.

four turns in length ($n \geq 4$). Figure 4.3 illustrates an example of conversation augmentation and the redistribution of the emotion labels for a generic conversation of length $n = 4$. With $bw = 0$ there is no context for the emotion to be predicated, so this case is not used for augmentation. In the case where $bw = 1$, the conversation is augmented into three conversations where each conversation has one previous utterance in the context. In the case where $bw = 2$ the conversation is augmented into two conversation each conversation has two previous utterance in the context. In the case where $bw = 3$ the conversation has the original three previous utterance in the context and is not augmented. The table in the Figure 4.3 shows that this results in different distributions of classified emotion class labels (shown in the stars) for different bw for different window sizes. Table 4.2 shows this effect for the case of the DAILYDIALOG dataset, where the emotion class label distribution is given for varying values of bw. This work recognizes that this data augmentation makes the dataset less independent, but this a compromise given the limited amount of labelled data that is available and used for data analysis purposes only.

Evaluation metrics

Due to the highly imbalanced nature of the datasets, the results are reported in terms of the **macro-averaged F1 score** that combines the per-class F1-scores into a single number, the classifier's overall F1-score. F1-score for a single class is defined as $F1 - score = \frac{2 \times p \times r}{p+r}$, [95] where p and r are the precision and recall, respectively for a given class. Given the F1-scores for the classes individually, the macro-averaged F1 score for all classes combined is defined as

$$F1_{macro} = \sum_{k \in class} w_k F1(class_k)$$

for every $class_j$. Where w_{class_j} is the weight given to $class_j$ and F1 denotes F1-score [95]. The weights are calculated using smooth weights as described later. For evaluation, the datasets are split into train and test sets with a 80/20 ratio.

4.2.5 Computational considerations

For each one of the experimental results in this work, the datasets are split 80% training and 20% testing. Training took place over 30 epochs and maximum macro-averaged F1 score value reported. All experiments took place on a Dell Alienware Aurora R7 i7 desktop with a Nvidia 1080Ti RTX Graphic card.

4.2.6 Implementation of data analysis

This work utilizes a number of pre-existing software packages. For the textual input Glove's 840B token and 300d vector embeddings are used. Pre-processing of the text input is done using (i) the Keras tokinzer and sequence padding, and (ii) NLTK stopwords and lemmatizer. For Metric reporting Sklearn metrics are used. For the models and neural network layers, Keras 2.3.1 and PyTorch 1.0 are used.

When running DGCN-nEMC, the parameters for cuda and nodal attention are set to False. All other parameters use the default values of the original implementation of DialougeGCN as described in [42] .

4.2.7 Data analysis evaluation scenarios

. This data analysis seeks answers to the following questions:

- Which of the three types of sequence models introduced earlier; *emotion sequence*, *text sequence*, or *(emotion, text) sequence* provides better performance?
- Given the different architectures; graph-based model (DGCN-nEMC) and sequential model (BiLSTM-nEMC) are evaluated, which provides the best performance?
- What are the effects of incorporating self-dependency (versus other-dependency) on the accuracy of the prediction models ?

4.2.8 Handling imbalanced classes

For each of the datasets, the class distribution was examined to determine the nature of the dataset. Typically, the label distribution in emotion datasets is quite imbalanced, and that remains true for conversation datasets labeled with emotion categories as well. Using BiLSTM-nEMC, for the DAILYDIALOG dataset three strategies for handling the imbalanced data were explored using the same training setup.

1. **Oversampling (OS)**: All minority classes are over-sampled to match the support of the majority class (i.e., *neutral*) using sampling with replacement.
2. **Class weights (CW)**: Assuming $L = l_1, ..., l_k$ to be the set of all possible emotion classes, where $|l_i|$ is the number of samples in class l_i, weights are assigned to each of the classes as $CW(l_i) = \frac{|L|}{|l_i|}$.

wLB	1LB				2LB				3LB				4LB			
Method	NoB	SW	CW	OVS	NoB	SW	CW	OVS	NoB	SW	CW	OVS	NoB	SW	CW	OVS
Neutral	.92	.92	.90	.90	.91	.92	.89	.89	.91	.91	.85	.85	.92	.91	.84	.85
Anger	0	0	.14	.14	.56	.56	.43	.43	.56	.54	.41	.28	0	.49	.37	.36
Disgust	0	0	.03	.03	0	.48	.35	.35	.05	.43	.31	.35	0	.41	.29	.29
Fear	0	0	0	0	0	.29	.22	.26	0	.29	.24	.25	0	.34	.28	.21
Happiness	.54	.54	.54	.54	.49	.49	.57	.57	.55	.55	.56	.56	.54	.45	.54	.54
Sadness	0	0	.07	.07	0	.28	.22	.23	0	.38	.29	.29	0	.32	.2	.16
Surprise	0	0	0	0	0	0	.13	.14	0	0	.07	.08	0	.01	.06	.07
macro avg	.21	.21	**.24**	**.24**	.28	**.43**	.40	.41	.30	**.44**	.39	.38	.21	**.42**	.37	.35

Table 4.3: **Handling imbalanced data in BiLSTM-nEMC:** The Macro-average F1 score on *emotion(E)* sequences in the DAILYDIALOG dataset with wLB with wLB= $\{1, 2, 3, 4\}$, where NoB= No-Balancing, SW = Smooth-Weights, CW = Count-Weights and OVS = OVer-Sampling. Best results are in bold.

3. **Smooth weights (SW)**: The class weights (CW) described above can be further smoothed by defining $score(l_i) = log(\mu \frac{|L|}{|l_i|})$ ($\mu = 0.15$ for our experiments), and $SW(l_i) = \max(score(l_i), 1)$. Logarithmic smoothing helps reduce the influence of large values and balance the weight distribution.

The results of applying the various strategies including no balancing (NB) are presented in Table 4.3 for the DAILYDIALOG dataset. Identical experiments were conducted on each of the data sets to determine the best possible strategy. One can observe that balancing the dataset yields improvement over leaving it as is, especially as the length of the temporal window wLB increases. CW and OVS provide superior performance with 1LB but SW demonstrated the best consistent performance, by outperforming for wLB where $w \in \{2, 3, 4\}$. Based on this performance and to be consistent, smooth weights (SW) are adopted for training and metric reporting in the remainder of the experiments reported in this Chapter.

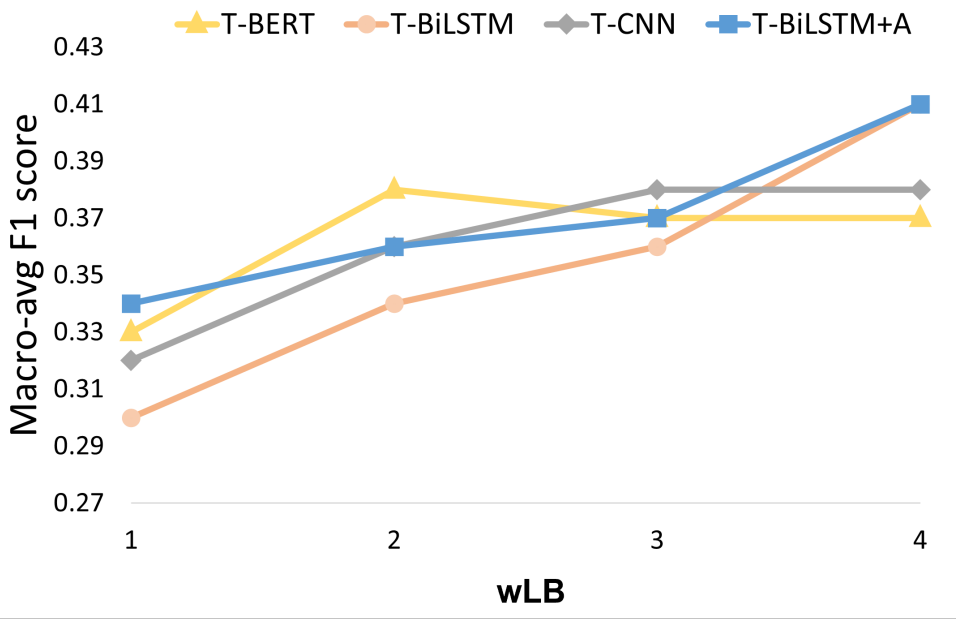


Figure 4.4: Comparing different sequence classifiers with $\boldsymbol{w}$LB and text sequences for DailyDialog. With the exception of T-BERT, all classifiers use Glove embedding.

4.2.9 Comparing classifiers

When comparing between different classifiers using the *text (T)* sequence, BiLSTM with attention, denoted as BiLSTM+A in our sequential neural net model, was substituted by one of BiLSTM, CNN and BERT classifiers to generate the models T-BiLSTM, T-CNN and T-BERT. In general, the results of all the classifiers fall within a narrow range albeit with varying trends as seen in Figure 4.4. Exploring such trends further may be possible with datasets suitable for longer look backs. Unsurprisingly, BiLSTM+A (BiLSTM with attention) is consistently better than BiLSTM. However BiLSTM with bert embedding performed better for a look back of 2. Another interesting observation is the similar behavior given other classifiers such as FGCN, as shown in Figure 4.5.

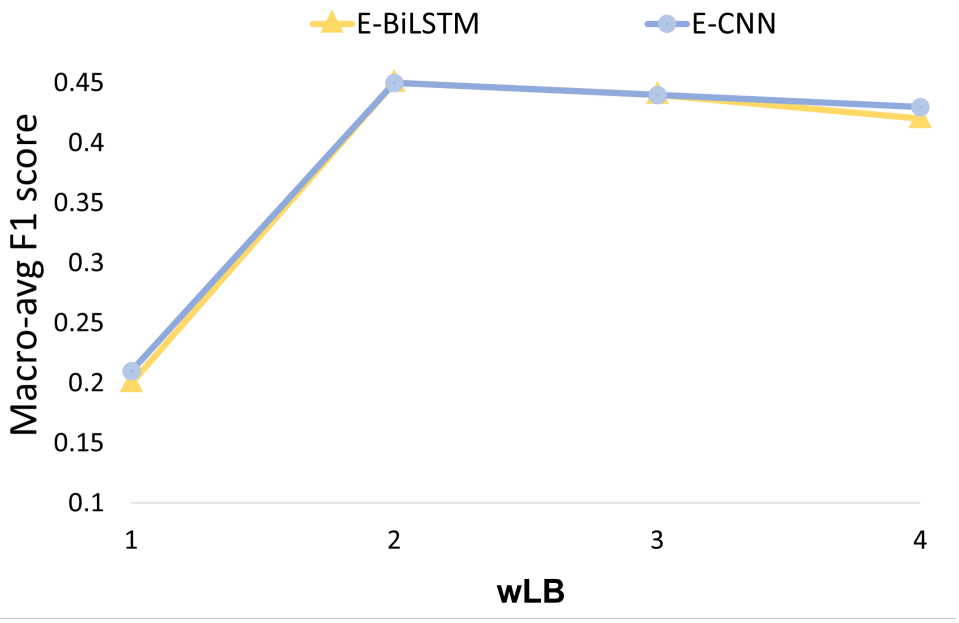


Figure 4.5: Comparing classifiers $\boldsymbol{w}$LB in the Emotion model using in DAILYDIALOG.

4.2.10 Comparing word embeddings

To evaluate the choice of the embedding layer of the *text* (T) sequence classifiers, experiments were conducted with four types of pre-trained word embeddings; word2vec [91], GloVe [98], EWE [1], and BERT (base and uncased) [35]. The results of the choice of embedding as tested on DAILYDIALOG are shown in Figure 4.6. No consistently best word embedding was found, and therefore, the GloVe representations was selected for all further experiments as it had the best overall performance given sufficient context.

4.2.11 Sequence models analysis

The performance of the wLB sequence models was evaluated without incorporating user dependency, instantiated for the three different types of sequences; *emotion sequence (E)*, *text sequence (T)*, and *(emotion, text (ET))* sequence, and for varying values of the size of the look back (wLB) window length $w = \{1, 2, 3, 4, 5\}$ for BiLSTM-nEMC and window length

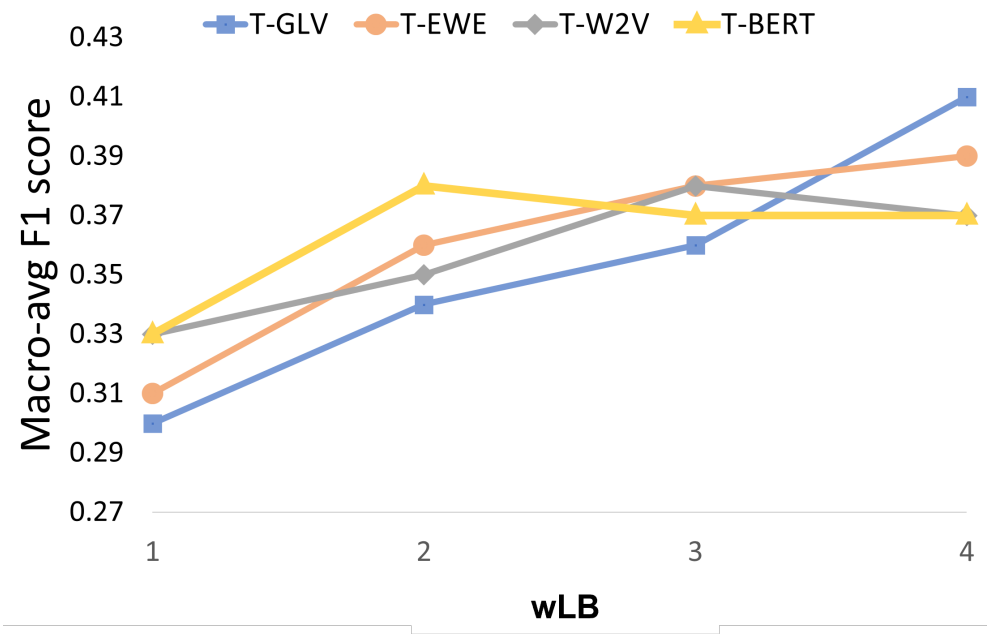


Figure 4.6: Comparing $\boldsymbol{w}$LB using BiLSTM-nEMC with pre-trained text embeddings GloVe, word2vec, EWE and BERT for theDailyDialog dataset.

$w = \{2, 3, 4, 5\}$ for DGCN-nEMC, since DGCN-nEMC requires at least two utterances in the conversation sequence to create its graph. The results of this analysis are shown in Table 4.4 for dyadic and group conversations datasets.

Looking at the overall trend across both dyadic and group conversations datasets, one can observe that DGCN-nEMC *text (T)* only and *(emotion, text) (ET)* wLB sequence models outperform the BiLSTM-nEMC models, suggesting that graph-based models which inherently incorporate user information yield better results than sequential models that do not incorporate such information.

When considering dyadic conversations only, we observe that for each dataset (DailyDialog and Iemocap), the best performance is obtained by ET-DGCN-nEMC with 5LB. However, it seems that the best result of **0.46** obtained by ET-DGCN-nEMC with five look backs is comparable to that of **0.45** obtained by E-BiLSTM-nEMC with just

Group	Dataset	DAILYDIALOG						IEMOCAP					
Two-Speaker	Model	BiLSTM-nEMC			DGCN-nEMC			BiLSTM-nEMC			DGCN-nEMC		
	Type	E	T	ET	E	T	ET	E	T	ET	E	T	ET
	1LB	.20	.34	.34				.28	.24	.25			
	2LB	.45	.36	.41	.44	.37	.41	.38	.33	.35	.38	.23	.41
	3LB	.44	.37	.42	.43	.39	.43	.36	.35	.36	.31	.31	.44
	4LB	.42	.41	.42	.41	.44	.46	.35	.37	.35	.28	.41	.45
	5LB	.40	.42	.43	.39	.45	.46	.34	.38	.36	.27	.44	.45
Group	Dataset	MELD						FRIENDS					
Multi-Speaker	Model	BiLSTM-nEMC			DGCN-nEMC			BiLSTM-nEMC			DGCN-nEMC		
	Type	E	T	ET	E	T	ET	E	T	ET	E	T	ET
	1LB	.28	.17	.19				.27	.14	.18			
	2LB	.30	.21	.21	.30	.29	.31	.27	.19	.19	.28	.27	.31
	3LB	.28	.23	.22	.34	.30	.36	.25	.21	.20	.28	.29	.34
	4LB	.27	.24	.23	.29	.31	.37	.24	.22	.21	.26	.30	.35
	5LB	.26	.25	.24	.28	.34	.39	.23	.24	.22	.25	.33	.40

Table 4.4: Macro-average F1 scores of the $\boldsymbol{w}$LB sequence model, instantiated as *emotion(E)*, *text(T)* and *(emotion, text)(ET)* sequence model types. Here DGCN-nEMC outperforms BiLSTM-nEMC as a classifier on the dyadic and group conversation datasets in the *text* and *(emotion, text)* sequence sequence model types. A green highlight denotes the best performance of a model. All *emotion(E)* sequence model type trend positively from 1LB to 2LB with green highlighted cells with lower wLB suggesting *recency* in the emotion data. The text models *text(T)* trend positively with more textual context but are not able to capture the local emotion as early as the *emotion(E)* models are. For all BiLSTM models there appears to be a large positive trend from 1LB to 2LB as 2LB provides the most *recent same speaker information.*

2 look backs ($w = 2$). From an efficiency point of view, for many real-time applications it is desirable to predict the next emotion using the lowest number of looks backs. In such applications using E-BiLSTM-nEMC with 2 look backs ($w = 2$) would be more efficient than

Dataset	DAILYDIALOG			MELD		
	E	T	ET	E	T	ET
1SLB	**.47**	.42	**.43**	.28	.30	.38
2SLB	.45	**.43**	.38	.28	**.36**	.41
3SLB	.43	.42	.36	**.29**	.33	**.43**
1OLB	.43	.37	.40	.23	.24	.31
2OLB	.40	.30	.37	.25	.26	.34
3OLB	.35	.29	.35	.22	.30	.36

Table 4.5: Comparing the $\boldsymbol{w}$SLB and $\boldsymbol{w}$OLB sequence models in dyadic conversations, using DAILYDIALOG and group conversations using DGCN-nEMC on MELD. SLB denotes self-dependency look backs and OLB denotes other dependency look backs. Best performing models are in bold.

using ET-DGCN-nEMC with five look backs ($w = 5$).

Performance with look backs is dependent on the type of input. For emotion data, less look backs provide better performance but for text only more look backs (therefore more context) provides better performance. Text only models are not able to outperform models that have access to the true emotion label values. But in many real-world applications human emotion annotation is not necessarily available and at best the model would be provided with estimated emotion information. This is a limitation in mirroring the effect the of type of data in the study on the actual real-world performance of such models. This study (data analysis) is more concerned in deriving insights for conversation structuring than producing actual reported results. the focus is the comparative analysis rather than the actual value of model performance.

In Table 4.4 the best performance of a model in all wLBs is highlighted as green. Notice that in the case of E-BiLSTM-nEMC across both the dyadic datasets, the prediction based on more than 2 look backs ($w = \{3, 4, 5\}$) performs worse, suggesting that *recency* plays an important role in predicting conversation emotions. Since these experiments are performed

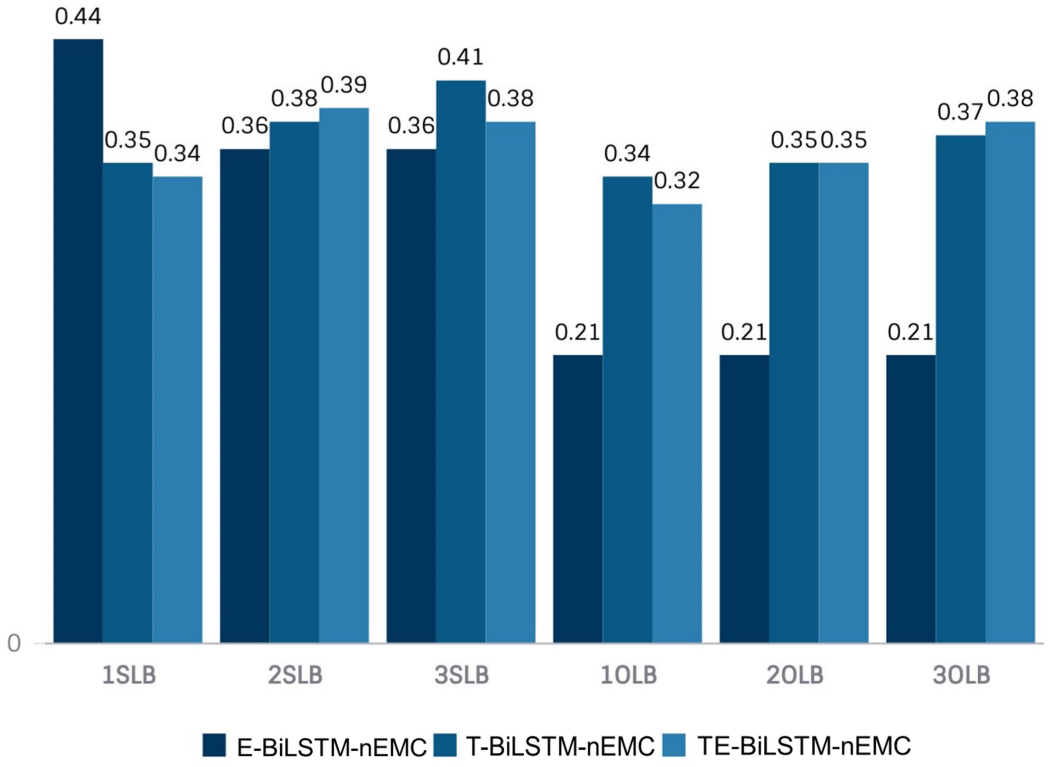


Figure 4.7: Comparing the $\boldsymbol{w}$SLB and $\boldsymbol{w}$OLB sequence models in DAILYDIALOG using BiLSTM-nEMC. Vertical axis represents the F1-score for the different SLB and OLB models.

on dyadic conversations, speakers $\mathcal{A}$ and $\mathcal{B}$ take turns. Taking part in a conversation would appear as $\mathcal{A} \rightarrow \mathcal{B} \rightarrow \mathcal{A} \rightarrow \ldots$. The prediction performance at $(w = 2)$ indicates that the emotion expressed in the current turn and the emotion expressed by the same user in the previous turn is an important factor for predicting the emotion expressed in the same user's next turn. This also indicates the importance of *self-dependency*.

