ENHANCING TRANSFORMER ARCHITECTURES FOR DIALOGUE MODELLING THROUGH CONTEXTUAL REENCODING

GUENDALINA CALDARINI

A thesis submitted in partial fulfilment of the
requirements of the University of Sunderland
for the degree of Doctor of Philosophy

May 2025

School of Computer Science

Enhancing Transformer Architectures for Dialogue Modelling Through Contextual Reencoding

Guendalina Caldarini

PhD 2025

Updated May 2025

Table of Contents

Abstract

Declaration

Copyright

Dedication

Acknowledgement

Chapter 1: Introduction 14

1.1 Research Background Overview of Chatbot Design 15

1.1.1 Research Aim and Objectives 16

1.1.2 Research Contribution 16

1.1.3 Research Methodology 16

1.2 Thesis Outline 17

Chapter 2: Literature Review 18

2.1 Introduction 18

2.2 Research Background 19

2.3 Rule-Based approaches to Chatbots 19

2.4 Artificial Intelligence Approaches to Chatbots (Machine Learning Powered) 20

2.4.1 Information-Retrieval Chatbots 21

2.4.2 Generative Models 22

2.4.2.1 Transformer Chatbots 24

2.4.3 Multimodal Chatbots 28

2.4.4 State of the Art 30

2.4.4.1 ChatGPT 30

2.5 Advances in Chatbots and Dialogue Modelling post-ChatGpt 33

2.6 Summary 34

Chapter 3: Research Methodology 35

3.1 Introduction 35

3.2 Datasets 36

3.2.1 Data Cleaning and Preparation 40

3.2.2 Extracting Audio Embeddings 42

3.3 Multimodal Dialogue modelling 43

3.3.1 Incorporating Audio Embeddings 46

3.4 Training Procedure 48

3.4.1 Embedding Layers in Large Language Models 50
3.4.1.1 Fundamentals of Embedding Layers 50
3.4.1.2 Embedding Initialization 51
3.4.1.3 Training Dynamics 51
3.4.1.4 Semantic Similarity and Contextual Understanding 51
3.4.1.5 Dimensionality and Vector Space Properties 51
3.4.1.6 Contextualised Embeddings 51
3.4.2 Tokenization 52
3.4.2.1 TensorFlow's Tokenizer 52
3.4.2.2 TensorFlow's Subword Text Encoder 52
3.4.2.3 GloVe (6b) 52
3.4.2.4 BERT Tokenizer 53
3.5 Evaluation Metrics 54
3.5.1 BLEU 57
3.5.2 METEOR 57
3.5.3 TER 58
3.5.4 Perplexity 59
3.6 Summary 59
Chapter 4: Implementation 61
4.1 Introduction 61
4.2 Architectures 61
4.2.1 Encoder-Decoder Transformer Architecture 61
4.2.2 Reencoder 62
4.2.3 Extractor 64
4.2.4 Integrating Audio Embeddings 66
4.3 Code refactoring 67
4.3.1 Data caching and retrieval for optimization 67
4.4 Summary 68
Chapter 5: Experiments and Results 70
5.1 Introduction 70
5.2 Experimental Methodologies 70
5.3 Training Process 71
5.4 Unimodal Results 73
5.4.1 Results on the DailyDialog Dataset (Text Embeddings) 73
5.4.2 Results on the Cornell Dataset (Text Embeddings) 74
5.4.3 Results on the OpenSubtitles Dataset (Text Embeddings) 74

5.4.4 Results on the Meld Dataset (Text Embeddings) 75
5.4.5 Results on the OpenSubtitles Datasets with Training data corresponding to 1% of the entire dataset (Text Embeddings) 77
5.4.6 Results on the Meld Dataset (Audio Embeddings) 79
5.4.7 Results on the IEMOCAP dataset (Text Embeddings) 79
5.4.8 Results on the IEMOCAP dataset (Audio Embeddings) 80
5.5 Multimodal Results 81
5.5.1 Results on the MELD dataset (text and audio embeddings) 81
5.5.2 Results on the IEMOCAP Dataset (audio and text embeddings) 83
5.6 Comparing our Multimodal Results with previous Research Findings 85
5.7 Comparing Audio, Text, and Multimodal models 87
5.8 Best Performing Architecture and Embedding Layers 90
5.9 Unexpected Findings 90
5.10 Summary 91
Chapter 6: Discussion 93
6.1 Introduction 93
6.2 Architecture Comparison Performance 93
6.3 The effect of dataset quality, size, and model complexity on performance 95
6.3.1 Text only models' training on Dailydialog and IEMOCAP 95
6.3.2 Factors Contributing to Low Bleu Performance with the MELD Dataset 96
6.4 Embedding layers and model performance 96
6.5 Dataset impact on model performance 99
6.6 Effect of Tokenization on the Embedding Layer of Large Language Models 101
6.7 Interplay between Model Dimensionality, Data Size, and Task Specificity 102
6.7.1 The Dimensionality-Data-Task Landscape 103
6.8 Beyond the Correlation: Exploring the Reencoder Architecture 105
6.9 Summary 106
Chapter 7: Conclusion and Future Work 107
7.1 Conclusion 107
7.2 Contribution 108
7.3 Future Work 110
7.3.1 Leveraging Small Language Models and Focused Datasets 111
7.3.2 The relationship between model dimensionality and dataset size 113
7.3.3 Generalizability and Transferability of SLMs 115
7.3.4 The Reencoder Architecture: Scalability and Performance 115
References 117

Appendix A 138
Appendix B 149
Computational Resources 149

Word Count: 42817

List of Tables

Table 1. Table comparing the different datasets 39

Table 2. Table comparing the different tokenizers used for experimentation 54

Table 3. System evaluation on DailyDialog dataset using different embedding algorithms and performance measures. 73

Table 4. System evaluation on Cornell dataset using different embedding algorithms and performance measures 74

Table 5. System evaluation on OpenSubtitles dataset using different embedding algorithms and performance measures 74

Table 6. System evaluation on MELD dataset using text embeddings 75

Table 7. System evaluation on OpenSubtitles Dataset with Training data corresponding to 1% of the entire dataset 77

Table 8. System evaluation on MELD Dataset using audio embeddings 79

Table 9. System evaluation on IEMOCAP Dataset using text embeddings 79

Table 10. System evaluation on IEMOCAP Dataset using audio embeddings 80

Table 11. System evaluation on Meld Dataset with Audio and Text extracted using different embedding algorithms and performance measures 82

Table 12. System evaluation on IEMOCAP Dataset with Audio and Text extracted using different embedding algorithms and performance measures. 84

Table 13. Perplexity scores of Multimodal Dataset with Audio and Text extracted using different embedding algorithms and compared to Young et al., (2020) 86

List of Figures

Figure 2.1 Recurrent Neural Network 23

Figure 2.2 Using computational efficiency criteria, RNN, Convolutional Neural Network (CNN), and Self-Attention models are compared (Vaswani et al., 2017) 24

Figure 3.1 Transformer architecture modified to receive audio embeddings along text embeddings as inputs in both the encoder and the decoder stack 47

Figure 4.1 Standard Encoder-Decoder Transformer Architecture 62

Figure 4.2 Modified Transformer architecture named the Reencoder 63

Figure 4.3 Diagram of the modified Transformer architecture presented by Riley et al. (2021) 64

Figure 4.4 Diagram of the modified Transformer architecture used in the current research 65

Figure 4.5 Diagram of the modified Transformer architecture named the Extractor 65

Figure A Technical specifications for the GPU used in the experiments conducted 150

Abstract

ENHANCING TRANSFORMER ARCHITECTURES FOR DIALOGUE modelling THROUGH CONTEXTUAL REENCODING

GUENDALINA CALDARINI

A thesis submitted to University of Sunderland for the degree of Doctor of Philosophy,