One observes a similar trend for the emotion data across the two group conversation datasets where generelly, the smaller the number of look backs, the better the performance. Suggesting that *recency* plays an important role in group conversations as well.

4.2.12 Incorporating self-dependency

The wSLB and wOLB sequence models, incorporate *self-dependency* and *dependency on others*, respectively. Temporal look back windows of length $w = \{1, 2, 3\}$ for BiLSTM-nEMC and $w = \{2, 3, 4\}$ for DGCN-nEMC were evlauated. Results are presented in Table 4.5 for both dyadic and group datasets using DGCN-nEMC. The self-dependency models (wSLB) outperform all the other-dependency models (wOLB), for the same look back w for *emotion(E)*, *text(T)* and *(emotion, text)(ET)* sequence model types and in both types of datasets. Consider the results presented in Figure 4.7 for the dyadic dataset using BiLSTM-nEMC. Similar to DGCN-nEMC, (i) all self-dependency models (wSLB) outperform all other-dependency models (wOLB), for the same look back w, and (ii) the best overall performance is achieved by 1SLB using the *emotion (T)* sequence, thus providing further support to the importance of *recency* and *self-dependency* when predicting emotions in conversations.

Similar observations can be seen for the group dataset MELD using BiLSTM-nEMC on emotion data as shown in Figure 4.8. It is observed again that (i) the self-dependency models for each user in MELD (wSLB) consistently outperform all other-dependency models (wOLB), for the same temporal window w, and (ii) the best overall performance for every user in MELD is achieved by 1SLB using the *emotion sequence*, thus providing further evidence of the importance of *recency* and *self-dependency* when predicting emotions in conversations.

4.2.13 Speaker emotion profiles

To examine speaker dependency in group conversation datasets, data related to each user is extracted. For each character in the MELD dataset the same set of experiments on *recency* and *self-dependency* was conducted. Figure 4.9 shows the results for each of the six characters in the MELD group conversation dataset to explore the performance of the model by each of the emotion categories (*neutral*, *sad*, *surprise*, *anger*, *disgust*, *fear*, *happy*). Figure 4.9

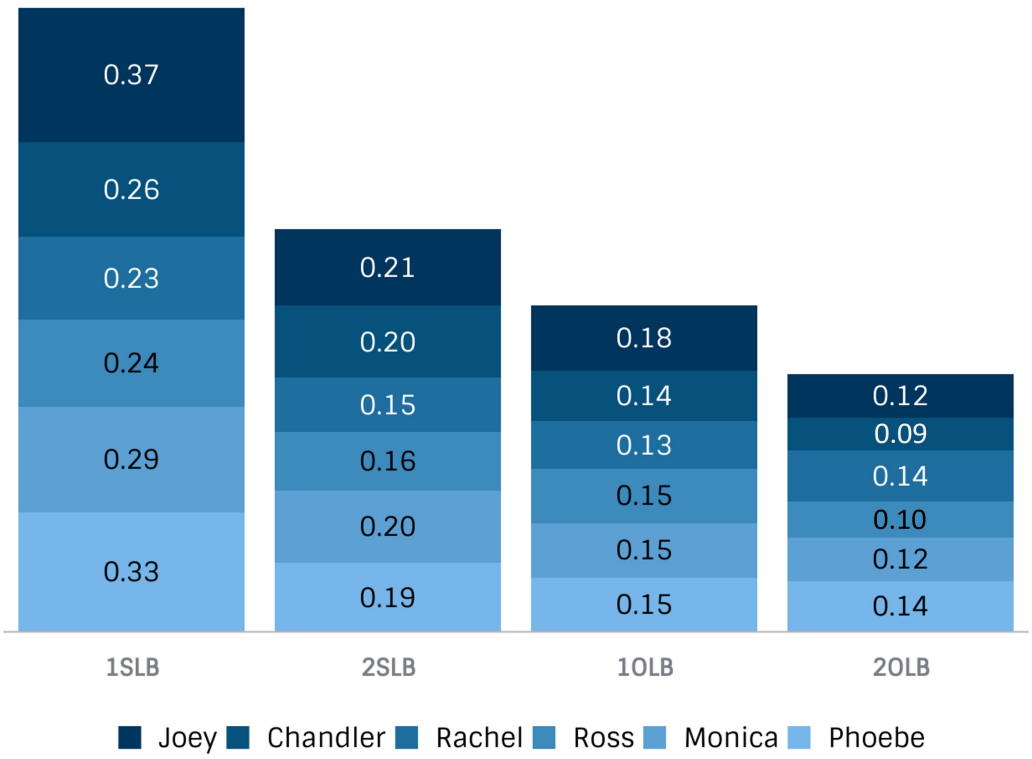
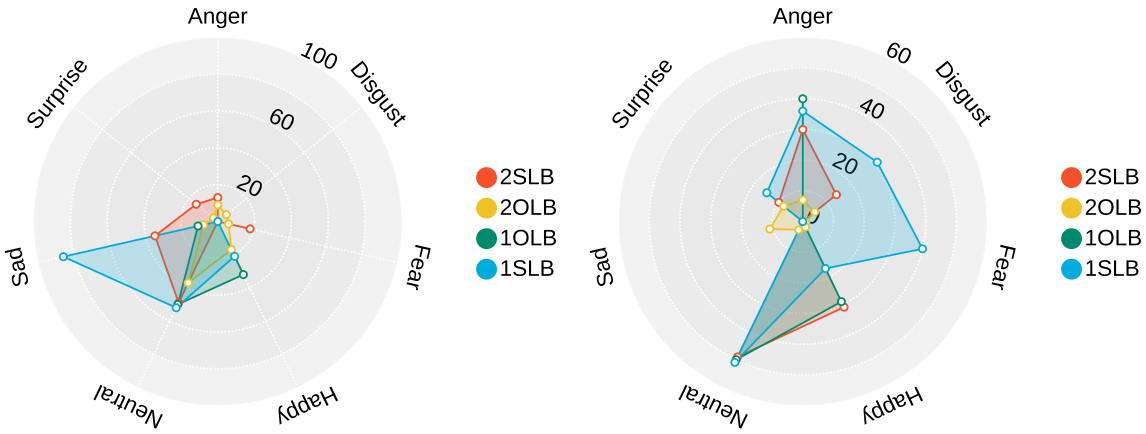


Figure 4.8: Macro-avg F1 score using E-BiLSTM-nEMC for the MELD group conversation dataset for different SLB and OLB. Each color represents one of the six characters in the dataset. Results are plotted by character

shows the radar plots of three of the main characters (*Chandler, Joey, Pheobe, Rachel, Ross and Monica*) and their emotion signals. One can observe that for all characters, the (*w*SLB) models (1SLB, 2SLB) outperform the (*w*OLB) models (1OLB, 2OLB) as characterized by the larger area occupied by the self-dependent and more recent models. Other than the common patterens found on *recency* and *self-dependency*, Figure 4.9 shows each emotion predictability profile of each user. The profiles shows that users are different in the level of productivity of the different emotions. This may indicate that models may need more sophistication to capture these unique individual characteristics. One example is to include common sense knowledge which has shown to be able to capture complex emotional context [41, 146].



(a) Chandler (b) Joey

(c) Phoebe (d) Ross

(e) Monica (f) Rachel

Figure 4.9: Emotion profiles the friends characters coming from the MELD dataset.

Dataset	$pw = 3$	$pw = 4$	$pw = 5$
DAILYDIALOG	**.46**	.39	.39
IEMOCAP	**.45**	.38	.40
MELD	**.39**	.34	.36
FRIENDS	**.40**	.34	.35

Table 4.6: Sensitivity analysis of the past window size parameter in the graph construction pw. The results of varying the DGCN-nEMC graph's past window pw on the best result for each dataset in Table 4.4, that is ET-DGCN-nEMC and wLB sequence model with 5 look backs ($w = 5$). A past window $pw = 3$ that captures the most recent past three turns in the graph provides the best results. $pw = 1$ and $pw = 2$ are not reported as they do not provide meaningful graph formations and give very poor results.

4.2.14 Varying graph past look back lengths, pw

In Table 4.4, we presented results where the overall best performance for each dataset was obtained using ET-DGCN-nEMC with five look backs ($wLB = 5$). For these results the graph past window pw variable used for creating the edges in DGCN-nEMC was empirically set to $pw = 3$ capturing only recent data around any text in the graph. Table 4.6 presents a detailed analysis varying the values of pw in the graph construction from $pw = \{3, 4, 5\}$. Note that increasing pw, i.e., increasing the connected history data in the graph construction, gives worse results, confirming the importance of *recency*. In this work, results on varying a future window showed no benefit to the problem. These results are not reported here.

4.2.15 Varying speaker edges

The role of *self-dependency* was explored further by creating a variant of DGCN-nEMC, denoted DGCN-nEMC-S, that uses only same speaker edges in the graph (i.e., only uses edges of type $u_i \rightarrow u_i$ for i in the range of speakers). Table 4.7 summarizes the results and shows

	DailyDialog						MELD					
	DGCN-nEMC			DGCN-nEMC-S			DGCN-nEMC			DGCN-nEMC-S		
	E	T	ET	E	T	ET	E	T	ET	E	T	ET
2LB	.44	.37	.41	**.46**	.38	.41	.30	.29	.31	.28	.30	**.33**
3LB	**.43**	.39	.40	**.43**	.39	.42	.34	.30	**.36**	.31	.32	**.36**
4LB	.41	.44	**.46**	.40	.44	**.46**	.29	.31	**.37**	.26	.32	**.37**
5LB	.39	.45	.46	.37	.46	**.47**	.28	.34	**.39**	.28	.33	.38

Table 4.7: Comparing DGCN-nEMC to DGCN-nEMC-S in dyadic conversations, using DailyDialog and in group conversations, using MELD. The bold values represent the best performance in a conversation group (with and without speaker) in a given look back. For Both groups of conversations the sparsification (using only same speaker edges) results in similar or comparable results.

that on average for the *text (T)* only and the combined *emotion, text (ET)* sequences, results are either higher for DGCN-nEMC-S or the same for both DGCN-nEMC and DGCN-nEMC-S, suggesting that it is both more efficient and accurate to use same speaker edges only. This further confirms the importance of *self-dependency* in predicting future evoked emotion.

4.2.16 Summary

This section analyzed the emotional data available for the problem of predicting next emotions in conversations (*nEMC*). Two neural network models were developed to address the next emotion predication problem – BiLSTM-nEMC and DGCN-nEMC. These two models are explainable in terms of recency and self-dependnecy. To analyze the data three modeling dimensions relevant to this task were proposed and an extensive empirical analysis was conducted to determine the effect of *recency* and *self-dependency* on a model's prediction accuracy. The results indicate that (*i*) for (*text*) and (*emotion,text*) turns, DGCN-nEMC, the *graph network model*, that inherently accounts for user information and recency outperforms

BiLSTM-nEMC, the *sequence model*; (*ii*) the use of same speaker data (and/or same speaker edges) further improves the results of DGCN-nEMC, confirming the role of *self-dependency* in emotion prediction; and (*iii*) for *emotion* sequences, the BiLSTM-nEMC model that uses only same user data and as few as two look backs, performs similarly to the more complicated DGCN-nEMC model using at least four look backs. A simpler model that can predict emotions with fewer look backs may be more efficient for certain applications provided it is very effective in estimating the emotion of each turn. Text only models are not able to outperform models that have access to the true emotion label values. In real-world applications human emotion annotation is not necessarily available and at best the model is provided with estimated emotion information. This is a limitation in mirroring the effect of the type of data in the study and the actual real-world performance of such models. The importance of this data analysis is not in producing a estimated value of efficiency but rather deriving considerations for designing conversation forecasting models. In conclusion, when designing conversation emotion prediction models, taking into consideration the dimensions of *recency* and *self-dependency* is beneficial.

4.3 The FGCN forecasting model

This section describes how the results of the emotion data analysis provided earlier are incorporated into the forecasting model FGCN-nEMC, shown in Figure 4.10, for the problem of next emotion prediction. This section also describes a comparative study of the FGCN-nEMC forecasting model with baselines in the literature.

4.3.1 Incorporating data analysis conclusions in FGCN

Since the best results are obtained by the graph model with both textual and emotional data combined, and the emotion data analysis shows that taking into consideration the dimensions

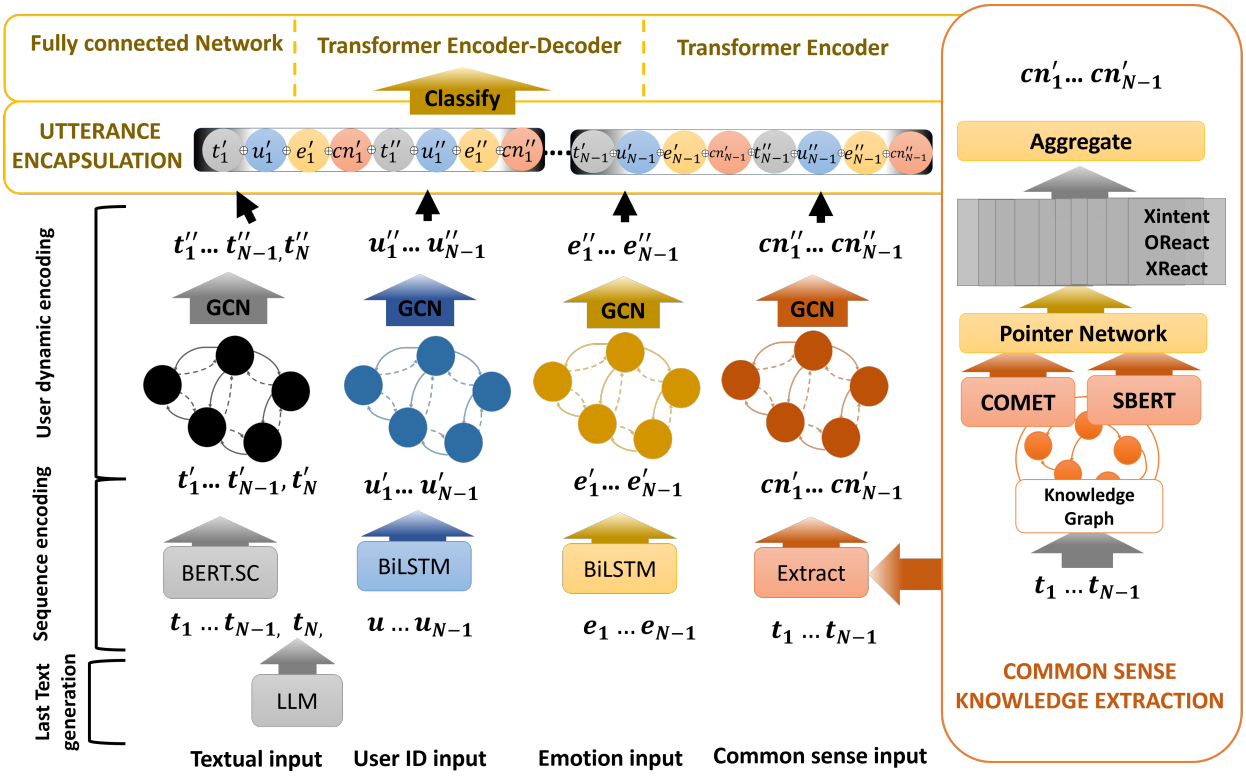


Figure 4.10: The FGCN-nEMC model architecture, A varinat of FGCN for next emotion prediction in conversations. A sample encoding of a conversation of length 6 with three classes of optional data (E and CN) is shown. For each type of data the model separately (shown in the data layers) uses sequential encoding to populate the vertices in a graph where it undergoes GCN transformation. The results of the sequential encoder and user dynamics encoder are concatenated for classification. The details of graph construction in the user dynamics encoding are shown in Figure 4.11.

of *recency* and *self-dependency* can be particularly beneficial, these results are used to inform the construction of the FGCN-nEMC graph model for the next emotion prediction problem.

Graph construction

Results presented in Section 4.2.14 indicated that $pw = 3$ enhances performance by accounting for recency. Each vertex v_i is thus linked with the three immediate past turns: $v_{i-1}, v_{i-2}, v_{i-3}$. That is a sparsification of the fully connected graph of the forecasting model described in Chapter 3. Given the importance of self dependency shown in Section 4.2.12, we also connect each vertex with other vertices that belong to the same user to include self-dependency in the graph structure. The edge weights are set using a similarity based attention module. The attention function is computed in a way such that, for each vertex, the incoming set of edges has a sum total weight of 1. The weights are calculated as $\alpha_{ij} = softmax(v_i \odot W_e[v_{i-1}, v_{i-2}, v_{i-3} +$ same-user-vertices]), ensuring the vertex v_i receives a total weight contribution of 1. As the graph is directed, two vertices can have edges in both directions with different relations. The relation r of an edge r_{ij} is set based on the user dependency between user of turn i and user of turn j. Figure 4.11 illustrates an example graph construction of a five turn conversation between two users applying graph sparsification of the fully connected graph.

4.3.2 Datasets

The datasets used here are used to compare results with available baselines in the literature. More complete details on these datasets can be found in Appendix A.

IEMOCAP

The IEMOCAP dataset [19] was collected by the SAIL lab at USC. It consists of approximately 12 hours of multimodal conversation data. Only the text modality is used in this work. The

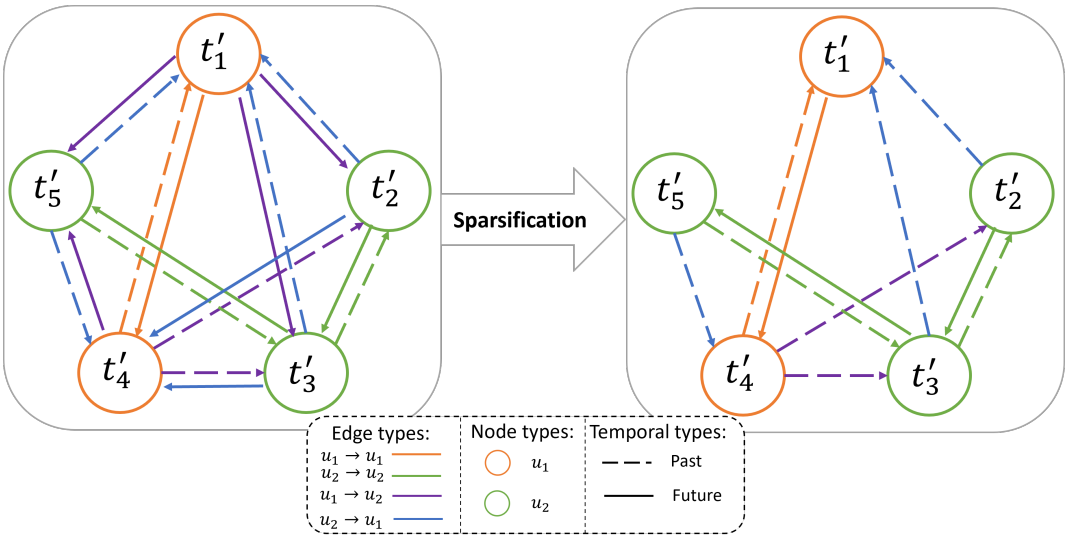


Figure 4.11: An example FGCN-nEMC graph construction of a five turn conversation between two users using graph sparsification of the fully connected graph described in Chapter 3. Each turn has an edge with the immediate three turn of the past. The edge types set the speaker dependency between speakers of the turns. A edge between the same user is a self-dependency edge. All self dependency edges are maintained.

dataset contains 152 dialogues with a total of 7,433 utterances, and is labeled with six emotion categories: happy, sad, neutral, anger, excited and frustrated.

MELD

The MELD dataset [103] contains the conversations from the Friends TV show transcripts, which is a multimodal extension of the EmotionLines dataset [55]. In this work, we only use the text modality. The dataset contains 1,433 dialogues with a total of 13,708 utterances. The dataset is labeled with the user and seven emotion categories: neutral, surprise, fear, sadness, joy, disgust and anger.

EmoryNLP

The EmoryNLP dataset [139] also contains the conversations from Friends TV show transcripts. The dataset contains 897 dialogues with a total of 12,606 utterances. It is labeled with the user and seven emotion categories: sad, mad, scared, powerful, peaceful, joyful and neutral.

4.3.3 Baselines

The FGCN-nEMC model and its variants are evaluated against state-of-the-art methods nEMC. The baseline results shown here are reported directly from [77]. The implementation of the results in this work is compared to these baselines following the same methodology used in [77]. Details of the baselines are:

- **CNN** [70] The models are trained at the utterance level to infer the emotion class of the next turn.
- **RoBERTa Large** [150] The models are trained at the utterance level to infer the emotion class of next turn.
- **sc-LSTM** [101] The model is a simple contextual unidirectional LSTM model to infer the emotion class of next turn.
- **DialogueRNN** [86] the model is an RNN-based model, which uses three separate GRU networks to track individual speaker states.
- **DialogueGCN**[42] The model uses a relational GCN to model the relation between turn text.
- **COSMIC** [41] Is a model that incorporates different elements of commonsense knowledge and tracks speaker individual speaker states.

- **DialogInfer** [77] This model and it variants are the current state-of-the-art model in emotion recognition in conversations. The DialogInfer-(S+G) is a variant with sequential and graph encoding. DialogInfer-(S) is a variant with a variant with sequence encoding only. DialogInfer-ensamble is an ensembale modelof a graph and sequence model.

4.3.4 nEMC results

Results of the static and dynamic training of the model are reported and compared with baselines.

Next emotion predication

The results in Table 4.8 show that across all datasets, approaches that combine graph and sequence models such as FGCN-nEMC, DialogInfer-(S+G) and DialogInfer-Ensemble outperform sequence-only models such as CNN, sc-LSTM, RoBERTa and DialogueRNN. That is, combined graph and sequence models outperform sequence-only using both Glove and RoBERTa embeddings. This result is more evident when we compare the variants of the base model DialogInfer. The results show that both DialogInfer-(S+G) and DialogInfer-Ensemble, that combine both sequence and graph models, outperform DialogInfer-S a sequence only model. Similar results can be seen when comparing RoBERTa to FGCN-nEMC-T. FGCN-nEMC-T outperforms its sequence base model RoBERTa by combining a graph model with the sequence model. This illustrates the effectiveness of combining a graph model with sequence model classifiers. When comparing the best preforming graph infused models DialogInfer and FGCN-nEMCin Table 4.8, results show that the DialogInfer graph infused model outperforms FGCN-nEMC's with less data, that is using only text data. A closer look to the graph model introduced in DialogInfer shows that the model uses an attention

	Model	IEMOCAP	MELD	EmoryNLP
Glove	CNN	44.09	36.31	20.97
	sc-LSTM	56.22	36.06	19.75
	DialogueRNN	58.12	36.93	20.37
	DialogueGCN	56.48	36.98	19.59
	DialogInfer-S	60.45	38.09	<u>21.08</u>
	DialogInfer-G	59.48	36.62	20.22
	DialogInfer-(S+G)	<u>60.74</u>	<u>38.46</u>	**21.69**
	DialogInfer-Ensemble	**65.31**	**38.48**	20.95
RoBERTa	RoBERTa Large	43.24	36.99	20.46
	sc-LSTM	58.81	37.71	22.26
	DialogueRNN	59.53	38.70	21.98
	COSMIC	61.50	39.49	21.60
	DialogInfer-S	63.63	40.32	23.09
	DialogInfer-G	59.94	38.06	22.81
	DialogInfer-(S+G)	64.70	**40.67**	22.63
	DialogInfer-Ensemble	**66.39**	39.41	**24.09**
	FGCN-nEMC-T	60.69	38.98	22.09
	FGCN-nEMC-TU	60.72	38.99	22.21
	FGCN-nEMC-TE	64.45	39.89	23.09
	FGCN-nEMC-TCN	63.24	39.35	22.98
	FGCN-nEMC-TEU	64.53	39.97	23.18
	FGCN-nEMC-TCNU	63.36	39.54	22.99
	FGCN-nEMC-TCNE	65.56	40.21	23.53
	FGCN-nEMC-TCNEU	<u>65.62</u>	<u>40.34</u>	<u>23.61</u>

Table 4.8: Experimental results for next emotion predication nEMC with static training for each embedding. Best F1-macro scores for each dataset are in bold. The second best are underlined. The table also shows the results of two types of embedding (Glove and RoBERTa). Roberta embedding results in a better performing model than GLOVE. DialogInfer-Ensemble and DialgoInfer-(S+G) are the best performing algorithms across both embedding types.

mechanism to infer the participant's next emotion and model the emotion shifts relating to persistence and contagion. The FGCN-nEMC model focus helps to eliminate noise in its user relational graph through graph sparsification that models recency and self-dependency.

Model	IEMOCAP	MELD	EmoryNLP
FGCN-nEMC-T	58.53	37.24	21.67
FGCN-nEMC-TU	58.52	37.33	21.01
FGCN-nEMC-TE	62.23	38.79	22.18
FGCN-nEMC-TCN	61.35	38.22	21.78
FGCN-nEMC-TEU	62.41	38.81	22.03
FGCN-nEMC-TCNU	61.43	38.46	21.79
FGCN-nEMC-TCNE	<u>63.31</u>	<u>39.17</u>	<u>22.37</u>
FGCN-nEMC-TCNEU	**63.61**	**39.25**	**22.59**

Table 4.9: Experimental results for next emotion predication nEMC with dynamic training and RoBERTa embedding. Best F1-macro scores for each dataset are in bold. Second best are underlined

The results indicate that the attention mechanism used in DialogInfer has a higher effect on performance. One possible direction to take is to combine both graph optimizations into the graph portion of the data extensible FGCN-nEMC model.

Table 4.9 shows the impact of dynamically training on the FGCN-nEMC model. When comparing the statically trained FGCN-nEMC results in Table 4.8 and the dynamically trained FGCN-nEMC results in Table 4.9 the statically trained models outperform their corresponding dynamically trained models. Static training contains all context for every sample. It is a natural result for the F1-macro score to improve when the entire conversation context is provided in the training process.

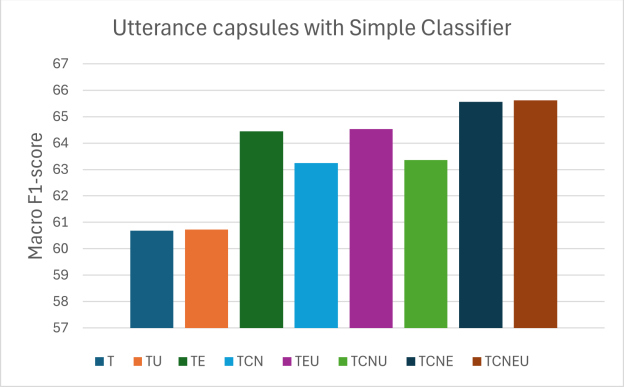
The results shown in Tables 4.8 and 4.9 also show that incorporating more input types leads to better performance in the FGCN-nEMC model. When comparing FGCN-nEMC-TU, FGCN-nEMC-TE and FGCN-nEMC-TCN, for example, FGCN-nEMC-TE outperforms these variants indicating that including the emotion data in the forecasting graph neural network model provides the highest impact, improving the models' predicting macro F1-score. This is a natural result as the emotion data contains data given by human annotators. The emotion

data explicitly contains the emotion shifts that relate to relating to persistence and contagion. As suggested by the results in Table 4.8, modeling the emotion data in the FGCN-nEMC graph enables the model to capture this emotion information but not as effectively as the attention mechanism used in DialogInfer. FGCN-nEMC-TCN comes second best in the FGCN-nEMC performing models. This model contains common sense knowldge, which suggests the effectiveness of incorporating common sense knowledge for the problem of next emotion prediction in conversations.

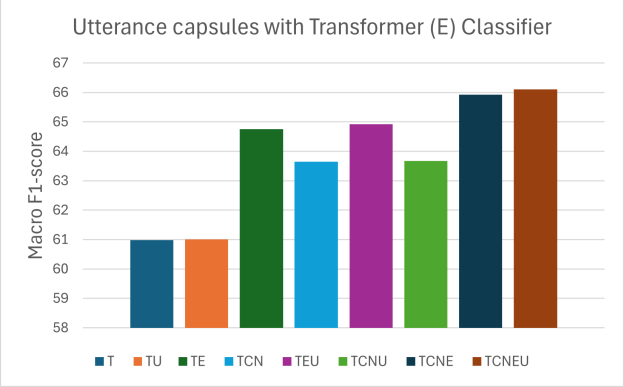
Utterance capsule information and classifier type sensitivity

To understand the impact of the different types of data included in the information capsules passed to the classifier and the type of classifier used in the FGCN-nEMC model, the results of such on the IEMOCAP dataset is shown in Table 4.10. This table displays more clearly that more data always outperforms less data and that individually, emotion data followed by common sense knowledge are the most influential data sources. The emotion data directly contains the emotion shifts that relate to relating to persistence and contagiousness and common sense knowledge are features that aenable the model to encode complex characteristics, dynamics and probation in conversation. The extensible structure of FGCN-nEMC enables the creation of an emotion and a knowledge aware forecasting model that is able to capture emotion shifts, context propagation, user dynamics and mood shifts during the conversation and improve performance.

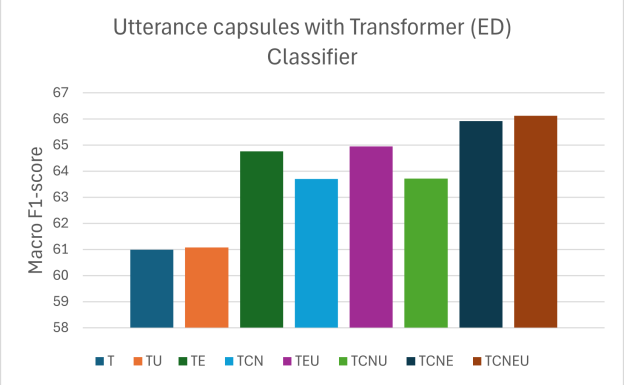
The FGCN-nEMC results in Table 4.10 also indicate that using a Transformer classifier (E, ED) obtains better results than the simple fully connected network classifier. However, the encoder-decoder Transformer classifier (ED) performance is similar to the encoder only Transformer classifier (E). This suggests that the complexity of ED does not necessarily provide significant benefit for this single label classification task. This is expected since the decoder here is limited by the simplicity of the underlying problem when compared to more



(a) Simple

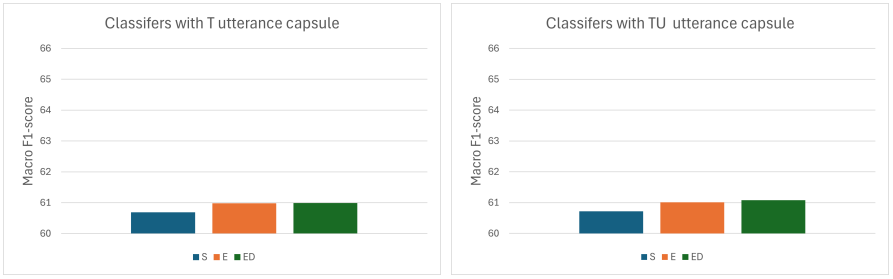


(b) Transformer(E)



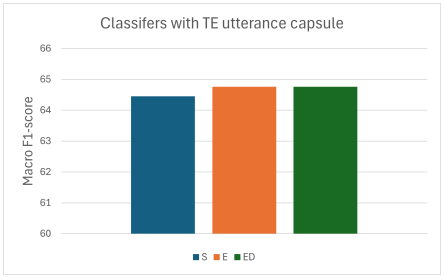
(c) Transformer(E)

Figure 4.12: Impact of data types in the information capsules for next emotion predication nEMC from the IEMOCAP dataset passed to the classifier. More data in the information capsule enables enhanced performance. Information capsules with (Emotion) E data, outperforms information capsules without E using the same classifier.

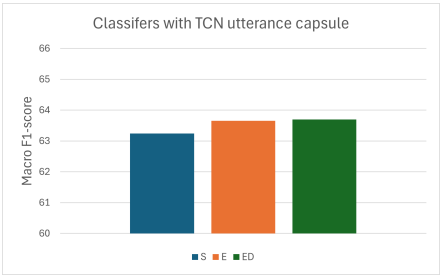


(a) T capsule

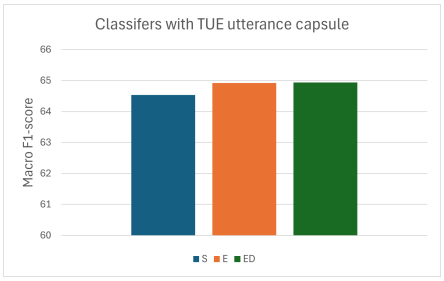
(b) TU capsule



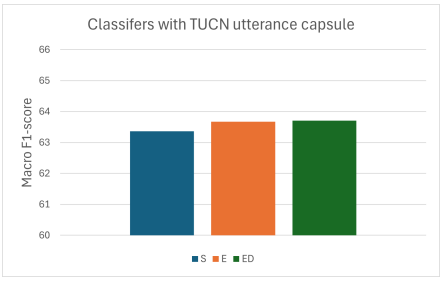
(c) TE capsule



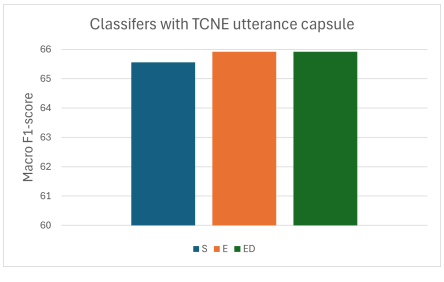
(d) TCN capsule



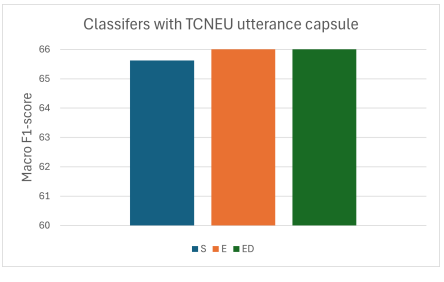
(e) TUE capsule



(f) TUCN capsule



(g) TCNE capsule



(h) TCNEU capsule

Figure 4.13: The classifier performance for each utterance capsule in next emotion predication nEMC is shown for the IEMOCAP dataset. The Transformer classifier (E, ED) is better than the simple fully connected network classifier. However, the encoder-decoder Transformer classifier (ED) performance is similar to the encoder only Transformer classifier (E).

	Classifier		
Variant	**Simple(S)**	**Encoder(E)**	**Encoder-Decoder(ED)**
T	60.69	60.98	60.99
TU	60.72	61.01	61.08
TE	64.45	64.76	64.76
TCN	63.24	63.65	63.70
TEU	64.53	64.92	64.94
TCNU	63.36	63.67	63.71
TCNE	65.56	65.92	65.92
TCNEU	65.62	66.10	66.12

Table 4.10: Varying the utterance information capsule and classifier in next emotion predication nEMC for IEMOCAP using static training in FGCN-nEMC. S (Simple), E(Encoder) and ED(Encoder-Decoder) are the types of classifiers used to obtain the reported F1-macro score.

complex multi output decoding tasks.

4.3.5 Varying the text embedding

The text embedding that initializes the graph neural network is varied in order to better understand the impact of using SLM versus LLM embeddings. LLM embeddings are more generalized compared to SLM embeddings. Varying the text embedding enables the examination of the influence of using a more generalized embedding on the specific task of next emotion prediction.

- **GPT embedding extraction**. For non-fine-tuned embedding extraction the OpenAI GPT-2 is used. The fine tuned embedding extraction uses DialogGPT [144]. DialogGPT repository is based on huggingface pytorch-transformer and OpenAI GPT-2, and contains data extraction script, model training code and pretrained small (117M) medium (345M) and large (762M) model checkpoint. The raw small model is fine

tuned to reduce the chances of over fitting due to the size of our dataset. The model is fine tuned on the two datasets for 2 epoches with a max sequence length of 200. The AdamW optimizer, a learning rate of 1e-5, and a batch size of 4 is used for training and validation. Figure 4.14 presents the format of a training instance for the fine tuned model.

```
"Utterance1.... [EOS] Utterance 2 [EOS].... [PAD]"
```

Figure 4.14: GPT2 format of a training instance for the fine tuned model.

- **BERT embedding extraction**, For non-fine-tuned embedding extraction 'bertbase-uncased' is used. 'bertbase-uncased' is fine-tuned using the DialogBERT approach [46]. Since the base-size model is vulnerable to over-fitting when using small datasets, The response generation model is trained with small-sized models. The small-size model reduces the 'bertbase-uncased' configuration to 6 transformer layers, has a hidden size of 256, and contains 2 attention heads (L=6, H=256, A=2). All of the experiments use the default BERT tokenizer. The model is fine tuned on the two datasets for 2 epochs with a max sequence length of 200. The AdamW optimizer, a learning rate of 5e-5, and a batch size of 4 is used for training and validation. Figure 4.15 presents the format of a training instance for the fine tuned model.

```
"[CLS] Utterance1.... [SEP] Utterance 2 [SEP].... [PAD]"
```

Figure 4.15: BERT format of a training instance for the fine tuned model

- **Gemini embedding extraction**, For non-fine-tuned embedding extraction the Gemini API with gemini-1.0-pro is used directly [126]. For fine-tuned embedding extraction

the default supervised learning fine tuning setting for Gemini API is used. The model is fine tuned on the datasets for 2 epochs with a learning rate of 1. Figure 4.16 shows the format of a training instance/prompt.

```
"messages": ["role": "system", "content": "Based on the following conversation with this sequence of utterance, classify the emotion of the coming unseen utterance into one of the following emotion classes: [set of emotion labels]. The output should be a single label from the emotion classes", "role": "user", "content": "CONVERSATION: Utterance 1:.... Utterance 2:... LABEL:", "role": "model", "content": "label value"]
```

Figure 4.16: Gemini format of a training instance/prompt

- **Llama embedding extraction**, For non-fine-tuned embedding extraction the 'Llama-2-7b' is used. For fine-tuned embedding extraction the Hugging Face fine-tuning with PEFT QLoRA with 'Llama-2-7b' is used. QLoRA pre-trained models are loaded to GPU as quantized 4-bit weights. QLoRA is required since a single GPU is used to fine-tune the model. The model is fine tuned on the two datasets for 2 epochs with a max sequence length of 200. The AdamW optimizer, a learning rate of 2e-5, and a batch size of 4 is used for training and validation. Figure 4.17 presents the format of a training instance/prompt.

The results in Table 4.11 shows the results of varying the text embedding extracted from language models using the best performing variant from FGCN-nEMC. The results show that fine tuning the language models provides better results when compared to using extracted embedding without fine tuning. These results also confirm previous findings [143] that LLMs' have unsatisfactory performance in many emotion recognition tasks without fine-tuning on

"text": <s>[INST] «SYS»"Based on the following conversation with this sequence of utterance, classify the emotion of the coming unseen utterance into one of the following emotion classes: |set of emotion labels|. The output should be a single label from the emotion classes"«/SYS»Utterance 1:.... |/INST] Utterance 2:... </s><s>[INST] label value</s>

Figure 4.17: Llama format of a training instance/prompt

the emotional knowledge. Table 4.11 shows variants of the model using embeddings of two SLM models GPT and Bert, and two LLM models Gemini and Llama.

The results presented in Table 4.11 show that SLM GPT and Bert outperform LLMs for next emotion prediction. It is important here to recognize that LLMs are sensitive to prompt engineering since they are more generalized models. The prompt engineering used to obtain the results in the table are straightforward. More sophisticated prompts that incorporate more context instruction and/or RAG might produce better results.

We can also see from the results that static training provides superior results to dynamic training in both fine tuned and not fine tuned models. This indicates that additional context in the training examples, when available, produces better results.

Model infused with response generation

This section explores the impact of infusing the FGCN-nEMC model with a generated last utterance N into the text layer. I first describe the specifics of the generative models used to create the last utterance and then analyze the results and discuss the impact of last utterance generation and inclusion in the FGCN-nEMC model.

		Model	IEMOCAP	MELD
Not fine tuned	Static	FGCN-nEMC-GPT	62.45	36.98
		FGCN-nEMC-RBRT	**63.33**	**37.98**
		FGCN-nEMC-Gemini	56.78	32.78
		FGCN-nEMC-Llama	56.13	34.79
	Dynamic	FGCN-nEMC-GPT+	60.78	36.34
		FGCN-nEMC-RBRT+	**61.11**	**38.98**
		FGCN-nEMC-Gemini+	54.78	33.34
		FGCN-nEMC-Llama+	55.22	34.12
Fine tuned	Static	FGCN-nEMC-GPT	64.21	38.79
		FGCN-nEMC-RBRT	**65.62**	**40.34**
		FGCN-nEMC-Gemini	59.26	34.62
		FGCN-nEMC-Llama	58.34	35.53
	Dynamic	FGCN-nEMC-GPT+	62.98	38.79
		FGCN-nEMC-RBRT+	**63.61**	**39.25**
		FGCN-nEMC-Gemini+	56.56	35.21
		FGCN-nEMC-Llama+	57.39	35.79

Table 4.11: Experimental results for next emotion predication nEMC using SLM and LLMs embedding with and without fine tuning. This tables shows the F1-score. Best F1- macro scores for each dataset are in bold. Second best are underlined. The FGCN-nEMC used here uses the best performing models shown in Tables 4.8 and 4.9.

.

Response Generation models

- **GPT response generation.** For the GPT response generation results, this work uses DialogGPT [144]. DialogGPT repository is based on huggingface pytorch-transformer and OpenAI GPT-2, containing the data extraction script, model training code and

pretrained small (117M) medium (345M) and large (762M) model checkpoint. The model is fine tuned using the small pretrained model to reduce the chances of over fitting due to the size of our dataset. The model is fine tuned on the two datasets for 2 epoches with a max sequence length of 200. The AdamW optimizer, a learning rate of 1e-5, and a batch size of 4 is used for training and validation. The format of a training instance for response generation is shown in Figure 4.18.

```
"Utterance1.... [EOS] Utterance 2 [EOS].... [PAD]"
```

Figure 4.18: GPT2 format of a training instance for response generation.