2024

Chatbots have emerged as intelligent conversational computer programs that simulate human conversation, processing user input and generating relevant responses. They find applications across diverse fields, offering support, assistance, and entertainment to users. Recent advancements in Artificial Intelligence and Natural Language Processing have led to the widespread adoption of chatbots, driven by increased computational power and the availability of open-source technologies. However, challenges remain in improving contextual understanding, emotional responsiveness, and addressing gender biases in chatbot interactions. Despite their prevalence, existing chatbot models often rely on a next-step approach, lacking the ability to consider the broader conversational context and underlying information shared among participants.

This thesis investigates the impact of contextual embedding information on transformer architectures for dialogue modelling tasks. Through a series of experiments, various transformer architectures were evaluated, and an innovative architectural approach called the Reencoder model was developed. A key feature of this new architecture is the inclusion of an additional reencoding step. This reencoding process enhances the model's capability to effectively capture and incorporate contextual information from previous turns in the dialogue history. It was consistently observed that such models exhibited superior performance and greater consistency compared to those employing alternative embedding strategies. This study sheds light on the mechanisms underlying the enhanced performance of contextual embedding layers and explores factors contributing to their effectiveness in dialogue modelling tasks. To strengthen the validity of the presented results, the thesis also presents enhanced algorithms that combine textual and audio embeddings that further enhance contextual understanding in dialogue modelling. The findings contribute to the ongoing research efforts aimed at improving chatbot implementations and evaluation methodologies, addressing critical challenges in human-chatbot interaction and advancing the field of conversational AI.

Declaration

No portion of the work referred to in this thesis has been submitted in support of an application for another degree or qualification of this or any other university or other institute of learning.

Copyright

- The author of this thesis (including any appendices and/or schedules to this thesis) owns certain copyright or related rights in it (the “Copyright”) and s/he has given University of Sunderland certain rights to use such Copyright, including for administrative purposes.

- Copies of this thesis, either in full or in extracts and whether in hard or electronic copy, may be made only in accordance with the Copyright, Designs and Patents Act 1988 (as amended) and regulations issued under it or, where appropriate, in accordance with licensing agreements which the University has from time to time. This page must form part of any such copies made.

- The ownership of certain Copyright, patents, designs, trademarks and other intellectual property (the “Intellectual Property”) and any reproductions of copyright works in the thesis, for example graphs and tables (“Reproductions”), which may be described in this thesis, may not be owned by the author and may be owned by third parties. Such Intellectual Property and Reproductions cannot and must not be made available for use without the prior written permission of the owner(s) of the relevant Intellectual Property and/or Reproductions.

- Further information on the conditions under which disclosure, publication and commercialisation of this thesis, the Copyright and any Intellectual Property and/or Reproductions described in it may take place is available in the University IP Policy (see https://www.ip-rank.co.uk/individual-report/?id=201), in any relevant Thesis restriction declarations deposited in the University Library, The University Library’s regulations (see https://library.sunderland.ac.uk/images/internal-websites/uls/policy-and-regs-and-guides-pdfs/Library-regulations---version-March-2020.pdf) and in The University’s policy on presentation of Theses.

Dedication

In dedication to all the amazing women who have made this journey so much easier.

Acknowledgements

My deepest gratitude goes to Dr. Sardar Jaf, for his continued feedback, support, and understanding throughout this project. He guided me so positively, and I always came off our meetings with renewed motivation and confidence in my abilities.

Thank you to Dr. Kenneth McGarry for his support and guidance during these four years.

I would also like to thank my husband for the endless nights and weekends he spent encouraging me, acting as my rubber duck, and believing in me even when I didn't.

Ai miei nonni, per averci dato un futuro migliore – to my grandparents, for the future they gave us.

List of Abbreviations

The following is a list of various abbreviations used throughout the thesis.

Abbr.	**Full form**
AI	Artificial Intelligence
ASR	Automatic Speech Recognition
BERT	Bidirectional Encoder Representations from Transformers
BLEU	Bilingual Evaluation Understudy
BPE	Byte-Pair Encoding
ELMo	Embeddings from Language Models
GloVe	Global Vectors for Word Representation
GPT	Generative Pre-trained Transformer
GPU	Graphics Processing Unit
HTML	HyperText Markup Language
JSON	JavaScript Object Notation
LM	Language Model
LLM	Large Language Model
METEOR	Metric for Evaluation of Translation with Explicit Ordering
ML	Machine Learning
MP4	MPEG-4
MT	Machine Translation
NLP	Natural Language Processing
OOV	Out Of Vocabulary
RL	Reinforcement Learning
RLHF	Reinforcement Learning from Human Feedback
SLM	Small Language Model
TER	Translation Error Rate
URL	Uniform Resource Locator
WAV	Waveform Audio File Format

Chapter 1: Introduction

Chatbots have emerged as powerful conversational agents that simulate human-like interactions, offering users support, assistance, and entertainment across diverse domains. However, existing chatbot models often rely on a limited, next-step approach, failing to fully leverage the broader conversational context and underlying information shared among participants. This research aims to address this limitation by investigating the impact of contextual embedding information on transformer architectures for dialogue modelling tasks.

The growing demand for AI language models (LMs), exemplified by the widespread adoption of systems like ChatGPT, underscores their transformative potential across various domains. However, the accessibility challenges posed by the substantial computational resources required for training and deploying such sophisticated AI models present significant barriers, particularly for smaller businesses and individual developers.

The formidable infrastructure demands, including the need for high-performance computing hardware, extensive datasets, and specialised technical expertise, often render it financially and logistically daunting for many stakeholders to develop or customise AI language models to meet their specific requirements (Pan et al., 2023). This digital divide exacerbates disparities in innovation and competitiveness, as entities without access to substantial resources or established infrastructure struggle to harness the full potential of AI LMs.

Amidst the exploration of existing architectures and embedding methods, an opportunity to innovate and address existing limitations was identified. Motivated by this observation, the study introduced a novel architecture known as the Reencoder model. This architectural innovation incorporates an additional re-encoding step, which enriches the model's ability to capture and integrate contextual information from previous turns in the conversation.

By developing more efficient and accessible dialogue modelling approaches, this research aims to complement and potentially offer an alternative to the resource-intensive LM solutions currently dominating the market. Improving the performance of smaller, task-oriented language models not only addresses the challenges of infrastructure demand but also holds the potential to mitigate the environmental impact associated with the operation of large-scale LMs (Weidinger et al., 2022).

The primary objective of this thesis is to explore a new deep learning framework that departs from the traditional next-utterance prediction paradigm for conversational AI. By incorporating past dialogue turns into the training process, the proposed architecture aims to model conversations in a more holistic and contextually-aware manner, emulating the way humans engage in discourse by considering prior statements and the overarching conversational context. In order to do so, the study compares and contrasts three different deep learning architectures against a baseline model provided by the TensorFlow organisation. The architectures to be explored are as follows:

- **Baseline**: The TensorFlow transformer model architecture developed for dialogue modelling, which will serve as the baseline.
- **Encoder-decoder transformer**: An encoder-decoder transformer architecture, identical to the TensorFlow baseline, except for the embedding layer, where three different embedding methods will be employed and compared.
- **Extractor:** An extended transformer architecture based on the one proposed in (Riley et al., 2021), where a fixed-width "contextual vector" extracted from the preceding sentence is used to provide contextual information during training. This architecture was named the "Extractor," and three different embedding methods will be evaluated.
- **Reencoder**: A novel transformer architecture that uses the embeddings of the previous utterance in the conversation to "inform" the embeddings of the current turn, in order to provide contextual information. This architecture is named the "Reencoder," and three different embedding methods will be explored.
- **Multimodal Architectures**: recognizing the importance of multimodal data in real-world conversational scenarios, this research extends the proposed architectures to handle both textual and audio input.