- **BERT response generation.** For the BERT response generation results, this work uses DialogBERT [46]. This thesis uses 'bertbase-uncased' which is fine tuned using the DialogBERT approach. Since the base-size model is vulnerable to over-fitting when using small datasets, the response generation model is trained with small-sized models. The small-size model reduces the 'bertbase-uncased' configuration to 6 transformer layers, has a hidden size of 256, and contains 2 attention heads (L=6, H=256, A=2). All of the experiments use the default BERT tokenizer. The model is fine tuned on the two datasets for 2 epochs with a max sequence length of 200. The AdamW optimizer, a learning rate of 5e-5, and a batch size of 4 is used for training and validation. The format of a training instance for response generation is shown in Figure 4.19.

```
"[CLS] Utterance1.... [SEP] Utterance 2 [SEP].... [PAD]"
```

Figure 4.19: BERT format of a training instance for response generation

- **Gemini response generation.** For the Gemini response generation results, this work

uses Geminin API with gemini-1.0-pro [126]. This work use the default supervised learning fine tuning setting for Gemini API. The model is fine tuned on the datasets for 2 epochs with a learning rate of 1. The format of a training instance/prompt for response generation is shown in Figure 4.20.

"messages": ["role": "system", "content": "Based on the following conversation history generate the next possible utterance", "role": "user", "content": "CONVERSATION: Utterance 1:.... Utterance 2:... LABEL:", "role": "model", "content": "Next utterance"]

Figure 4.20: Gemini format of a training instance/prompt

- **Llama response generation.** For the Llama response generation results this work uses the Hugging Face fine-tuning with PEFT QLoRA with 'Llama-2-7b'. With QLoRA pre-trained models are loaded to GPU as quantized 4-bit weights. QLoRA is required since a single GPU is used to fine-tune the model. The model is fine tuned on the two datasets for 2 epochs with a max sequence length of 200. The AdamW optimizer, a learning rate of 2e-5, and a batch size of 4 is used for training and validation. The format of a training instance/prompt for response generation is shown in Figure 4.21.

"text": <s>[INST] «SYS»"Based on the following conversation history generate the next possible utterance" «/SYS»Utterance 1:.... [/INST] Utterance 2:... </s><s>[INST] Next utterance</s>

Figure 4.21: Llama format of a training instance/prompt

Model performance with response generation

Table 4.12 shows the results of including a LLM-generated conversation into the text layer of the FGCN-nEMC model. The table uses the best performing model of FGCN-nEMC for

LLM	IEMOCAP	MELD
GPT	65.21	39.78
BERT	65.32	39.98
Gemini	63.78	37.83
LlaMA	63.98	38.32
None	<u>65.62</u>	<u>40.34</u>
Actual	**67.9**	**52.23**

Table 4.12: Experimental results for forecasting conversation derailment with static training including a generated last utterance N. This tables shows the F1-score. Best F1-score for each dataset are in bold. Second best are underlined. The FGCN-nEMC used here uses the best performing model shown in Table 4.8. Unsurprisingly the model that uses the actual next text scores best. But the second best is to not use any generated text at all

both the IEMOCAP and MELD dataset. ALL variants of the model that include a LLM generated last utterance under-perform in predicting the nest emotion when infused with the FGCN-nEMC. The table shows that the variant without a LLM generated last turn's text outperforms all variants with a generated last turn's text regardless of the LLM used to generate the last turn's text. The results also vary in performance depending on the LLM used, with BERT providing the best result. This suggests that estimating the last turn's text using a SLM (Bert and GPT) or a LLM (Llama and Gemini) result in added noise to the data through an estimated utterance. One limitation of this work with the LLM embedding is the use of straightforward prompt for the LLM which may not be able to harness the power of LLM without using methods to specialize the LLM. Other than fine tuning, examples of such approaches include PAFT and RAG.

4.4 Summary

Unlike previous models which were based on simpler sequence models, FGCN-nEMC and DialogInfer use a graph convolutional neural network built on top of the basic sequence models. A closer look to the graph model introduced in DialogInfer shows that the model uses an attention mechanism to infer the participant's next emotion by directly modeling the emotion shifts relating to persistence and contagion. The FGCN-nEMC model reduces the noise in its user relational graph through graph sparsification that models recency and self-dependency. DialogInfer outperforming FGCN-nEMC indicates that the attention mechanism used in DialogInfer has a higher effect on performance than the approved structure of FGCN-nEMC. One possible direction to improve overall performance would be to combine both of these graph optimizations into the graph portion of the data extensible FGCN-nEMC model.

The FGCN-nEMC model architecture is an extensible architecture with layers of homogeneous graphs in which different data can be included through additional layer. Results show that extending the model with additional layers encoding different data types results in enhanced performance. This architecture also supports the study of impact of the different types of data. Results show that emotion followed by common sense knowledge data are the most influential sources of data in terms of model performance. It is perhaps not surprising that encoding the emotion data directly has a significant impact. The emotion data contains the emotion shifts that relate to persistence and contagion. Commonsense knowledge features encode complex characteristics including the dynamics of multi-party dialogue, context propagation and mood shifts in conversations.

The architecture also allows for varying the classifier and understanding the impact of simple and more complex classifiers on performance. Results show that a Transformer classifier (E, ED) provides superior performance than the simple fully connected network classifier. However, the encoder-decoder Transformer classifier (ED) performance is similar to

the encoder only Transformer classifier (E). This suggests that the complexity of ED does not necessarily provide significant benefit over an encoder only model for this binary classification task. This is to be expected as the decoder here is limited by the simplicity of the underlying problem when compared to more complex decoding tasks.

When varying the text embedding populating the nodes in the graph neural network results show that using SLM embeddings (GPT and Bert) outperform the use of LLM embeddings for nEMC. The results also show that fine tuning before embedding extraction results in better performance in the FGCN-nEMC models.

When including an estimated last utterance into the data input to the FGCN-nEMC model, the results show that the variant without a LLM generated last utterance outperforms all variants with a generated last utterance regardless of the LLM tested to generate the last utterance. One possible reason for LLMs under performing is that LLMs are sensitive to prompt engineering since they are more generalized models. The prompt engineering used to obtain the results in this work are straight forward. More sophisticated prompts that incorporate more context instruction and/or RAG could produce better results.

Chapter 5

Forecasting conversation derailment (fConvD)

5.1 Introduction

Emotional derailment in conversations refers to the process by which a conversation or discussion is redirected away from its original topic or purpose, typically as a result of inappropriate or off-topic comments or actions by one or more participants. In online conversations, derailment can be exacerbated by the lack of nonverbal cues and the presumed anonymity provided by the internet leading to escalation with anger and frustration. In Chapter 3 a collection of models for forecasting in conversations, how the input to these models could be structured, and the architecture of an AI that could be used to model such conversations and predict derailment was presented. Here we describe the process of training these architectures and their performance on standard datasets. Figure 5.1 shows an example conversation, coming from the popular Reddit ChangeMyView (CMV) benchmark dataset. The goal here is to forecast the derailment label l of the n^{th} utterance given a conversation $\mathcal{C}$ from 1 up to $n-1$ turns that has not derailed prior to turn n. This chapter discusses the specifics of graph construction used in the forecasting model described in Chapter 3 to obtain a forecasting model for conversation derailment FGCN-fConvD, shown in Figure 5.2, and a

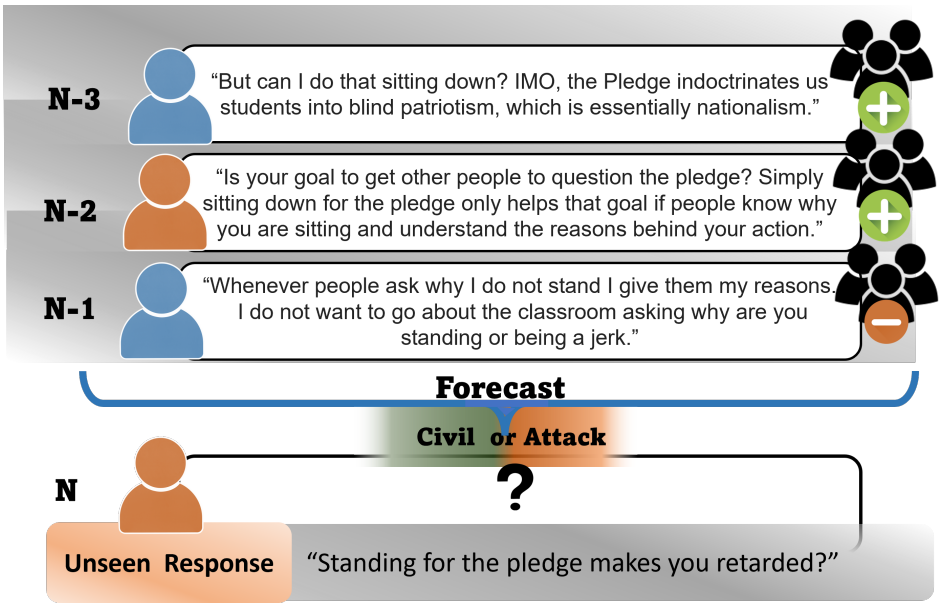


Figure 5.1: The problem of forecasting derailment

comparative study of the forecasting model against standard baselines in the literature for the derailment forecasting problem.

5.2 Graph construction: user to user edge relationship

Given the availability of the conversation structure we can establish the direct user to user relationship in a conversation and represent this structure through the edge construction of a graph that represents the conversation. This results in a more efficient graph model with reduced complexity compared to the complete graph model used in the literature [42, 68] which does not model this information. Figure 5.3 shows an example graph edge construction for a given conversation. This conversation is an online conversation with comments presented in a parent-child tree-like nature. What is known as a comment in online conversations is referred to as a turn here. In the user-to-user relationship edge construction, each vertex v_i represent a turn in the conversation. The vertex has directed edges connecting it to its

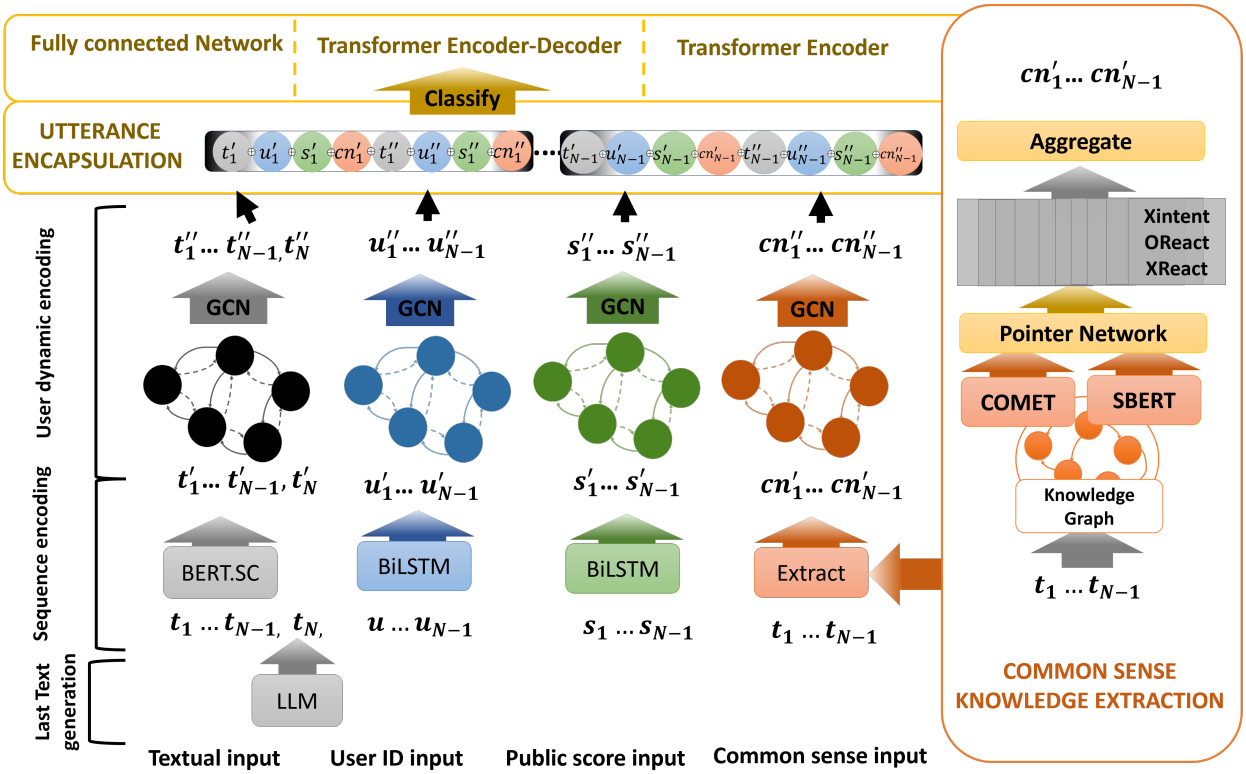


Figure 5.2: The FGCN-fConvD model architecture, A varinat of FGCN for forecasting derailment in conversations. A sample encoding of a conversation of length 6 with three classes of optional data (S and CN) is shown. For each type of data the model separately (shown in the data layers) uses sequential encoding to populate the vertices in a graph where it undergoes GCN transformation. The results of the sequential encoder and user dynamics encoder are concatenated for classification. The details of graph construction in the user dynamics encoding are shown in Figure 5.3.

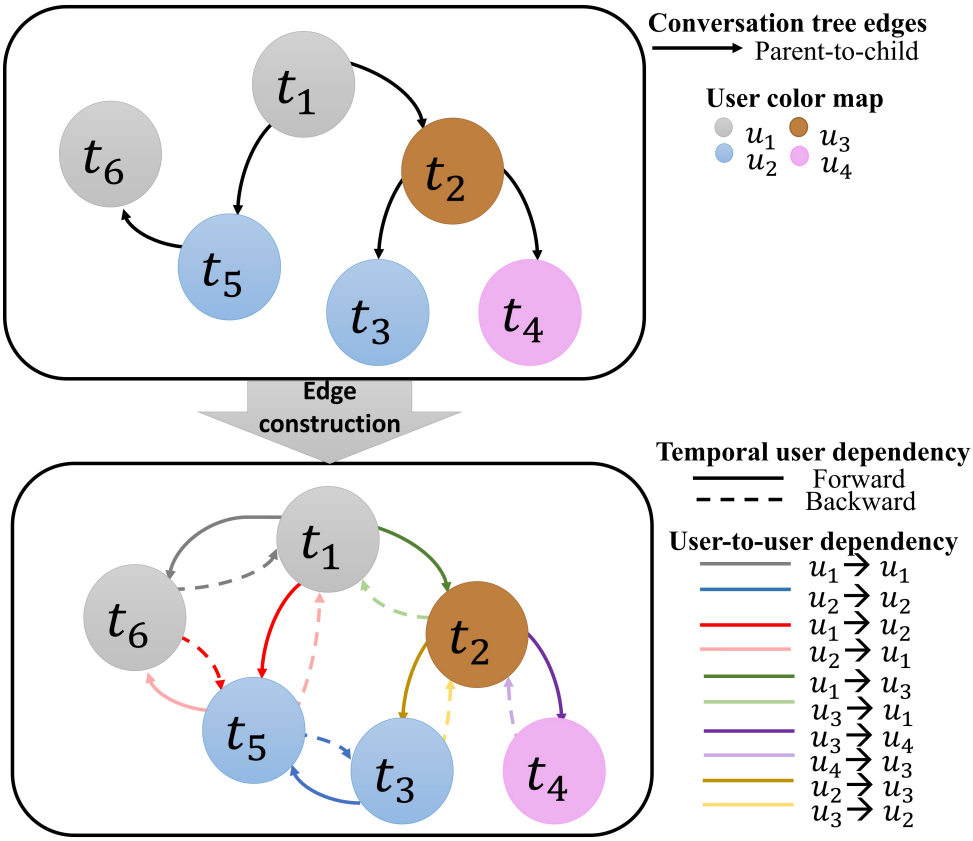


Figure 5.3: An example graph edge construction of six turns for the given conversation with user-to-user and temporal dependency. The vertices represent the text associated with a turn. In this example, the turn text features in the graph are encoded with the letter t and a serial number describing the utterances' sequential occurrence in time. The turn generated by users is encoded with different vertex colors. The structure of the conversation is based on a parent-child relationship as presented in online conversations. The edge colors encode the edge user-to user relationship. The edge temporal user dependency is encoded in with a solid or dashed line. We add same speaker edges to the conversation natural parent-child structure.

(preceding or parent) and (succeeding or child) turns in the conversation. The child turn is a direct reaction to the parent turn. Same user turns are also connected through directed edges that represent "same user". The formation of graphs based on this natural conversation structure is generally neglected in the literature [42, 68].

We also establish all same user edges between utterances of the same user to model sensitivity to self dependency. This direct modeling of recency and self dependency and the investigation of its importance is currently neglected in the literature [42, 68].

The user-to-user relation R of an edge r_{ij} is based on the user-to-user dependency between user u_i (turn i) and user u_j (another turn j). For example, in Figure 5.3, there are four users, so the set of edge labels R has the relation types shown under user-to-user dependency. As the graph is directed, two vertices can have edges in both directions with different relations. This is used to represent the past (preceding) and future (succeeding) temporal dependency between the vertices, shown as temporal user dependency in the Figure. Following [42], the edge weights are set using a similarity based attention module. The attention function is computed such that, for each vertex, the incoming set of edges has a sum total weight of 1. The weights are calculated as, $\alpha_{ij} = \text{Softmax}(\hat{x}_i \odot W_e)$ where $\odot$ is the Hadamard product and W_e is the vector of weights associated with the edges connecting vertex i to vertex j.

5.3 Forecasting derailment datasets

Two datasets are commonly used for the task of forecasting derailment in conversations; The Conversations Gone Awry (CGA) and Reddit ChangeMyView (CMV). The datasets are summarized in Table 5.1. See also Appendix A. The splits introduced in the datasets are maintained throughout the study to insure comparability with previous studies.

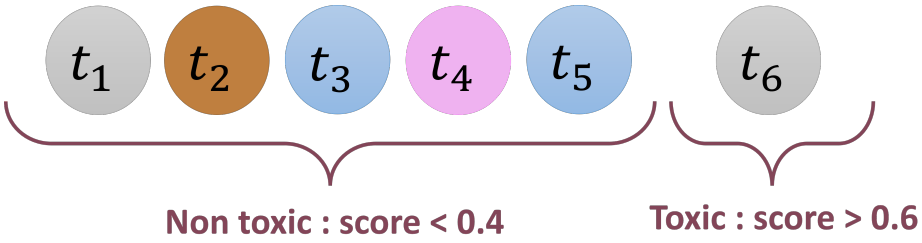


Figure 5.4: Conversation sampling in the CGA dataset.

5.3.1 The Conversations Gone Awry dataset(CGA)

The Conversations Gone Awry (CGA) dataset [141] was extracted from Wikipedia Talk Page conversations. The conversations were sampled from WikiConv [57] based on an automatic measure of toxicity that ranges from 0 (not toxic) to 1 (is toxic). A conversation is extracted as a sample of derailment if the nth comment in it has a toxicity score higher than 0.6 and all the preceding comments have a score lower than 0.4. Conversations having all comments with a toxicity score below 0.4 are extracted as samples of non-derailment. See Figure 5.4. The set of conversations with the potential for derailment is further filtered through manual annotation to determine whether after an initial civil exchange a personal attack occurs from one user towards another. The conversations include the last turn n as the turn with a possible personal attack. This means that the earlier $n-1$ turns in these conversations are civil and the n'th one is either civil or contains a personal attack. This provides a dataset of derailed conversations and civil conversations. The dataset also contains additional information about each comment in the conversation such as the identification of the user posting the comment and the parent-child relationship between the comments. This dataset contains 4,188 conversations, split 60-20-20 (training-validation-testing), with a balanced representation of the two classes.

Dataset	Train	Val	Test	Input		
				t	u	s
CGA	2508	840	840	✓	✓	✗
CMV	4106	1368	1368	✓	✓	✓

Table 5.1: Statistics of the datasets. t denotes text input, u denotes user ID input and s denotes public perception score input. All splits are balanced between the two classes.

5.3.2 The Reddit ChangeMyView dataset(CMV)

The Reddit ChangeMyView (CMV) dataset [23] was extracted from Reddit conversations held under the ChangeMyView subreddit. Conversations were identified as derailed if there was a deletion of a turn by the platform moderators. This could have been done under Reddit's Rule: "Don't be rude or hostile to other users." Unlike CGA, there is no control to ensure that all the preceding prior comments to the last one would be civil, resulting in some noise in the data. See Figure 5.5. The dataset also contains additional information about each comment in the conversation such as the user posting the comment, the ID of the parent comment that this comment was a reply to, and a votes score (i.e., the number of up-votes minus the number of down-votes). Up-votes signify that the community supports the turn text, down-votes signify that they do not support it. The number of votes is an indication of the community's interest in the conversation or the speakers. This dataset consists of 6,842 data points, with a split of 60-20-20 (training-validation-testing), balanced between the two classes (derail, no-derail).

5.4 Baselines

The FGCN-fConvD model and its variants are evaluated against the state-of-the-art methods for conversation derailment listed below. The baseline results shown here are reported directly

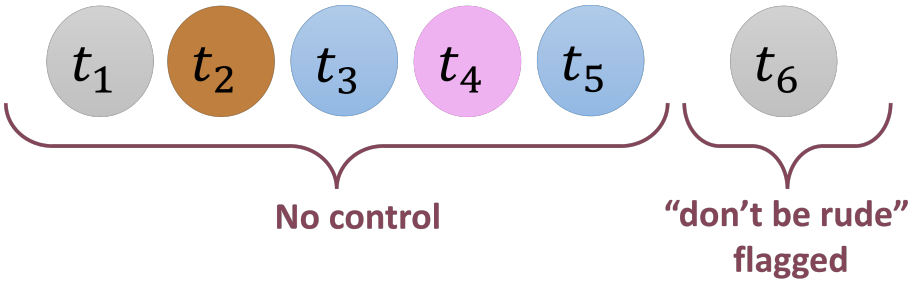


Figure 5.5: Conversation sampling in the CMV dataset.

from [68]. The evlauation criteria in this thesis follows the same methodology used in [68] to ensure comparability.