By incorporating audio embeddings alongside text embeddings, the extended architectures aim to capture the rich information contained in speech, such as tone, emotion, and acoustic features, enhancing the overall understanding and generation of contextually appropriate responses.
The multimodal extension involves integrating pre-trained audio embeddings from raw audio data. These audio embeddings are then combined with the textual embeddings using various fusion techniques, such as concatenation, attention mechanisms, to achieve multimodal transformers. The fused multimodal embeddings are then fed into the proposed architectures for training and inference.

By comparing the performance of these architectures, this research aims to provide insights into the mechanisms underlying the enhanced performance of contextual embedding layers and explore the factors contributing to their effectiveness in dialogue modelling tasks, as well as proposing viable solutions to address the step-to-step approach currently used for dialogue modelling. The findings of this study will contribute to the ongoing efforts to improve chatbot implementations and evaluation methodologies, addressing critical challenges in human-chatbot interaction and advancing the field of conversational AI.

1.1 Research Background Overview of Chatbot Design

Chatbots have revolutionised the way users interact with digital systems, offering human-like conversational experiences across various domains. However, current chatbot models often employ a limited, next-step approach that fails to capture the broader context and information flow within a conversation, as humans naturally do. This limitation hinders the ability of chatbots to engage in truly natural and contextually aware dialogues.

1.1.1 Research Aim and Objectives

The primary aim of this research is to investigate the impact of incorporating contextual embedding information into transformer architectures for dialogue modelling tasks, through previous turns of the conversation and by combining audio and text embeddings in multimodal dialogue models. Specifically, the objectives are:

1. Develop and evaluate novel deep learning architectures that move beyond the traditional next-utterance prediction paradigm.
2. Integrate previous dialogue turns into the training process to model conversations holistically, considering prior statements and the overarching context.
3. Extend the proposed architectures to handle both textual and audio input to achieve multimodal chatbots.
4. Compare and contrast the performance of different architectures, embedding methods, and contextual integration techniques.

1.1.2 Research Contribution

This research contributes to the field of conversational AI by:

1. Introducing an enhanced transformer architecture, the “Extractor,” inspired by Riley et al., (2021), which incorporates a fixed-width contextual vector from preceding sentences.
2. Introducing an enhanced version of the Google TensorFlow baseline architecture by extending it to utilise BERT embedding algorithm and audio data modality.
3. Proposing a novel transformer architecture, the “Reencoder,” that utilises contextual information through the use of previous utterance embeddings to inform the current turn.
4. Offering three audio unimodal systems based on different deep learning architecture designs.
5. Submitting three multimodal architectures trained on text and audio data.
6. Offering insights into the mechanisms underlying the enhanced performance of contextual embedding layers and the factors contributing to their effectiveness in dialogue modelling tasks.

1.1.3 Research Methodology

The research methodology involves a comprehensive approach to develop and evaluate novel neural architectures for dialogue modelling tasks. The first step involves designing and implementing the proposed architectures, namely the Encoder-Decoder Transformer, Reencoder, and Extractor models, along with baseline models for comparison. These architectures are then integrated with different embedding methods, including the popular GloVe and BERT embeddings, to capture semantic and contextual information from the input data.

To enhance the capabilities of the proposed architectures, they are extended to handle multimodal input, allowing for the processing of both textual and audio data. This extension

is particularly valuable for applications such as speech-based dialogue systems or audio transcription tasks.

The next step involves training and evaluating the models on various dialogue modelling tasks using appropriate datasets. This process involves carefully selecting and preprocessing the data to ensure that it is suitable for the specific task at hand.

Once the models are trained, a comparative analysis is conducted to evaluate the performance of the proposed architectures, baseline models, and different embedding methods. This analysis involves measuring relevant metrics, such as accuracy, perplexity, or other task-specific evaluation criteria.