- **CRAFT** [23] is a model with a hierarchical recurrent neural network architecture, which integrates a generative dialog model that learns to represent conversational dynamics in an unsupervised fashion, and a supervised component that fine-tunes this representation to forecast future events. This model is trained statically.

- **BERT.SC** [68] is a model consisting of the BERT checkpoint with a sequence classification (SC) head, trained statically. BERT.SC uses a learning rate of 1e-5 and a batch size of 16.

- **BERT.SC+** [68] similar to BERT·SC consists of the BERT checkpoint with a sequence classification (SC) head, but is instead trained dynamically. BERT.SC+ uses a learning rate of 1e-5 and a batch size of 16.

- **HR-Multi** [137] is a hierarchical transformer-based framework that combines utterance-level and conversation-level information. This model is trained dynamically. HR-Multi uses a learning rate of 1e-5 and a batch size of 16.

	Model	CGA				CMV			
		Acc	P	R	F1	Acc	P	R	F1
Static	CRAFT	64.4	62.7	71.7	66.9	60.5	57.5	81.3	67.3
	BERT·SC	64.7	61.5	79.4	69.3	62.0	58.6	82.8	68.5
	FGCN-fConvD-T	66.4	63.0	79.5	70.3	62.9	59.2	83.0	69.1
	FGCN-fConvD-TU	66.9	63.3	80.2	70.8	63.2	59.5	83.0	69.3
	FGCN-fConvD-TCN	67.2	63.5	80.8	<u>71.1</u>	63.7	59.8	83.1	70.0
	FGCN-fConvD-TS	-	-	-	-	64.2	60.3	83.2	69.9
	FGCN-fConvD-TSU	-	-	-	-	64.7	60.7	83.3	70.2
	FGCN-fConvD-TUCN	67.4	63.7	81.0	**71.3**	65.9	61.8	83.5	<u>70.8</u>
	FGCN-fConvD-TSUCN	-	-	-	-	66.6	62.7	84.1	**71.1**
Dynamic	BERT·SC+	64.3	61.2	78.9	68.8	56.5	56.0	73.2	61.7
	HR-Multi+	65.2	62.3	76.9	68.9	64.2	62.0	73.8	67.4
	FGCN-fConvD-T+	65.7	62.2	79.7	69.9	62.1	58.5	82.0	68.3
	FGCN-fConvD-TU+	65.9	62.4	80.2	70.2	62.7	58.8	82.7	68.8
	FGCN-fConvD-TS+	-	-	-	-	62.9	59.2	82.9	69.1
	FGCN-fConvD-TCN+	66.3	62.8	80.5	<u>70.4</u>	63.2	59.8	83.2	69.3
	FGCN-fConvD-TSU+	-	-	-	-	64.1	60.5	83.5	69.5
	FGCN-fConvD-TUCN+	66.7	63.0	81.0	**70.9**	64.4	60.7	83.7	69.6
	FGCN-fConvD-TSCN+	-	-	-	-	64.7	60.9	84.0	<u>69.7</u>
	FGCN-fConvD-TSUCN+	-	-	-	-	65.0	61.1	84.2	**69.9**

Table 5.2: Experimental results for forecasting conversation derailment. Accuracy (Acc), precision (P), recall (R), and F1-score (F1) are reported. Best F1-score scores are in bold. Second best results are in underlined.

5.5 Results and discussion

In the forecasting conversation derailment experiment we report the results in terms of accuracy (Acc), precision (P), recall (R), and F1-score (F1) of the static and dynamic training of our model and compare performances with baselines. When analyzing mean forecast horizon experiment we report how early each model can forecast derailment. The forecast

horizon H introduced by [68] is reported, which is the mean of the turns in which the first detection of derailment occurred for the set of conversations that derail. If the model was unable to detect derailment the value is set to the length of the conversation n. This enables a more straightforward comparison with this earlier work that use these metrics.

5.5.1 Forecasting conversation derailment

The results provided in Table 5.2 show that based on F1-score performance across both datasets FGCN-based approaches outperform baseline models for both static and dynamic training. To insure that the results shown are representative of the models performance, all results shown is this table are the average of 10 training and testing runs. In the static training, the FGCN-fConvD-T variant that uses only text input outperforms the baselines that also only use text as input (CRAFT and BERT.SC). The results also show that incorporating more input types leads to better performance, for both static and dynamic trained models. When comparing FGCN-fConvD-TU, FGCN-fConvD-TS and FGCN-fConvD-TCN, FGCN-fConvD-TCN outperforms the other variants by including common sense knowledge in the forecasting graph neural network model provides the highest impact, improving the models' forecasting F1-score. This suggests the effectiveness of incorporating common sense knowledge for conversation derailment forecasting. The model variants FGCN-fConvD-TSUCN for the CMV dataset and FGCN-fConvD-TUCN for the GCA dataset are variants that incorporate all available textual data for the forecasting problem. These variants outperform all other variants evaluated indicating that including more data context enables better performance in the derailment forecasting problem. Similar results can be seen for the dynamic training results. See Table 5.2.

These results confirm previous studies comparing static and dynamic training for the problem of derailment forecasting [68, 5]. For all models that are trained both statically and

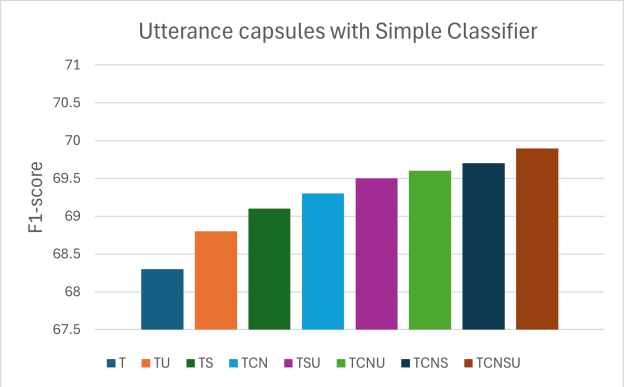
	Classifier		
Variant	**Simple(S)**	**Encoder(E)**	**Encoder-Decoder(ED)**
T	68.3	68.7	68.8
TU	68.8	69.1	69.2
TS	69.1	69.5	69.5
TCN	69.3	69.8	69.9
TSU	69.5	69.9	70.0
TUCN	69.6	70.1	70.1
TSCN	<u>69.7</u>	<u>70.2</u>	<u>70.3</u>
TSUCN	**69.9**	**70.5**	**70.5**

Table 5.3: Varying the utterance information capsule and classifier for CMV using dynamic training in FGCN-fConvD. S(Simple), E(Encoder) and ED(Encoder-Decoder) are the types of classifiers used to obtain the reported F1-score. The best performing information capsules are shown in bold and second best are shown underlined for each classifier. TSUCN outperforms other models for the classifiers evaluated.

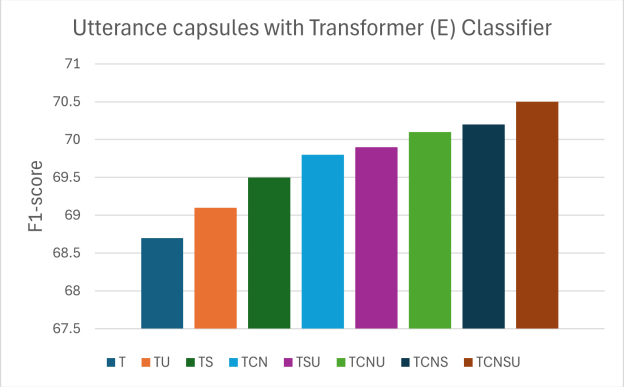
dynamically, statically trained models outperform their corresponding dynamically trained models. This comparison is not available for HR-Multi as it is only dynamically trained. Static training contains all context for every sample. It is a natural result for the F1-score to improve when the entire conversation context is provided in the training process. In addition, our model dynamically trained on the CGA dataset outperform statically trained baselines indicating the effectiveness of graph neural networks in overcoming the lack of context in a given conversation instance.

5.5.2 Utterance capsule information and classifier type sensitivity

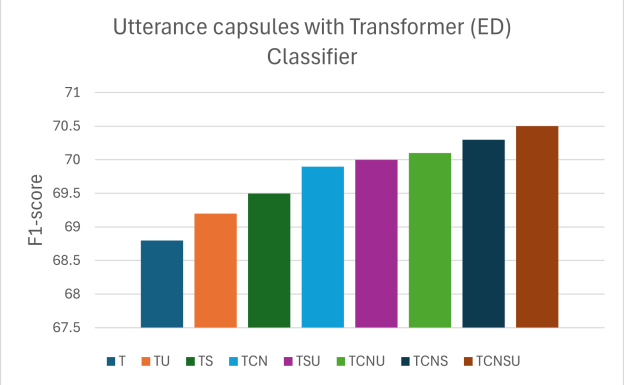
To understand the impact of the types of data included in the information capsules passed to the classifier and the type of classifier used, the results of these decisions on performance using the CMV dataset is discussed in this section. Table 5.3 and Figure 5.6 summarize performance



(a) Simple

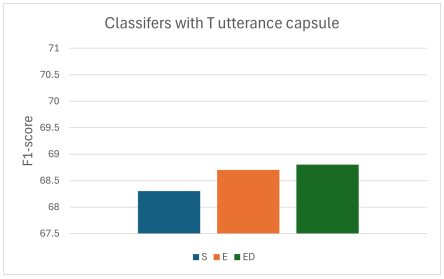


(b) Transformer(E)

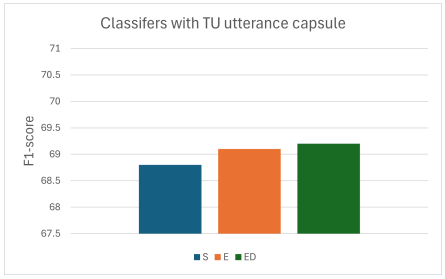


(c) Transformer(E)

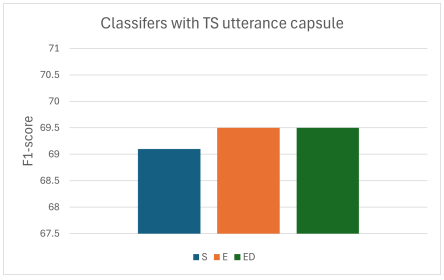
Figure 5.6: Impact of the types of data included in the information capsules passed to the classifier. More data in the information capsule enables enhanced performance. information capsules with CN, outperforms Information capsules without CN using the same classifier.



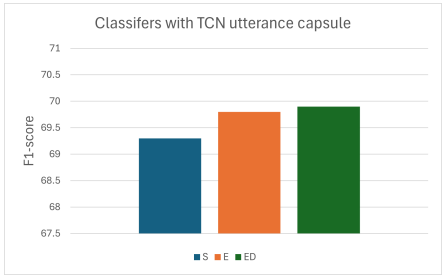
(a) T capsule



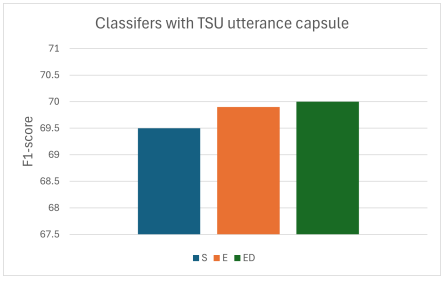
(b) TU capsule



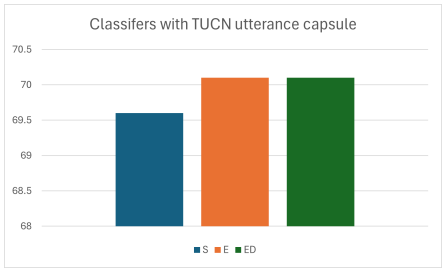
(c) TS capsule



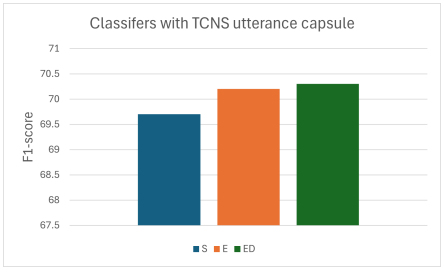
(d) TCN capsule



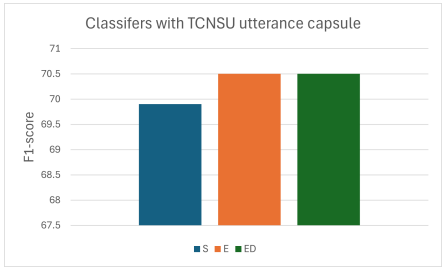
(e) TSU capsule



(f) TUCN capsule



(g) TCNS capsule



(h) TCNSU capsule

Figure 5.7: The classifier performance for each utterance capsule is shown for the CMV dataset. The Transformer classifier (E, ED) is better than the simple fully connected network classifier. However, the encoder-decoder Transformer classifier (ED) performance is similar to the encoder only Transformer classifier (E).

using different variants on the CMV dataset. More data enables enhanced performance over less data. These results also show that FGCN-fConvD with CN, that is informed by ,common sense knowledge, in its information capsule, outperforms FGCN-fConvD without CN using the same classifier. This indicates the importance of incorporating common sense knowledge in creating a knowledge aware forecasting model that is able to capture context propagation and mood shifts during the conversation and improve performance.

As shown in Table 5.3 and Figure 5.7, using a Transformer classifier (E, ED) provides better performance than the simple fully connected network classifier. However, the encoder-decoder Transformer classifier (ED) performance is similar to or slightly better than the encoder only Transformer classifier (E). The additional complexity of (ED) related to (E) does not always provide a better result, and if it does so then only slightly better perfomance is observed. This may be due to the binary nature of the classification task. Since the decoder here is limited by the simplicity of the underlying problem when compared to more complex decoding tasks.

5.5.3 Varying the text embedding

We vary the text embedding that initializes the graph neural network in order to understand the impact of SLM verses LLM embeddings. LLM embeddings are more generalized compared to SLM embeddings. Varying the text embedding allows us to investigate the influence of using more generalized embedding on forecasting conversation derailment.

- **GPT embedding extraction.** For non-fine-tuned embedding extraction the OpenAI GPT-2 is used. The fine tuned embedding extraction uses DialogGPT [144]. The DialogGPT repository is based on the huggingface pytorch-transformer and OpenAI GPT-2, containing a data extraction script, model training code and pretrained small (117M) medium (345M) and large (762M) model checkpoints. The model is fine tuned

using the small pretrained model to reduce the chances of over fitting due to the size of our dataset. The model is fine tuned on the two datasets for 2 epoches with a max sequence length of 200. The AdamW optimizer, a learning rate of 1e-5, and a batch size of 4 is used for training and validation. The format of the training instance for the fine tuned model is given in Figure 5.8.

```
"Utterance1.... [EOS] Utterance 2 [EOS].... [PAD]"
```

Figure 5.8: GPT2 format of a training instance for the fine tuned model for embedding extraction.

- **BERT embedding extraction.** For non-fine-tuned embedding extraction the 'bertbase-uncased' is used. The fine tuned embedding extraction uses 'bertbase-uncased' which is fine-tuned using the DialogBERT approach [46]. Since the base-size model is vulnerable to over-fitting when using small datasets. The response generation model is trained with small-sized models. The small-size model reduces the 'bertbase-uncased' configuration to 6 transformer layers, has a hidden size of 256, and contains 2 attention heads (L=6, H=256, A=2). All experiments use the default BERT tokenizer. The model is fine tuned on the two datasets for 2 epochs with a max sequence length of 200. The AdamW optimizer, a learning rate of 5e-5, and a batch size of 4 is used for training and validation. The format of the training instance for the fine tuned model is given in Figure 5.9.

- **Gemini embedding extraction.**, For non-fine-tuned embedding extraction Geminin API with gemini-1.0-pro is used directly [126].For fine-tuned embedding extraction the default supervised learning fine tuning setting for Gemini API is used. The model is fine tuned on the datasets for 2 epochs with a learning rate of 1. The format of the

"[CLS] Utterance1.... [SEP] Utterance 2 [SEP].... [PAD]"

Figure 5.9: BERT format of a training instance for the fine tuned model for embedding extraction.

training instance/prompt is given in Figure 5.10.

"messages": ["role": "system", "content": "Based on the following conversation, determine if the conversation derails into one of the following classes: [True, False]. The output should be a single label, either True or False.", "role": "user", "content": "CONVERSATION: Utterance 1:.... Utterance 2:... LABEL:", "role": "model", "content": "label value"]

Figure 5.10: Gemini format of a training instance/prompt.

- **Llama embedding extraction.** For non-fine-tuned embedding extraction the 'Llama-2-7b' is used. For fine-tuned embedding extraction the Hugging Face fine-tuning with PEFT QLoRA with 'Llama-2-7b' is used. With QLoRA, pre-trained models are loaded to the GPU as quantized 4-bit weights. QLoRA is required since a single GPU is used to fine-tune the model. The model is fine tuned on the two datasets for 2 epochs with a max sequence length of 200. The AdamW optimizer, a learning rate of 2e-5, and a batch size of 4 is used for training and validation. The format of the training instance/prompt is given in Figure 5.11.

The results presented in Table 5.4 use the best FGCN-fConvD variant for the CMV dataset, FGCN-fConvD-TSUCN, and the best variant for the GCA dataset, FGCN-fConvD-TUCN, as the base FGCN-fConvD models for varying the embeddings.

"text": <s>[INST] «SYS»"Based on the following conversation, determine if the conversation derails into one of the following classes: [True, False]. The output should be a single label, either True or False."«/SYS»Utterance 1:.... [/INST] Utterance 2:... </s><s>[INST] label value</s>

Figure 5.11: Llama format of a training instance/prompt.

In this section the results of varying the text embedding extracted from language models is shown with and without fine-tuning. The results show that fine tuning the language models provides better results when compared to using extracted embedding without fine tuning. These results confirm previous findings [143] that LLMs' have unsatisfactory performance in many emotion recognition tasks without fine-tuning on emotion knowledge. Table 5.4 shows variants of the model using the embeddings of two SLM models GPT and Bert and two LLM models Gemini and Llama.

The results in Table 5.4 show that SLM GPT and Bert outperforms LLMs for derailment foresting problem. LLMs are sensitive to prompt engineering since they are more general models. The prompt engineering used to obtain the results in the table are straight-forward. More sophisticated prompts that incorporate more context instruction and/or RAG may produce better results. We also observe that static training trumps dynamic training in both fine tuned and not fine tuned models. When available, more context in the training examples produces better results.

5.5.4 Model infused with response generation

This section explores infusing the FGCN-fConvD model with a LLM-generated last turn's text n in the text layer and compare performance against variant without the last turn's text.

		Model	CGA	CMV
Not fine tuned	Static	FGCN-fConvD-GPT	<u>67.8</u>	<u>65.9</u>
		FGCN-fConvD-BRT	**68.9**	**66.3**
		FGCN-fConvD-Gemini	63.3	60.2
		FGCN-fConvD-Llama	64.5	61.2
	Dynamic	FGCN-fConvD-GPT+	<u>65.4</u>	<u>63.1</u>
		FGCN-fConvD-BRT+	**67.8**	**65.2**
		FGCN-fConvD-Gemini+	62.4	59.3
		FGCN-fConvD-Llama+	63.6	60.4
Fine tuned	Static	FGCN-fConvD-GPT	<u>69.2</u>	<u>67.3</u>
		FGCN-fConvD-BRT	**71.3**	**71.1**
		FGCN-fConvD-Gemini	66.4	63.1
		FGCN-fConvD-Llama	67.5	64.3
	Dynamic	FGCN-fConvD-GPT+	<u>68.6</u>	<u>66.8</u>
		FGCN-fConvD-BRT+	**70.9**	**69.9**
		FGCN-fConvD-Gemini+	65.4	62.7
		FGCN-fConvD-Llama+	66.3	63.4

Table 5.4: Experimental results for forecasting conversation derailment using SLM and LLMs embedding with and without fine tuning. This tables shows the F1-score. The FGCN-fConvD used here uses the best performing model shown in Table 5.2. Best results are in bold and second best are underlined.

.

Response generation models

- **GPT response generation.** For the GPT response generationGPT response generation results, this work uses DialogGPT [144]. The DialogGPT repository is based on huggingface pytorch-transformer and OpenAI GPT-2, containing a data extraction

script, model training code and pretrained small (117M) medium (345M) and large (762M) model checkpoints. The model is fine tuned using the small pretrained model to reduce the chances of over fitting due to the size of our dataset. The model is fine tuned on the two datasets for 2 epoches with a max sequence length of 200, AdamW optimizer, a learning rate of 1e-5, and a batch size of 4 is used for training and validation. The format of the training instance for response generation is given in Figure 5.12.

```
"Utterance1.... [EOS] Utterance 2 [EOS].... [PAD]"
```

Figure 5.12: GPT2 format of a training instance for response generation.

- **BERT response generation.** For the BERT response generation results, this work uses DialogBERT [46]. This work uses 'bertbase-uncased' which is fine tuned using the DialogBERT approach. Since the base-size model is vulnerable to over-fitting in small datasets, the response generation model is trained with small-sized models. The small-size model reduces the 'bertbase-uncased' configuration to 6 transformer layers, has a hidden size of 256, and contains 2 attention heads (L=6, H=256, A=2). All of the experiments use the default BERT tokenizer. The model is fine tuned on the two datasets for 2 epochs with a max sequence length of 200, AdamW optimizer, a learning rate of 5e-5, and a batch size of 4 for training and validation. The format of the training instance for response generation is shown in Figure 5.13.

```
"[CLS] Utterance1.... [SEP] Utterance 2 [SEP].... [PAD]"
```

Figure 5.13: BERT format of a training instance for response generation.

- **Gemini response generation.** For the Gemini response generation results, this

work uses Geminin API with gemini-1.0-pro [126]. The default supervised learning fine tuning setting for Gemini API is used. The model is fine tuned on the datasets for 2 epochs with a learning rate of 1. The format of the training instance/prompt for response generation is shown in Figure 5.14.

```
"messages": ["role": "system", "content": "Based on the following conversation history generate the next possible utterance", "role": "user", "content": "CONVERSATION: Utterance 1:.... Utterance 2:... LABEL:", "role": "model", "content": "Next utterance"]
```

Figure 5.14: Gemini format of a training instance/prompt.

- **Llama response generation.**For the Llama response generation results this work uses Hugging Face fine-tuning with PEFT QLoRA with "Llama-2-7b" with QLoRA pre-trained models loaded to the GPU as quantized 4-bit weights. QLoRA is required since a single GPU is used to fine-tune the model. The model is fine tuned on the two datasets for 2 epochs with a maximum sequence length of 200. The AdamW optimizer, a learning rate of 2e-5, and a batch size of 4 for training and validation were used. The format of the training instance/prompt for response generation is shown in Figure 5.15.

```
"text": <s>[INST] «SYS»"Based on the following conversation history generate the next possible utterance" «/SYS»Utterance 1:.... [/INST] Utterance 2:... </s><s>[INST] Next utterance</s>
```

Figure 5.15: LLama format of a training instance/prompt.

LLM	CGA	CMV
GPT	70.3	70.1
BERT	70.5	70.2
Gemini	69.2	68.4
LlaMA	69.9	69.3
None	<u>71.3</u>	<u>71.1</u>
Actual	**85.4**	**79.7**

Table 5.5: Experimental results for forecasting conversation derailment with static training including a generated last utterance n. The FGCN-fConvD used here uses the best performing model shown in Table 5.2

Model performance with response generation

Table 5.5 shows the results of including a LLM generated conversation into the text layer of the FGCN-fConvD model. The table uses the best performing model of FGCN-fConvD for both datasets, FGCN-fConvD-TSUCN for CMV and FGCN-fConvD-TUCN for CGA. The results show that ALL variants of the model that include a LLM-generated last utterance under-perform. The table shows that the variant without a LLM generated conversation last utterance outperforms all variants with a generated last utterance regardless of the LLM used to generate the last utterance. The results also vary in performance depending on the LLM used, with BERT providing the best result. This suggests that estimating the last utterance using SLM (Bert and GPT) or LLM (Llama and Gemini) result in added noise to the data through an estimated utterance. The limitations of this work with the LLM embedding is the use of straightforward approaches to prompt the LLM which may not be able to harness the power of LLM without using methods to specialize the LLM. Other than fine tuning, examples of such approaches include PAFT and RAG.

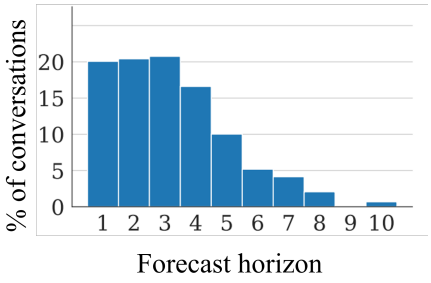
Mean forecast horizon		
Model	**CGA**	**CMV**
CRAFT	2.36	4.01
BERT·SC	2.60	3.90
BERT·SC+	2.85	4.06
HR-Multi	2.98	3.78
FGCN-fConvD-T	2.73	4.03
FGCN-fConvD-T+	<u>2.96</u>	<u>4.12</u>
FGCN-fConvD-TCN	2.90	4.11
FGCN-fConvD-TCN+	**3.02**	**4.16**

Table 5.6: Experimental results of mean forecast horizon (H). The best result is shown in bold. The second best result has been underlined. The + sign denotes dynamically trained models.

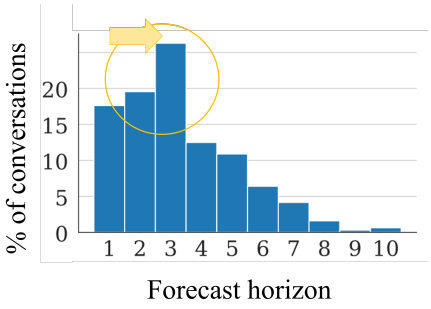
5.5.5 Analyzing mean forecast horizon

How early can the model forecast the derailment? To answer this question we calculate the forecast horizon H, the mean of the turn in which the first detection of derailment occurred for the set of conversations that derail. A forecast horizon H of 1 means that a derailment occurring on turn n was first detected on turn $n-1$. If the model was unable to detect derailment in one of the conversation in the set that derails used here, the model will use the length of the conversation n as the value forecast horizon H. A longer forecast horizon (i.e., a higher H) allows for earlier interventions and potentially allows moderators to respond to an upcoming personal attack earlier. Models that are able to detect a potential intervention earlier have a clear advantage.

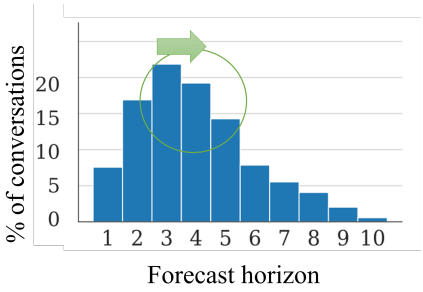
In Table 5.6 we report the results of the mean forecast horizon H. Results show that FGCN-fConvD+ models with its dynamic training provides the earliest overall forecasting of derailment with its commonsense knowledge variant FGCN-fConvD-TCN+ obtaining a mean H of 4.16 for CMV and, 3.02 for CGA. Followed by other dynamically trained models



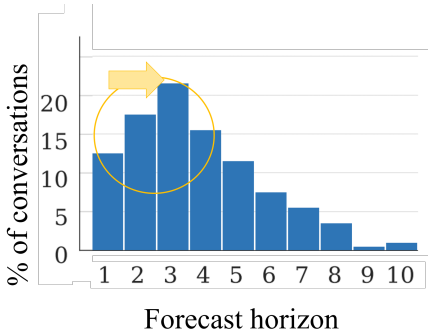
(a) CRAFT



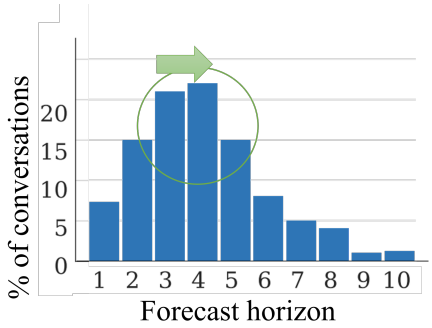
(b) BERT-SC



(c) BERT-SC+



(d) FGCN-fConvD



(e) FGCN-fConvD+

Figure 5.16: Forecast horizon on the CGA dataset with a model drawn at random from among the 10 available ones. A horizon of 1 means that an upcoming derailment was only predicted on the last turn before it occurred. The bars indicate the percentage of conversations that where successfully predicted to derail at the forecast horizon. The yellow arrow indicates an shift towards a better forecast horizon and the green arrow indicates a more prominent shift towards better forecast horizon.

BERT·SC+ for CMV and HR-Multi for CGA. For the statically trained models (CRAFT, BERT.SC, FGCN-fConvD-T and FGCN-fConvD-TCN), the FGCN models have the best performance as it seems to be able to better model the dynamic relationships between the users of the turns with its graph model, with its common sense knowledge variant FGCN-fConvD-TCN obtaining a mean H of 4.11 for CMV and, 2.90 for CGA. The results also show that FGCN-fConvD-TCN, a knowledge aware variant with common sense knowledge CN, has the best performance overall as it seems to not only capture the dynamics of multi-party dialogue but also context propagation and mood shifts.

We perform further analysis of the forecast horizon results for CGA. Figure 5.16 illustrates the forecast horizon of the different models. In earlier work [23, 68], both CRAFT and BERT·SC models make a prediction with a forecast horizon $H > 1$ turn at a high rate. Only 20% of CRAFT's forecasts and 17.5% of BERT·SC's forecasts came on the last turn before the derailment. Turning to BERT·SC+, we see that dynamic training has shifted density $H <= 3$ towards $H > 3$. The last-minute forecasts for BERT·SC+ model come at a rate of only 7.5%. FGCN-fConvD and FGCN-fConvD+ use BERT·SC in its sequential textual component and combines it with a graph component. For FGCN-fConvD density was shifted from $H <= 3$ towards $H > 3$ even though it was trained statically, indicating that the result was due to the graph modeling. The dynamically trained FGCN-fConvD+ outperforms all of the models by shifting even more density towards $H > 3$ and a last minute forecast at a rate of only 7%.

5.6 Summary

Unlike previous models which conversation derailment were based on simpler sequence models, FGCN-fConvD is built on a graph convolutional neural network model and is able to more fully capture the dynamics of multi-party dialogues, including user relationships and public

perception of conversation statements. FGCN-fConvD performed significantly better than state-of-the-art sequence based models on two widely used benchmark datasets, CGA and CMV.

The FGCN-fConvD model architecture incorporates an extensible architecture with layers of homogeneous graphs where additional data can be easily included. Results show that extending the model with more data results in a better performing model. This architecture also enables the study of the impact of the different types of data. Results on GCA and CMV show that common sense knowledge is the most influential of the data used on the model performance. Common sense knowledge features encode complex characteristics, the dynamics of multi-party dialogue, context propagation and mood shifts in conversations. This encoding may be the reason for the impact of common sense knowledge on the knowledge aware FGCN-fConvD model performance.

The FGCN-fConvD extensible architecture also supports varying the classifier and understanding the impact of simple and more complex classifiers on performance. A transformer classifier (E, ED) is more effective than the simple fully connected network classifier. However, the encoder-decoder Transformer classifier (ED) performance is similar to the perfomance of the encoder only and only slightly better than the perfomance of the Transformer classifier (E). This indicates that the complexity of ED does not always provide benefits over the Transformer classifier (E) and if so, only slightly. This is as expected since the decoder here is limited by the simplicity of the underlying problem when compared to more complex decoding tasks.

When varying the text embedding populating the nodes in the graph neural network the results show that using SLM embeddings (GPT and Bert) outperform the use of LLM embeddings for the problem of the derailment foresting. The results also show that fine tuning before embedding extraction results in better performance in the FGCN-fConvD models.

When including an estimated last utterance into the data input to the FGCN-fConvD

model, the results show that the variant without a LLM generated last utterance outperforms all variants with a generated last utterance regardless of the LLM used to generate the last utterance. A possible reason for LLMs under performing is that LLMs are sensitive to prompt engineering since they are more generalized models. The prompt engineering used to obtain the results in this work are straightforward. More sophisticated prompts that incorporate more context instruction and/or RAG could produce better results.

Conversation derailment frequently impacts our online social interactions. The ability to accurately predict derailment has the potential to enhance the effectiveness of moderation and thus protect individuals who are vulnerable to emotional abuse or harm and improve the overall quality of online interactions.

Chapter 6

Summary and future work

Emotion is intrinsic to humans, and consequently emotion understanding is a key part of human-like artificial intelligence (AI). Unlike simple query-response interactions, in many systems interactions between the human and the machine take the form of a conversation, in which information and emotional state is communicated over multiple interactions between the two parties. Humans are exceptionally gifted at recognizing the emotional state presented by individuals with whom they are communicating, and adapting their conversational interactions in response to this. Psychological studies show that humans create mental models of emotion transitions and can predict others' emotions up to two transitions into the future with an above-chance accuracy [127]. Enabling a robot/machine to respond not only to the text but also the emotions in the text requires enabling a robot/machine with the ability to infer sentiment from an agent's actions. The feasibility of computational models for predicting emotion trajectories in conversations have been barely explored.

One natural application of such a predictive model is in the area of empathetic response generation where existing strategies either mimic previous emotion, require pre-determined emotion signals or jointly model emotion prediction and response generation. Although these studies show the possibility of generating a response capable of conveying an emotion in

exsisting systems, the approach is limited in that the emotion of the response is determined manually by the user or through training examples provide by the developer. The work here provides a first step towards building machine learning models of emotion recognition within generative conversation. Another interesting application lies in the area of pre-emptive toxicity detection or the problem of detecting early indicators of antisocial discourse, which has become a pertinent research topic. Interactive applications can benefit from a range of different conversation emotional classification tasks, including: emotion recognition, future/next emotion prediction, and forecasting emotional derailment. These inter-related topics are addressed in this work. The task of emotion recognition has been researched extensively. The more challenging task, but desirable in many applications, is predicting the emotion trajectory of a conversation before the actual (future) utterances become available to the model.

Persistence and *contagion* are two basic characteristics of emotion in conversations [102, 77]. The public up/down vote on utterances in online conversations is an example of such information. Our results show that a conversation model that incorporates such information when modeling the relationship and emotional dynamics between speakersis well studied to account for these characteristics and improve on the models' ability to infer emotions. Previous studies on conversation emotion predication and derailment forecasting (e.g., [145, 85, 81]) neglect features that relate to these two characteristics such as *recency* and *self-dependency* that improve predication performance. Other than *recency* and *self-dependency*, studies have shown that conversation classification architectures that incorporate design decisions to track emotion shifts which directly relate to the two emotion propagation characteristics *persistence* and *contagion* result in improved performance [77, 135]. [114] studied the relationship between structure and toxicity in conversations on Twitter at individual, dyad, and group level, and found that social relationships among users influence their behaviors. While these recent works stress the importance of user characteristics in conversation modeling,

models that incorporate such signals for the task of conversation forecasting remain largely unexplored. This thesis shows that extending on the classification gains of sequential models through network formation in GCN models to model self-dependency (persistence), other-depencdecny(contagion), and recency improves conversation forecasting.

Building on these observations, this work describes FGCN. FGCN is a graph-based conversation forecasting model. The results in this thesis show that unlike previous models which were based on simpler sequence models, FGCN which is built on a graph convolutional neural network, is able to more fully capture the dynamics of multi-party dialogue, including user relationships and public perception of conversation statements. FGCN performs significantly better than state-of-the-art sequence based models on two widely used benchmark datasets, CGA and CMV for the forecasting conversation problem. FGCN exceeded the second best performing model by 1.4% on CMV and 1.0% on GCA data using the only textual data using static training. Adding more types of data to the FGCN model provided further superior performance. Improving the results by 2.6% in CMV and 2% in CGA. Similarly, for the problem of next emotion prediction, the graph models, FGCN, and DialogInfer which use graph convolutional neural network built on top of basic sequence models performed significantly better than state-of-the-art sequence-based models on the emotion benchmark datasets. Comparing the two graph models, FGCN and DialogInfer, DialogInfer outperforms FGCN as its graph model uses an attention mechanism to infer the participant's next emotion by directly modeling the emotion shifts relating to persistence and contagiousness. The FGCN model reduces the noise in its user relational graph through graph sparsification that models recency and self-dependency. The attention mechanism used in DialogInfer has a higher effect on performance. One possible direction to improve overall performance would be to combine both of these graph optimizations into the graph portion of the data extensible FGCN model.

Models with extensible architectures that allow for multi-modal or multi-data designs

result in enhanced performance for down stream classification problems [52, 51, 64]. By incorporating various types of data into a model with a similar architecture, one can discern the individual and collective impact of these data inputs. This thesis introduces an extensible forecasting model that is layered with diverse data types, allowing for a comprehensive analysis of how each data type affects the model's predictions. By providing a range of information capsules for the classification model, this research investigates the influence of each data type both individually and in combination. The findings underscore the importance of data variation in enhancing model performance, with emotional and common sense knowledge data emerging as the most significant contributors. Specifically, The results show that for the Next Emotion Prediction problem, emotion followed by common sense knowledge data are the most influential of data on the model performance. The emotion data directly contains the emotion shifts that relate to relating to persistence and contagiousness. Common sense knowledge was the most influential type of data on performance for the forecasting derailment problem. Common sense knowledge are features that are able to encode complex characteristics, the dynamics of multi-party dialogue, context propagation and mood shifts in conversations. This suggests that models equipped with the capability to process such data can more effectively capture the nuances of complex characteristics.

These results pertain to local conversations, and a promising direction for future work is the development of user profiles by leveraging global conversation networks to derive a user's emotional profile. By analyzing the wording, trigger words, emotional tendencies, and gender-based language patterns within a user's interactions, we can construct a rich emotional profile that reflects not just isolated conversations but a broader understanding of a user's emotional dynamics over time. This profile could be enriched by considering user interactions with others who share similar demographic characteristics, psychological traits, or emotional profiles. Furthermore, integrating common-sense knowledge from both local and global histories of conversations would provide additional context, improving the accuracy

of emotional predictions. Such an approach could lead to more empathetic, personalized responses, benefiting applications such as preemptive toxicity detection, emotional derailment forecasting, and empathetic response generation. A user-centric model like this would be able to dynamically adjust to shifts in a user's emotional state, offering a more robust understanding of their emotional trajectory and facilitating more effective and adaptive communication.

FGCN's extensible architecture allows for varying the classifier and understanding the impact of simple and more complex classifiers on performance. The results show that a Transformer classifier (E, ED) is better than the simple fully connected network classifier. However, the encoder-decoder Transformer classifier (ED) performance is similar to or only slightly better than the encoder only Transformer classifier (E). This is expected since the decoder here is limited by the simplicity of the underlying problem when compared to more complex decoding tasks.

The extensible nature of the model's architecture facilitates the inclusion of impactful data types and architectural parts, thereby enriching the model's understanding and interpretation of intricate conversational elements. This approach not only improves performance but also provides a framework for future research to further refine and expand upon the model's capabilities, potentially leading to more nuanced and human-like interactions in artificial intelligence systems.

The problems of emotion prediction and emotional derailment forecasting involve predicting the future trajectory of a conversation. Large language models (LLM), in addition to being classification models, are generative models that can be used to generate next most likely utterances given a conversation. Standard practice in the literature assumes the text of the upcoming utterance in a conversation is omitted. That is, prediction takes place without knowledge of the related utterance. An approach that has not yet been fully investigated in the literature is to generate a prediction of this utterance using a LLM and then use this

additional information to inform these tasks. This thesis explores using LLM's generated utterance information for emotion prediction tasks, and explores the use of curriculum based learning within the classification model. This thesis shows that including an estimated last utterance in the FGCN model's data input seems to have an interesting but negative effect on performance. The variant that does not include a LLM-generated last utterance outperforms those that do. This suggests that the generation capabilities of LLMs, while powerful, may introduce noise or irrelevant information that detracts from the model's forecasting accuracy. LLMs are indeed sensitive to prompt engineering; their generalized nature means they require careful crafting of prompts to produce the desired output. The straightforward prompt engineering employed in this study may not have leveraged the full potential of LLMs. Sophisticated prompts that incorporate more contextual instructions or utilize techniques like Retrieval-Augmented Generation (RAG) could potentially yield better results. When including an estimated last utterance into the data input to the FGCN model, the results show that the variant without a LLM generated last utterance outperforms all variants with a generated last utterance regardless of the LLM used to generate the last utterance. One possible reason for LLMs under performing is that LLMs are sensitive to prompt engineering since they are more generalized models. The prompt engineering used to obtain the results in this work are straight forward. More sophisticated prompts that incorporate more context instruction and/or RAG could produce better results.

This thesis also explored the use of a variation of text embeddings in the forcaesting model. It appears that smaller language models (SLMs) like GPT and BERT, when used to create embeddings for populating the nodes in a GNN, have shown superior performance over larger language models (LLMs). This could be due to the more focused training and data representation capabilities of SLMs, which may align better with the specific features of derailment forecasting tasks. Fine-tuning these models before embedding extraction further enhances their performance, likely because it allows the model to adjust more closely to the

nuances of the task-specific data.When varying the text embedding populating the nodes in the graph neural network the results show that using SLM embeddings (GPT and Bert) outperform the use of LLM embeddings for the problem of the derailment foresting. The results also show that fine tuning before embedding extraction results in better performance in the FGCN models.

Chapter 7

Limitations

Despite the significant advancements demonstrated by the FGCN model in forecasting conversation derailment and emotion prediction, several limitations remain that impact its practical applicability. The nature of conversational AI requires robust adaptability to dynamic and unpredictable interactions, yet certain constraints in the current approach hinder its effectiveness in specific scenarios. These limitations highlight areas where future research could enhance the model's flexibility, efficiency, and generalizability.

One key challenge arises from the reliance on structured graph formation. While GNN-based models like FGCN excel at capturing relational dependencies and emotional propagation, they require a minimum number of utterances to establish meaningful graph representations. This requirement presents a challenge in early-stage derailment detection, where critical intervention is often needed before sufficient conversational history is available. Similarly, the integration of LLMs introduces potential improvements in contextual understanding, but their effectiveness is constrained by the sensitivity of prompt engineering and the chosen method of incorporating generated information into the model. Additionally, the reliance on multiple data sources enhances predictive accuracy but also adds complexity, which may not always be feasible in real-world applications where lightweight and efficient models are

preferred.

These constraints underscore the need for hybrid models that incorporate sequential processing for early-stage prediction, more adaptive LLM utilization techniques, and optimized data dependency strategies. Addressing these limitations can drive further innovation in conversational AI, enabling models to perform more effectively across diverse and challenging conversational settings.