Finally, the research methodology includes interpreting the results obtained from the comparative analysis. This step involves identifying the factors that contribute to the effectiveness of the contextual embedding layers and understanding how they influence the overall performance of the models. Additionally, potential limitations or areas for improvement are identified, providing insights for future research directions.

Throughout the research methodology, a rigorous and systematic approach is employed to ensure the validity and reproducibility of the results. The integration of various embedding methods, the extension to multimodal input, and the comprehensive evaluation and analysis contribute to the development of robust and effective dialogue modelling systems.

1.2 Thesis Outline

Chapter 1: Introduction
This chapter provides an overview of the research, highlighting the background, motivation, and objectives.
Chapter 2: Literature Review
A comprehensive review of relevant literature, covering chatbot design, dialogue modelling, transformer architectures, and contextual embedding techniques.
Chapter 3: Research Methodology
This chapter describes the research methodology, including the design and implementation of the proposed architectures, the integration of embedding methods, datasets used, and the evaluation strategies.
Chapter 4: Implementation
This chapter presents the experimental setup and a detailed description of the architectures considered.
Chapter 5: Experiments and Results
This chapter provides a detailed analysis of the results obtained from the comparative evaluation of the proposed architectures and baseline models.
Chapter 6: Discussion
An in-depth discussion of the research findings, highlighting the impact of contextual embedding information on dialogue modelling tasks, and the factors contributing to the effectiveness of the proposed architectures.
Chapter 7: Conclusion and Future Work

This chapter summarises the main conclusions of the research, outlines the contributions, and provides recommendations for future work in the field of conversational AI and chatbot design.

Chapter 2: Literature Review

2.1 Introduction

Chatbots are intelligent conversational computer programs that mimic human conversation in its natural form (Jia, 2003; Sojasingarayar, 2020). A chatbot can process user input and produce an output (Ayanouz et al., 2020; Kumar & Ali, 2020). Usually, chatbots take natural language text as input, and the output should be the most relevant output to the user input sentence. Chatbots can also be defined as “online human-computer dialogue system(s) with natural language”. (Cahn, 2017)

In recent years, with the commoditization and the increase of computational power and the sharing of open-source technologies and frameworks, chatbots programmes have become increasingly common. Recent developments in Artificial Intelligence and Natural Language Processing techniques have made chatbots easier to implement, more flexible in terms of application and maintainability, and increasingly capable to mimic human conversation. However, human-chatbot interaction is not perfect; some areas for improvements are contextual and emotional understanding and gender biases. Chatbots are, in fact, less able to understand conversational context (Christensen et al., 2018) and emotional linguistic cues compared to humans, which affects their ability to converse in a more entertaining and friendly manner (Fernandes, 2018). At the same time, chatbots tend to take on traditionally feminine roles which they carry out with traditionally feminine features and often displaying stereotypical behaviour, revealing a gender bias in chatbots’ implementation and application (Costa, 2018).

Chatbots are nowadays applied to a variety of different fields and applications, spanning from education to e-commerce, encompassing healthcare and entertainment, where they appear to be more engaging to users than static Frequently Asked Questions (FAQ) pages and can simultaneously assist multiple users, resulting in increased productivity and cost-effectiveness compared to human customer support services (Okuda & Shoda, 2018). In addition to support and assistance, chatbots can provide entertainment and companionship for end-users (Costa, 2018). Nonetheless, different levels of embodiment – how human-like the chatbot is (Go & Sundar, 2019)– and disclosure – how and when the nature of the chatbot is revealed to the user (Luo et al., 2019)– seem to impact users' engagement with and trust in chatbots. Given their extensive adoption and versatile applications across diverse fields, it is imperative that ongoing research focuses on enhancing their implementations and refining evaluation methodologies. As chatbots continue to play increasingly integral roles in various sectors such as customer service, healthcare, education, and entertainment, addressing existing limitations and optimising their functionalities becomes paramount. Therefore, exploring innovative approaches to improve chatbot performance, usability, and adaptability is essential to unlock their full potential and meet the evolving needs of users in different domains (Caldarini et al., 2022).