7.1 Dependence on graph structure formation

A fundamental limitation of GNNs, including FGCN, is their reliance on a well-formed graph structure to make accurate predictions. The model requires at least three utterances to construct a meaningful conversational graph, which is necessary for capturing relational dependencies and emotional propagation. However, in practical applications, particularly in early derailment detection, the problematic utterance may appear within the first or second turn of the conversation before a sufficient graph structure has been formed. This limitation means that FGCN may struggle to provide accurate predictions in very short conversations or in early-stage derailment scenarios. To address this, an adaptive model that integrates the advantages of sequential models capable of making inferences from a single utterance could complement the GNN-based approach, enabling more effective early-stage prediction.

7.2 Sensitivity to prompt engineering in LLM integration

LLMs are known to be highly sensitive to prompt engineering, where even small modifications to input prompts can lead to significantly different outputs. In this study, only a zero-shot prompt was employed, meaning that no additional contextual information was provided to guide the model's responses. While this approach allows for a more general evaluation of

LLM capabilities, it does not leverage the full potential of these models. More sophisticated prompt engineering techniques, such as few-shot learning, instruction tuning, or integrating RAG, could enhance the utility of LLM-generated features in conversation forecasting tasks. Future work should explore how structured prompting can refine LLM outputs, ensuring more contextually relevant and reliable embeddings for integration into GNN-based models.

7.3 Alternative approaches for LLM utilization

Rather than directly integrating the LLM as a RAG mechanism, this study opted to extract embeddings from LLM-generated outputs and incorporate them into the FGCN model. While embedding extraction provides useful contextual representations, it may not fully exploit the reasoning and generative strengths of LLMs. Using an LLM in an adaptive RAG framework, where the model retrieves relevant conversational context and dynamically generates complementary utterances, could offer a more sophisticated approach to improving derailment forecasting and emotion prediction. Exploring this alternative approach could allow the model to better contextualize conversations and refine its predictions.

7.4 Increased complexity due to additional data requirements

The FGCN model benefits from the inclusion of multiple data sources, such as textual data, emotion embeddings, and common-sense knowledge. However, reliance on additional data sources introduces increased complexity, both in terms of computational requirements and data availability. In real-world applications, access to rich, high-quality annotated data may be limited, and deploying models with extensive data dependencies may not always be feasible. In contrast, lightweight models that maintain high performance with minimal data

input are often preferred in practical settings. Future research should investigate methods for reducing data dependency, such as self-supervised learning techniques, transfer learning, or adaptive feature selection, to balance performance improvements with model efficiency.

7.5 Implications for model generalization

While the FGCN model demonstrates strong performance on benchmark datasets, its generalization to unseen conversational domains or out-of-distribution dialogues remains an open question. Many existing studies, including this one, focus on well-curated datasets that may not fully capture the diversity of real-world conversational settings. Additionally, the influence of speaker intent, sarcasm, and implicit emotional cues, often difficult to model, could impact the robustness of emotion prediction and derailment forecasting. Future extensions should consider domain adaptation techniques and real-world deployment evaluations to ensure that the model remains effective across diverse conversational contexts.

Appendix A

Conversational emotional trajectory datasets

This work relies on a number of publicly available emotional conversation datasets. These datasets consists of a number of conversations each including a number of utterances. Typically, each utterance is annotated by humans with a single emotion label from a predefined label set. Examples datasets include IEMOCAP [19], Emotionlines [55], MELD [103], DailyDialog [79], EmoContext [25] and SEMAINE [90]. Table A.1 provides a detailed comparison of the number of dialogues and utterances for the training, validation and testing sets of these datasets.

IEMOCAP, MELD and SEMAINE are multimodal. These datasets contain acoustic, visual and textual information. The remaining two, DailyDialog and EmoContext, contain textual data only. All datasets but SEMAINE contain categorical emotion labels for the utterences. In contrast, each utterance of SEMAINE dataset is annotated with four real valued affective attributes: valence ([-1, 1]), arousal ([-1, 1]), expectancy ([-1, 1]) and power ($[0,\infty]$)). Table A.2 shows the emotion label distribution of the data-sets containing categorical emotion lables. For theEmoContext dataset, only the last utterance of each dialogue is assigned an emotion label, this lack of annotated labels reduces its use in the literature.

For emotional derailment there exist two widely used datasets. These are the Conversations

Dataset	# dialogue			# utterances		
	train	val	test	train	val	test
IEMOCAP	120		31	5810		1623
SEMAINE	63		32	4368		1430
EmotionLines	720	80	200	10561	1178	2764
MELD	1039	114	280	9989	1109	2610
DailyDialog	11118	1000	100	87832	7912	7863
EmoContext	30159	2754	5508	90477	8262	16524

Table A.1: Number of dialouges and utterences in commonly used conversational datasets.

Gone Awry (CGA) [141] and Reddit ChangeMyView (CMV) [23] datasets. Properties of these two datasets are summarized in Table A.3.

The CGA dataset [141] was extracted from Wikipedia Talk Page conversations. The conversations were sampled from WikiConv [57] based on an automatic measure of toxicity that ranges from 0 (not toxic) to 1 (is toxic). A conversation is extracted as a sample of derailment if the N^{th} comment in the conversation has a toxicity score higher than 0.6 and all the preceding comments have a score lower than 0.4. Conversations having all comments with a toxicity score below 0.4 are identified as samples of non-derailment. This set of conversations is further filtered through manual annotation to determine whether after an initial civil exchange a personal attack occurs from one user towards another. The conversations include the turn with the personal attack. This means all $N-1$ turns in a conversation are civil and the Nth one is either civil or contains a personal attack. The dataset also contains additional information about each comment in the conversation such as an anonymous id of the user posting the comment and the ID of the parent comment that this comment was a reply to.

The CMV dataset [23] was extracted from Reddit conversations held under the Change-MyView subreddit. Conversations were sampled as derailed if there was a deletion of a turn

Label	DailyDialog	MELD	EmotionLines	IEMOCAP	EmoContext
Neutral	85572	6436	6530	1708	-
Happiness/Joy	12885	2308	1710	648	4669
Surprise	1823	1636	1658	-	-
Sadness	1150	1002	498	1084	5838
Anger	1022	1607	772	1103	5954
Disgust	353	361	338	-	-
Fear	74	358	255	-	-
Frustrated	-	-	-	1849	-
Excited	-	-	-	1041	-
Other	-	-	-	-	21960

Table A.2: Label distribution statistics in different conversation datasets.

Dataset	**Input**			**Train**	**Val**	**Test**
	t	u	s			
CGA	✓	✓	✓	2508	840	840
CMV	✓	✓	✗	4106	1368	1368

Table A.3: Properties of the datasets. t denotes text input, u denotes user ID input and s denotes public perception score input. All splits are balanced between the two classes.

by the platform moderators. This could have been performed under Reddit's Rule: "Don't be rude or hostile to other users [31]." Unlike CGA, there is no control to ensure that all preceding comments to the last one are civil, resulting in some noise in the data. The dataset also contains additional information about each comment in the conversation such as an anonymous id of the user posting the comment, the ID of the parent comment that this comment was a reply to, and a votes score (i.e., the number of up-votes minus the number of down-votes). These votes come from the public reacting to these comments through an up vote and down vote using the social media platform interface.

Datasets that encode day to day or common knowledge are known as common knowledge

include ConceptNet [121], SenticNet [20], ATOMIC [113] and Event2mind [106]. ConceptNet is a knowledge graph that connects words and phrases of natural language with labeled edges. Its knowledge is collected from many sources that include expert-created resources, crowd-sourcing, and games with a purpose. ConceptNet is designed to represent the general knowledge involved in understanding language, improving natural language applications by allowing the application to better understand the meanings behind the words people use. When ConceptNet is combined with word embeddings acquired from distributional semantics (such as word2vec [92]), it provides applications with understanding that they would not acquire from distributional semantics alone.

SenticNet [20] is a publicly available semantic and affective resource for concept-level sentiment analysis. Rather than using graph-mining and dimensionality-reduction techniques, SenticNet makes use of "energy flows" to connect various parts of extended common and common-sense knowledge representations to one another. SenticNet models nuanced semantics and sentics (that is, the conceptual and affective information associated with multi-word natural language expressions), representing information with a symbolic opacity of an intermediate nature between that of neural networks and typical symbolic systems.

ATOMIC [113] is an atlas of everyday commonsense reasoning, organized through 877k textual descriptions of inferential knowledge. Compared to existing resources that center around taxonomic knowledge, ATOMIC focuses on inferential knowledge organized as typed if-then relations with variables (e.g., "if X pays Y a compliment, then Y will likely return the compliment"). ATOMIC proposes nine if-then relation types to distinguish causes vs. effects, agents vs. themes, voluntary vs. involuntary events, and actions vs. mental states.

Event2mind [106] is a crowdsourced corpus of 25,000 event phrases covering a diverse range of everyday events and situations. It introduce a new commonsense inference task: given an event described in a short free-form text ("X drinks coffee in the morning"), a system reasons about the likely intents ("X wants to stay awake") and reactions ("X feels alert") of

the event's participants.

Bibliography

[1] Ameeta Agrawal, Aijun An, and Manos Papagelis. "Learning Emotion-enriched Word Representations". In: *Proceedings of the 27th International Conference on Computational Linguistics*. Santa Fe, New Mexico, USA: Association for Computational Linguistics, Aug. 2018, pp. 950–961. URL: https://www.aclweb.org/anthology/C18-1081.

[2] Ameeta Agrawal, Aijun An, and Manos Papagelis. "Leveraging Transitions of Emotions for Sarcasm Detection". In: *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR '20. Virtual Event, China: Association for Computing Machinery, 2020, pp. 1505–1508. ISBN: 9781450380164. DOI: 10.1145/3397271.3401183. URL: https://doi.org/10.1145/3397271.3401183.

[3] 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*. Vancouver, British Columbia, Canada: Association for Computational Linguistics, Oct. 2005, pp. 579–586. URL: https://aclanthology.org/H05-1073.

[4] Enas AlTarawneh et al. "An Infrastructure for Studying the Role of Sentiment in Human-Robot Interaction". In: *Pattern Recognition, Computer Vision, and Image*

Processing. ICPR 2022 International Workshops and Challenges. Ed. by Jean-Jacques Rousseau and Bill Kapralos. Cham: Springer Nature Switzerland, 2023, pp. 89–105. ISBN: 978-3-031-37745-7.

[5] Enas Altarawneh et al. "Conversation Derailment Forecasting with Graph Convolutional Networks". In: *The 7th Workshop on Online Abuse and Harms (WOAH)*. Toronto, Canada: Association for Computational Linguistics, July 2023, pp. 160–169. DOI: 10.18653/v1/2023.woah-1.16. URL: https://aclanthology.org/2023.woah-1.16.

[6] Enas Altarawneh et al. *Knowledge-Aware Conversation Derailment Forecasting Using Graph Convolutional Networks*. 2024. arXiv: 2408.13440 [cs.CL]. URL: https://arxiv.org/abs/2408.13440.

[7] Enas Altarawneh et al. *Predicting Evoked Emotions in Conversations*. 2023. arXiv: 2401.00383 [cs.CL]. URL: https://arxiv.org/abs/2401.00383.

[8] Julia Annas. "Ethics in Stoic Philosophy". In: *Phronesis* 52.1 (2007), pp. 58–87. (Visited on 04/16/2023).

[9] Nabiha Asghar et al. "Affective neural response generation". In: *European Conference on Information Retrieval*. Springer. Grenoble, France, 2018, pp. 154–166.

[10] Nabiha Asghar et al. *Generating Emotionally Aligned Responses in Dialogues using Affect Control Theory*. 2020. arXiv: 2003.03645 [cs.CL].

[11] Nastaran Babanejad et al. "A Comprehensive Analysis of Preprocessing for Word Representation Learning in Affective Tasks". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Ed. by Dan Jurafsky et al. Online: Association for Computational Linguistics, July 2020, pp. 5799–5810. DOI:

10.18653/v1/2020.acl-main.514. URL: https://aclanthology.org/2020.acl-main.514/.

[12] Nastaran Babanejad et al. "Leveraging Emotion Features in News Recommendations". In: *Proceedings of the 7th International Workshop on News Recommendation and Analytics (INRA)*. 2019.

[13] Nastaran Babanejad et al. "The Role of Preprocessing for Word Representation Learning in Affective Tasks". In: *IEEE Transactions on Affective Computing* 15.1 (2024), pp. 254–272. DOI: 10.1109/TAFFC.2023.3270115.

[14] *Baidu*. https://www.baidu.com/. Accessed: 2024-08-10.

[15] Y. Bengio et al. "Curriculum learning". In: vol. 60. June 2009, p. 6. DOI: 10.1145/1553374.1553380.

[16] *Bing*. https://www.bing.com/. Accessed: 2024-08-10.

[17] Antoine Bosselut et al. "COMET: Commonsense Transformers for Automatic Knowledge Graph Construction". In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, July 2019, pp. 4762–4779.

[18] Éloi Brassard-Gourdeau and Richard Khoury. "Using Sentiment Information for Preemptive Detection of Harmful Comments in Online Conversations". In: *Canadian Artificial Intelligence Association (CAIAC)*. Vancouver, Canada, June 2021.

[19] Carlos Busso et al. "IEMOCAP: Interactive emotional dyadic motion capture database". In: *Language Resources and Evaluation* 42 (Dec. 2008), pp. 335–359.

[20] Erik Cambria, Daniel Olsher, and Dheeraj Rajagopal. "SenticNet 3: A Common and Common-Sense Knowledge Base for Cognition-Driven Sentiment Analysis". In:

Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence. AAAI'14. Québec City, Québec, Canada: AAAI Press, 2014, pp. 1515–1521.

[21] Loredana Caruccio et al. "Claude 2.0 large language model: Tackling a real-world classification problem with a new iterative prompt engineering approach". In: *Intelligent Systems with Applications* 21 (2024), p. 200336. ISSN: 2667-3053. DOI: https://doi.org/10.1016/j.iswa.2024.200336. URL: https://www.sciencedirect.com/science/article/pii/S2667305324000127.

[22] Deeksha Chandola et al. "SERC-GCN: Speech Emotion Recognition In Conversation Using Graph Convolutional Networks." In: *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*. Seoul, Korea, 2024, pp. 76–80.

[23] Jonathan Chang and Cristian Danescu-Niculescu-Mizil. "Trouble on the Horizon: Forecasting the Derailment of Online Conversations as they Develop". In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 4743–4754.

[24] Jonathan P. Chang, Charlotte Schluger, and Cristian Danescu-Niculescu-Mizil. "Thread With Caution: Proactively Helping Users Assess and Deescalate Tension in Their Online Discussions". In: *Association for Computing Machinery*. Vol. 6. CSCW2. New York, NY, USA, Nov. 2022.

[25] Ankush Chatterjee et al. "SemEval-2019 Task 3: EmoContext Contextual Emotion Detection in Text". In: *Proceedings of the 13th International Workshop on Semantic Evaluation*. Minneapolis, Minnesota, USA: Association for Computational Linguistics, June 2019, pp. 39–48.

[26] Zhihong Chen et al. *Phoenix: Democratizing ChatGPT across Languages*. 2023. arXiv: 2304.10453 [cs.CL].

[27] Zhongxia Chen et al. "Neural response generation with relevant emotions for short text conversation". In: *CCF International Conference on Natural Language Processing and Chinese Computing*. Springer. Gansu, China, 2019, pp. 117–129.

[28] Marcus Tullius Cicero and Margaret R. Graver. *Cicero on the Emotions: Tusculan Disputations 3 and 4*. Chicago, USA: University of Chicago Press, Feb. 2002. ISBN: 9780226305776.

[29] *Co Pilot*. https://www.bing.com/chat. Accessed: 2024-08-10.

[30] Pierre Colombo et al. "Affect-Driven Dialog Generation". 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)*. Minneapolis, Minnesota: Association for Computational Linguistics, June 2019, pp. 3734–3743.

[31] *Content policy (no date) reddit*. https://www.redditinc.com/policies/content-policy. ccessed: 11 May 2024.

[32] Daniel Cordaro et al. "The voice conveys emotion in ten globalized cultures and one remote village in Bhutan." In: *Emotion* 16 1 (2016), pp. 117–28.

[33] Thomas Davidson et al. "Automated hate speech detection and the problem of offensive language". In: *Proceedings of the International AAAI Conference on Web and Social Media*. Vol. 11. 1. Philadelphia, California, 2017.

[34] Christine De Kock and Andreas Vlachos. "I Beg to Differ: A study of constructive disagreement in online conversations". In: *Proceedings of the 16th Conference of the*

European Chapter of the Association for Computational Linguistics: Main Volume. Online: Association for Computational Linguistics, Apr. 2021, pp. 2017–2027.

[35] Jacob Devlin et al. "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)*. Minneapolis, Minnesota: Association for Computational Linguistics, June 2019, pp. 4171–4186.

[36] Yao Dou et al. "Is GPT-3 Text Indistinguishable from Human Text? Scarecrow: A Framework for Scrutinizing Machine Text". In: *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Dublin, Ireland: Association for Computational Linguistics, May 2022, pp. 7250–7274.

[37] Jamie Dow. "Aristotle's Theory of the Emotions: Emotions as Pleasures and Pains". In: *Moral Psychology and Human Action in Aristotle*. Oxford University Press, Feb. 2011. ISBN: 9780199546541.

[38] Thibault Formal, Benjamin Piwowarski, and Stéphane Clinchant. "SPLADE: Sparse Lexical and Expansion Model for First Stage Ranking". In: *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR '21. Virtual Event, Canada: Association for Computing Machinery, 2021, pp. 2288–2292. ISBN: 9781450380379. DOI: 10.1145/3404835.3463098. URL: https://doi.org/10.1145/3404835.3463098.

[39] Bram M Fridhandler and James R Averill. "Temporal dimensions of anger: An exploration of time and emotion". In: *Anger and aggression*. Springer, 1982, pp. 253–279.

[40] Radhika Gaonkar et al. "Modeling Label Semantics for Predicting Emotional Reactions". In: *Proceedings of the 58th Annual Meeting of the Association for Computational*

Linguistics. Online: Association for Computational Linguistics, July 2020, pp. 4687–4692.

[41] Deepanway Ghosal et al. "COSMIC: COmmonSense knowledge for eMotion Identification in Conversations". In: *Findings of the Association for Computational Linguistics: EMNLP 2020*. Online: Association for Computational Linguistics, Nov. 2020, pp. 2470–2481.

[42] Deepanway Ghosal et al. "DialogueGCN: A Graph Convolutional Neural Network for Emotion Recognition in Conversation". In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 154–164.

[43] Deepanway Ghosal et al. "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*. Online: Association for Computational Linguistics, Aug. 2021, pp. 1435–1449. DOI: 10.18653/v1/2021.findings-acl.124. URL: https://aclanthology.org/2021.findings-acl.124.

[44] *Google*. https://www.google.ca/. Accessed: 2024-08-10.

[45] Goole. *Embeddings in the Gemini API | Google AI for Developers.* Accessed 8 Oct. 2024. 2024. URL: ai.google.dev/gemini-api/docs/embeddings.

[46] Xiaodong Gu, Kang Min Yoo, and Jung-Woo Ha. "DialogBERT: Discourse-Aware Response Generation via Learning to Recover and Rank Utterances". In: 2021.

[47] Xiusen Gu, Weiran Xu, and Si Li. "Towards automated emotional conversation generation with implicit and explicit affective strategy". In: *Proceedings of the 2019*

International Symposium on Signal Processing Systems. Beijing, China, 2019, pp. 125–130.

[48] Rachel Hajar. "The air of history (Part V): Ibn Sina (Avicenna): The great physician and philosopher". In: *Heart Views* 14.4 (2013), p. 196.

[49] Felix Hamborg and Karsten Donnay. "NewsMTSC: A Dataset for (Multi-)Target-dependent Sentiment Classification in Political News Articles". In: *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*. Online: Association for Computational Linguistics, Apr. 2021, pp. 1663–1675.

[50] Takayuki Hasegawa et al. "Predicting and Eliciting Addressee's Emotion in Online Dialogue". In: *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Sofia, Bulgaria: Association for Computational Linguistics, Aug. 2013, pp. 964–972.

[51] Devamanyu Hazarika et al. "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)*. New Orleans, Louisiana: Association for Computational Linguistics, June 2018, pp. 2122–2132.

[52] Devamanyu Hazarika et al. "ICON: Interactive Conversational Memory Network for Multimodal Emotion Detection". In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Brussels, Belgium: Association for Computational Linguistics, Oct. 2018, pp. 2594–2604.

[53] Han He, Liyan Xu, and Jinho D. Choi. *ELIT: Emory Language and Information Toolkit*. 2021. arXiv: 2109.03903 [cs.CL].

[54] Benjamin Horne, Sibel Adali, and Sujoy Sikdar. "Identifying the social signals that drive online discussions: A case study of Reddit communities". In: *2017 26th International Conference on Computer Communication and Networks (ICCCN)*. IEEE. Vancouver, Canada, 2017, pp. 1–9.

[55] Chao-Chun Hsu et al. "EmotionLines: An Emotion Corpus of Multi-Party Conversations". In: *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*. Miyazaki, Japan: European Language Resources Association (ELRA), May 2018.

[56] Dou Hu, Lingwei Wei, and Xiaoyong Huai. "DialogueCRN: Contextual Reasoning Networks for Emotion Recognition in Conversations". In: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Online: Association for Computational Linguistics, Aug. 2021, pp. 7042–7052.

[57] Yiqing Hua et al. "WikiConv: A Corpus of the Complete Conversational History of a Large Online Collaborative Community". In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Brussels, Belgium: Association for Computational Linguistics, Oct. 2018, pp. 2818–2823.

[58] Chenyang Huang et al. "Automatic Dialogue Generation with Expressed Emotions". In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*. New Orleans, Louisiana: Association for Computational Linguistics, June 2018, pp. 49–54.

[59] Yuyun Huang and Jinhua Du. "Self-Attention Enhanced CNNs and Collaborative Curriculum Learning for Distantly Supervised Relation Extraction". In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th*

International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 389–398. DOI: 10.18653/v1/D19-1037. URL: https://aclanthology.org/D19-1037.

[60] Taichi Ishiwatari et al. "Relation-aware Graph Attention Networks with Relational Position Encodings for Emotion Recognition in Conversations". In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational Linguistics, Nov. 2020, pp. 7360–7370.

[61] Eric Jang, Shixiang Gu, and Ben Poole. "Categorical Reparameterization with Gumbel-Softmax". In: *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. URL: https://openreview.net/forum?id=rkE3y85ee.

[62] Piotr Janiszewski, Mateusz Lango, and Jerzy Stefanowski. "Time Aspect in Making an Actionable Prediction of a Conversation Breakdown". In: *Machine Learning and Knowledge Discovery in Databases. Applied Data Science Track*. Cham, Switerland: Springer International Publishing, 2021, pp. 351–364.

[63] Wenxiang Jiao et al. "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)*. Minneapolis, Minnesota: Association for Computational Linguistics, June 2019, pp. 397–406.

[64] Wenxiang Jiao et al. "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)*. Minneapolis, Minnesota: Association

for Computational Linguistics, June 2019, pp. 397–406. DOI: 10.18653/v1/N19-1037. URL: https://aclanthology.org/N19-1037.

[65] Vladimir Karpukhin et al. "Dense Passage Retrieval for Open-Domain Question Answering". In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational Linguistics, Nov. 2020, pp. 6769–6781. DOI: 10.18653/v1/2020.emnlp-main.550. URL: https://aclanthology.org/2020.emnlp-main.550.

[66] Timo Kaufmann et al. *A Survey of Reinforcement Learning from Human Feedback*. 2024. arXiv: 2312.14925 [cs.LG]. URL: https://arxiv.org/abs/2312.14925.

[67] Dacher Keltner. "Ekman, emotional expression, and the art of empirical epiphany". In: *Journal of Research in Personality* 38.1 (2004), pp. 37–44.

[68] Yova Kementchedjhieva and Anders Søgaard. "Dynamic Forecasting of Conversation Derailment". In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Punta Cana, Dominican Republic: Association for Computational Linguistics, Nov. 2021, pp. 7915–7919.

[69] Sunghwan Mac Kim, Alessandro Valitutti, and Rafael A. Calvo. "Evaluation of Unsupervised Emotion Models to Textual Affect Recognition". In: *Proceedings of the NAACL HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text*. Los Angeles, CA: Association for Computational Linguistics, June 2010, pp. 62–70. URL: https://aclanthology.org/W10-0208.

[70] Yoon Kim. "Convolutional Neural Networks for Sentence Classification". In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar: Association for Computational Linguistics, Oct. 2014, pp. 1746–1751. DOI: 10.3115/v1/D14-1181. URL: https://aclanthology.org/D14-1181.

[71] Jean Knox. "The Archaeology of Mind: Neuroevolutionary Origins of Human Emotions by Jaak Panksepp and Lucy Biven". In: *Journal of Analytical Psychology* 58.3 (2013), pp. 439–441.

[72] Bernhard Kratzwald et al. "Deep learning for affective computing: Text-based emotion recognition in decision support". In: *Decision Support Systems* 115 (2018), pp. 24–35. ISSN: 0167-9236.

[73] Stanley Kubrick. *2001: A Space Odyssey*. Film. Directed by Stanley Kubrick. 1968.

[74] Akshi Kumar and Teeja Mary Sebastian. "Sentiment Analysis: A Perspective on its Past, Present and Future". In: *International Journal of Intelligent Systems and Applications* 4 (2012), pp. 1–14.

[75] Shanglin Lei et al. *InstructERC: Reforming Emotion Recognition in Conversation with a Retrieval Multi-task LLMs Framework*. 2024. arXiv: 2309.11911 [cs.CL].

[76] Mike Lewis et al. "BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, July 2020, pp. 7871–7880. DOI: 10.18653/v1/2020.acl-main.703. URL: https://aclanthology.org/2020.acl-main.703.

[77] Dayu Li et al. "Emotion Inference in Multi-Turn Conversations with Addressee-Aware Module and Ensemble Strategy". In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Punta Cana, Dominican Republic: Association for Computational Linguistics, Nov. 2021, pp. 3935–3941.

[78] Dayu Li et al. "Enhancing emotion inference in conversations with commonsense knowledge". In: *Knowledge-Based Systems* 232 (2021), p. 107449. ISSN: 0950-7051.

[79] Yanran Li et al. "DailyDialog: A Manually Labelled Multi-turn Dialogue Dataset". In: *Proceedings of the Eighth International Joint Conference on Natural Language Processing*. Taipei, Taiwan: Asian Federation of Natural Language Processing, Nov. 2017, pp. 986–995.

[80] Zaijing Li et al. "EmoCaps: Emotion Capsule based Model for Conversational Emotion Recognition". In: *Findings of the Association for Computational Linguistics: ACL 2022*. Dublin, Ireland: Association for Computational Linguistics, May 2022, pp. 1610–1618.

[81] Yunlong Liang et al. "Emotional conversation generation with heterogeneous graph neural network". In: *Artificial Intelligence* 308 (2022), p. 103714. ISSN: 0004-3702.

[82] Cao Liu et al. "Curriculum Learning for Natural Answer Generation". In: *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*. Stockholm, Sweden: International Joint Conferences on Artificial Intelligence Organization, July 2018, pp. 4223–4229. DOI: 10.24963/ijcai.2018/587. URL: https://doi.org/10.24963/ijcai.2018/587.

[83] Jiawang Liu et al. "Graph based emotion recognition with attention pooling for variable-length utterances". In: *Neurocomputing* 496 (2022), pp. 46–55. ISSN: 0925-2312. DOI: https://doi.org/10.1016/j.neucom.2022.05.007. URL: https://www.sciencedirect.com/science/article/pii/S0925231222005380.

[84] Nurul Lubis et al. "Eliciting positive emotion through affect-sensitive dialogue response generation: A neural network approach". In: *Thirty-Second AAAI Conference on Artificial Intelligence*. Louisiana, USA, 2018.

[85] Nurul Lubis et al. "Positive Emotion Elicitation in Chat-Based Dialogue Systems". In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 27 (2019), pp. 866–877.

[86] Navonil Majumder et al. “DialogueRNN: An Attentive RNN for Emotion Detection in Conversations”. In: *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence and Thirty-First Innovative Applications of Artificial Intelligence Conference and Ninth AAAI Symposium on Educational Advances in Artificial Intelligence*. AAAI'19/IAAI'19/EAAI'19. Honolulu, Hawaii, USA: AAAI Press, 2019. ISBN: 978-1-57735-809-1.

[87] Mika Mäntylä, Daniel Graziotin, and Miikka Kuutila. “The evolution of sentiment analysis—A review of research topics, venues, and top cited papers”. In: *Computer Science Review* 27 (2018), pp. 16–32. ISSN: 1574-0137.

[88] Shuiyang Mao, PC Ching, and Tan Lee. “Deep Learning of Segment-Level Feature Representation with Multiple Instance Learning for Utterance-Level Speech Emotion Recognition.” In: *Interspeech*. Graz, Austria, 2019, pp. 1686–1690.

[89] Ricardo Martins et al. “Hate speech classification in social media using emotional analysis”. In: *2018 7th Brazilian Conference on Intelligent Systems (BRACIS)*. IEEE. Sao Paulo, 2018, pp. 61–66.

[90] Gary McKeown et al. “The SEMAINE corpus of emotionally coloured character interactions”. In: *2010 IEEE International Conference on Multimedia and Expo*. Singapore, Singapore, 2010, pp. 1079–1084.

[91] Tomas Mikolov et al. “Distributed representations of words and phrases and their compositionality”. In: *arXiv preprint arXiv:1310.4546* (2013).

[92] Tomas Mikolov et al. “Efficient Estimation of Word Representations in Vector Space”. In: *International Conference on Learning Representations*. Arizona, USA, 2013. URL: https://api.semanticscholar.org/CorpusID:5959482.

[93] OpenAI. *ChatGPT: OpenAI's Language Model*. https://openai.com/gpt. Version X.X. 2024.

[94] OpenAI. *GPT-4 Technical Report*. 2023. arXiv: 2303.08774 [cs.CL].

[95] Juri Opitz and Sebastian Burst. *Macro F1 and Macro F1*. 2021. arXiv: 1911.03347 [cs.LG]. URL: https://arxiv.org/abs/1911.03347.

[96] Bo Pang and Lillian Lee. "Opinion Mining and Sentiment Analysis". In: *Found. Trends Inf. Retr.* 2.1–2 (Jan. 2008), pp. 1–135. ISSN: 1554-0669.

[97] Ekman Paul. "Darwin's contributions to our understanding of emotional expressions." In: *Philosophical transactions of the Royal Society of London. Series B, Biological sciences*. 2009, pp. 3449–3451.

[98] Jeffrey Pennington, Richard Socher, and Christopher D Manning. "Glove: Global vectors for word representation". In: *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. Doha, Qatar, 2014, pp. 1532–1543.

[99] Robert Plutchik. "A psychoevolutionary theory of emotions". In: *Social Science Information* 21.4-5 (1982), pp. 529–553.

[100] Fabio Poletto et al. "Resources and benchmark corpora for hate speech detection: a systematic review". In: *Language Resources and Evaluation* (2020), pp. 1–47.

[101] Soujanya Poria et al. "Context-Dependent Sentiment Analysis in User-Generated Videos". In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Vancouver, Canada: Association for Computational Linguistics, July 2017, pp. 873–883. DOI: 10.18653/v1/P17-1081. URL: https://aclanthology.org/P17-1081.

[102] Soujanya Poria et al. "Emotion recognition in conversation: Research challenges, datasets, and recent advances". In: *IEEE Access* 7 (2019), pp. 100943–100953.

[103] Soujanya Poria et al. “MELD: A Multimodal Multi-Party Dataset for Emotion Recognition in Conversations”. In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, July 2019, pp. 527–536.

[104] Alec Radford and Karthik Narasimhan. “Improving Language Understanding by Generative Pre-Training”. In: 2018. URL: https://api.semanticscholar.org/CorpusID:49313245.

[105] Colin Raffel et al. “Exploring the limits of transfer learning with a unified text-to-text transformer”. In: *J. Mach. Learn. Res.* 21.1 (Jan. 2020). ISSN: 1532-4435.

[106] Hannah Rashkin et al. “Event2Mind: Commonsense Inference on Events, Intents, and Reactions”. In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Melbourne, Australia: Association for Computational Linguistics, July 2018, pp. 463–473.

[107] Hannah Rashkin et al. “Towards Empathetic Open-domain Conversation Models: A New Benchmark and Dataset”. In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, July 2019, pp. 5370–5381.

[108] Nils Reimers and Iryna Gurevych. “Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks”. In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 3982–3992. DOI: 10.18653/v1/D19-1410. URL: https://aclanthology.org/D19-1410.

[109] Fuji Ren and Yanwei Bao. "A review on human-computer interaction and intelligent robots". In: *International Journal of Information Technology & Decision Making* 19.01 (2020), pp. 5–47.

[110] Stephen Robertson and Hugo Zaragoza. "The Probabilistic Relevance Framework: BM25 and Beyond". In: *Found. Trends Inf. Retr.* 3.4 (Apr. 2009), pp. 333–389. ISSN: 1554-0669. DOI: 10.1561/1500000019. URL: https://doi.org/10.1561/1500000019.

[111] Nazanin Salehabadi et al. "User Engagement and the Toxicity of Tweets". In: *arXiv preprint arXiv:2211.03856* (2022).

[112] Gerard Salton, Edward A. Fox, and Harry Wu. "Extended Boolean information retrieval". In: *Commun. ACM* 26.11 (Nov. 1983), pp. 1022–1036. ISSN: 0001-0782. DOI: 10.1145/182.358466. URL: https://doi.org/10.1145/182.358466.

[113] Maarten Sap et al. "ATOMIC: An Atlas of Machine Commonsense for If-Then Reasoning". In: *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*. AAAI Press, 2019, pp. 3027–3035.

[114] Martin Saveski, Brandon Roy, and Deb Roy. "The structure of toxic conversations on Twitter". In: *Proceedings of the Web Conference 2021*. online, 2021, pp. 1086–1097.

[115] K. R. Scherer, A. Schorr, and Tom Johnstone. *Appraisal processes in emotion: theory, methods, research*. Series in affective science. USA: Oxford University Press, 2001. URL: https://centaur.reading.ac.uk/4377/.

[116] Charlotte Schluger et al. “Proactive Moderation of Online Discussions: Existing Practices and the Potential for Algorithmic Support”. In: *Proceedings of the ACM on Human-Computer Interaction*. Vol. 6. CSCW2. New York, USA: ACM New York, NY, USA, 2022, pp. 1–27.

[117] Bongrae Seok. “The Emotions in Early Chinese Philosophy”. In: *The Stanford Encyclopedia of Philosophy*. Ed. by Edward N. Zalta. Winter 2021. Metaphysics Research Lab, Stanford University, 2021.

[118] Weizhou Shen et al. “DialogXL: All-in-One XLNet for Multi-Party Conversation Emotion Recognition”. In: *Proceedings of the AAAI Conference on Artificial Intelligence* 35.15 (May 2021), pp. 13789–13797.

[119] Weizhou Shen et al. “Directed Acyclic Graph Network for Conversational Emotion Recognition”. In: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Online: Association for Computational Linguistics, Aug. 2021, pp. 1551–1560.

[120] Ronald de Sousa. In: *Noûs* 25.3 (1991), pp. 367–373. ISSN: 00294624, 14680068. URL: http://www.jstor.org/stable/2215510 (visited on 05/19/2023).

[121] Robyn Speer and Joanna Lowry-Duda. “ConceptNet at SemEval-2017 Task 2: Extending Word Embeddings with Multilingual Relational Knowledge”. In: *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*. Vancouver, Canada: Association for Computational Linguistics, Aug. 2017, pp. 85–89.

[122] Jan E. Stets. “Emotions and Sentiments”. In: *Handbook of Social Psychology*. Ed. by John Delamater. Boston, MA: Springer US, 2006, pp. 309–335. ISBN: 978-0-387-36921-1.

[123] Carlo Strapparava and Rada Mihalcea. "Annotating and Identifying Emotions in Text". In: vol. 301. July 2010, pp. 21–38. ISBN: 978-3-642-13999-4. DOI: 10.1007/978-3-642-14000-6_2.

[124] Yixuan Su et al. "Dialogue Response Selection with Hierarchical Curriculum Learning". In: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Online: Association for Computational Linguistics, Aug. 2021, pp. 1740–1751. DOI: 10.18653/v1/2021.acl-long.137. URL: https://aclanthology.org/2021.acl-long.137.

[125] Yang Sun, Nan Yu, and Guohong Fu. "A Discourse-Aware Graph Neural Network for Emotion Recognition in Multi-Party Conversation". In: *Findings of the Association for Computational Linguistics: EMNLP 2021*. Punta Cana, Dominican Republic: Association for Computational Linguistics, Nov. 2021, pp. 2949–2958.

[126] Gemini Team et al. *Gemini: A Family of Highly Capable Multimodal Models*. 2024. arXiv: 2312.11805 [cs.CL]. URL: https://arxiv.org/abs/2312.11805.

[127] Mark Thornton and Diana Tamir. "Mental models accurately predict emotion transitions". In: *Proceedings of the National Academy of Sciences* 114.23 (2017), pp. 5982–5987.

[128] Hugo Touvron et al. *LLaMA: Open and Efficient Foundation Language Models*. 2023. arXiv: 2302.13971 [cs.CL].

[129] Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. "Pointer networks". In: *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2*. NIPS'15. Montreal, Canada: MIT Press, 2015, pp. 2692–2700.

[130] Xin Wang, Yudong Chen, and Wenwu Zhu. *A Survey on Curriculum Learning*. 2021. arXiv: 2010.13166 [cs.LG].

[131] Xuena Wang et al. *Emotional Intelligence of Large Language Models*. 2023. arXiv: 2307.09042 [cs.AI].

[132] Yan Wang et al. "Contextualized Emotion Recognition in Conversation as Sequence Tagging". In: *Proceedings of the 21th Annual Meeting of the Special Interest Group on Discourse and Dialogue*. 1st virtual meeting: Association for Computational Linguistics, July 2020, pp. 186–195.

[133] Michael Wiegand, Josef Ruppenhofer, and Thomas Kleinbauer. "Detection of Abusive Language: the Problem of Biased Datasets". 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)*. Minneapolis, Minnesota: Association for Computational Linguistics, June 2019, pp. 602–608.

[134] Lee Xiong et al. *Approximate Nearest Neighbor Negative Contrastive Learning for Dense Text Retrieval*. 2020. arXiv: 2007.00808 [cs.IR]. URL: https://arxiv.org/abs/2007.00808.

[135] Lin Yang et al. "Hybrid Curriculum Learning for Emotion Recognition in Conversation". In: *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022*. AAAI Press, 2022, pp. 11595–11603.

[136] Yang Yu and Xiangzhi Liu. "Research on Enterprise Text Classification Methods of BiLSTM and CNN Based on BERT". In: *Proceedings of the 2023 9th International Conference on Computing and Artificial Intelligence*. ICCAI '23. Tianjin, China:

Association for Computing Machinery, 2023, pp. 491–495. ISBN: 9781450399029. DOI: 10.1145/3594315.3594362. URL: https://doi.org/10.1145/3594315.3594362.

[137] Jiaqing Yuan and Munindar P. Singh. "Conversation Modeling to Predict Derailment". In: *Proceedings of the International AAAI Conference on Web and Social Media* 17.1 (June 2023), pp. 926–935. DOI: 10.1609/icwsm.v17i1.22200. URL: https://ojs.aaai.org/index.php/ICWSM/article/view/22200.

[138] Samira Zad and Mark Finlayson. "Systematic Evaluation of a Framework for Unsupervised Emotion Recognition for Narrative Text". In: *Proceedings of the First Joint Workshop on Narrative Understanding, Storylines, and Events*. Online: Association for Computational Linguistics, July 2020, pp. 26–37. DOI: 10.18653/v1/2020.nuse-1.4. URL: https://aclanthology.org/2020.nuse-1.4.

[139] Sayyed M. Zahiri and Jinho D. Choi. *Emotion Detection on TV Show Transcripts with Sequence-based Convolutional Neural Networks*. 2017. arXiv: 1708.04299 [cs.CL]. URL: https://arxiv.org/abs/1708.04299.

[140] Dong Zhang et al. "Modeling both Context- and Speaker-Sensitive Dependence for Emotion Detection in Multi-speaker Conversations". In: *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*. Macao, China: International Joint Conferences on Artificial Intelligence Organization, July 2019, pp. 5415–5421. DOI: 10.24963/ijcai.2019/752. URL: https://doi.org/10.24963/ijcai.2019/752.

[141] Justine Zhang et al. "Conversations Gone Awry: Detecting Early Signs of Conversational Failure". In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Melbourne, Australia: Association for Computational Linguistics, July 2018, pp. 1350–1361.

[142] Wenxuan Zhang et al. *Sentiment Analysis in the Era of Large Language Models: A Reality Check*. 2023. arXiv: 2305.15005 [cs.CL].

[143] Yazhou Zhang et al. *DialogueLLM: Context and Emotion Knowledge-Tuned Large Language Models for Emotion Recognition in Conversations*. 2024. arXiv: 2310.11374 [cs.CL].

[144] Yizhe Zhang et al. "DialoGPT: Large-Scale Generative Pre-training for Conversational Response Generation". In: *ACL, system demonstration*. online, 2020.

[145] Peixiang Zhong, Di Wang, and Chunyan Miao. "An Affect-Rich Neural Conversational Model with Biased Attention and Weighted Cross-Entropy Loss". In: *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence and Thirty-First Innovative Applications of Artificial Intelligence Conference and Ninth AAAI Symposium on Educational Advances in Artificial Intelligence*. AAAI'19/IAAI'19/EAAI'19. Honolulu, Hawaii, USA: AAAI Press.

[146] Peixiang Zhong, Di Wang, and Chunyan Miao. "Knowledge-Enriched Transformer for Emotion Detection in Textual Conversations". In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 165–176.

[147] Hao Zhou et al. "Emotional Chatting Machine: Emotional Conversation Generation with Internal and External Memory". In: AAAI'18/IAAI'18/EAAI'18. New Orleans, Louisiana, USA: AAAI Press, 2018.

[148] Xianda Zhou and William Yang Wang. "MojiTalk: Generating Emotional Responses at Scale". In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Melbourne, Australia: Association for Computational Linguistics, July 2018, pp. 1128–1137.

[149] Lixing Zhu et al. "Topic-Driven and Knowledge-Aware Transformer for Dialogue Emotion Detection". In: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Online: Association for Computational Linguistics, Aug. 2021, pp. 1571–1582.

[150] Liu Zhuang et al. "A Robustly Optimized BERT Pre-training Approach with Post-training". English. In: *Proceedings of the 20th Chinese National Conference on Computational Linguistics*. Ed. by Sheng Li et al. Huhhot, China: Chinese Information Processing Society of China, Aug. 2021, pp. 1218–1227. URL: https://aclanthology.org/2021.ccl-1.108.