The underlying problem is that this model tries to solve conversational problems with a next-step approach: given an input, it tries to predict the best fitting output. This is, however, not the reasoning behind human conversation, that does not simply advance one step at a time, but rather by taking into consideration a series of previous steps, the underlying context of the conversation, and the information being shared among the participants (Vinyals & Le, 2015).

This chapter will focus on providing an overview of chatbot implementation methods. A distinction will be drawn between two approaches to chatbot design: Rule-based chatbots and Machine Learning (ML) based chatbots. Within ML-based chatbots, a further distinction will be drawn between Information-Retrieval chatbots and Generative Chatbots. A distinct section will be dedicated to transformers and transformer-based chatbots, as these are the most recent algorithms applied to the problem of Dialogue Modelling.

2.2 Research Background

Alan Turing is thought to be the first person to have conceptualised the idea of a chatbot in 1950, when he asked: “Can machines think?”. Turing’s description of the behaviour of an intelligent machine evokes the commonly understood concept of a chatbot (Turing, 1950).

Chatbots have evolved with the progressive increase in computational capabilities and advances in Natural Language Processing tools and techniques, and are nowadays applied to a variety of fields, spanning from education to e-commerce, encompassing healthcare and entertainment.

The emergence of deep learning has expanded chatbot applications like smart personal assistants (Alexa, Siri, etc.). These voice-controlled assistants use speech recognition to convert audio to text, and natural language processing to understand user intent and requests.

Despite the popularity of chatbots, creating chatbots that deliver satisfactory responses to the requirements of specific users remains an arduous task. For example, a chatbot must understand any user's speech or text as an input request and respond appropriately (e.g., on the same topic, make sense), helpfully (e.g., contains useful and concrete information), and even be tone-aware (e.g., conveys feelings like empathy and passion) (A. Xu et al., 2017; Hu et al., 2018).

2.3 Rule-Based approaches to Chatbots

A traditional approach to designing chatbots is the rule-based method (Young et al., 2013; Mesnil et al., 2015). This method models the dialogue flow as a structured sequence of slots or fields that need to be populated through the conversation. The chatbot's responses are generated based on a predefined set of rules and patterns that are manually crafted by developers. These rules map specific dialogue states and slot configurations to predefined response templates or actions. For example, if the user provides their location, a rule may dictate that the chatbot should then ask for the desired cuisine to recommend a restaurant.

ELIZA

ELIZA, developed at MIT in 1966, is considered the first chatbot in history (Weizenbaum, 1966; Shum et al., 2018; Zemčík, 2019). Functioning as a Rogerian psychotherapist, it advanced conversations by rephrasing user statements without needing to fully understand input. ELIZA used "direct match" pattern matching rules to reformulate input and generate responses.

PARRY

PARRY (1972) simulated a paranoid schizophrenic patient's speech. In tests, psychiatrists could only identify real patients versus PARRY transcripts 52% of the time (Colby et al., 1972).

A.L.I.C.E.

A.L.I.C.E. was built on Artificial Intelligence Mark-up Language (AIML), designed to expand its dialogue knowledge base. It searched through input words to find the closest matches using folders and subfolders, representing a significant improvement over earlier systems while still relying on pattern-matching rules.

ChatScript

ChatScript was developed to process user text input and provide responses by manipulating natural language. It transforms input words using replacement files and databases for texting, spelling errors, contractions, abbreviations, noise, and interjections mapped to speech acts. ChatScript marked a shift toward semantic analysis and comprehension in chatbot development (Wilcox, 2014; AbuShawar & Atwell, 2015; Cahn, 2017; Shum et al., 2018; Zemčík, 2019).

Limitations

Rule-based models are prone to give incorrect replies if they encounter a sentence that does not fit any established pattern. Additionally, manually encoding pattern-matching rules may be time-consuming and complicated. While rule-based chatbots can be effective in constrained domains with well-defined dialogue flows, they have limitations in handling open-ended conversations or adapting to unexpected user inputs. The hand-crafted rules need to anticipate and account for various scenarios, which can be time-consuming and challenging, especially in complex domains.

Rule-based systems often struggle with understanding the contextual nuances and implicit intent in natural language, leading to potential misunderstandings or irrelevant responses. As a result, more advanced techniques, such as those based on machine learning (ML), have gained popularity in recent years to develop more flexible and adaptive conversational agents.

2.4 Artificial Intelligence Approaches to Chatbots (Machine Learning Powered)

Contrary to rule-based models, ML models are built using machine learning algorithms to learn from a library of recorded human interactions. The algorithms are initially applied to samples of data (known as training data) to learn patterns, features, and various information. This is known as the training phase in AI chatbot design. This phase produces a model that can produce responses in communicating with humans. This type of chatbot may be more adaptable and independent of domain-specific expertise thanks to the usage

of ML techniques, which eliminates the need to manually create and implement new pattern-matching rules. AI-based chatbots include two types: information retrieval-based models and generative models.

2.4.1 Information-Retrieval Chatbots

Information retrieval-based (IR) models are created such that the algorithm can successfully retrieve the required information from a given dataset based on the user's input. Typically, a Shallow Learning algorithm is employed. A database of question-answer pairs often serves as the knowledge foundation for these types of models. This database is used to create a chat index, which lists all potential responses according to the message that provoked them. An information retrieval approach like those used for online searches is used to match the user's input to comparable ones in the chat index when the user presents the chatbot with an input. Thus, the response sent to the user is the answer coupled with the chosen query from those listed in the chat index (Shum et al., 2018). The key benefit of this approach is that it guarantees the replies' quality because they are not produced mechanically. With the rise of Web 2.0 and the availability of more textual material on social media platforms, forums, and chats, these models have become increasingly popular (Yan et al., 2016).

One of the biggest drawbacks of IR chatbots is it can be expensive, time-consuming, and laborious to create the prerequisite knowledge base. Furthermore, matching a user's input to the right answer could be time inefficient due to processing/searching a large amount of data available, which also means a larger training set and knowledge base. A significant amount of time and resources must be used to train the system to choose one of the available correct answers (Yan et al., 2016). Finally, information retrieval systems are less suitable for conversational agents, the so-called social chatbots, because they do not develop replies but rather retrieve them from a pre-defined set in their knowledge base, and lack personality development, which is a crucial quality for this type of chatbot (Shum et al., 2018).

The development of novel information retrieval algorithms has recently, however, made some headway. It is important to note that machine learning techniques are now utilised as the foundation for these kinds of models.

The method provided by Yan et al. is a breakthrough that examines past turns in the discussion to gather additional contextual information and enhance the output's quality and accuracy (Yan et al., 2016). In this approach, a Deep Neural Network rates not only the question/answer pairs that correspond with the most recent user's input but also those question/answer pairs that match with rephrased versions of past conversation turns, enhancing the information retrieval process. The rating lists for the various reformulations are then combined. By doing so, contextual data from the user's prior inquiries may be utilised, and these bits of data can be used to extract a better response from the knowledge base (Yan et al., 2016).
By leveraging contextual data from the user's prior inquiries, this method allows the chatbot to extract a more relevant and coherent response from its knowledge base. The incorporation of information from previous turns in the conversation represents a significant step forward in addressing the limitations of traditional IR-based chatbots, which often

struggle to maintain contextual awareness and provide responses that are truly responsive to the evolving dialogue.

2.4.2 Generative Models

As their name implies, generative-based models create new replies word by word based on the user's input. Thus, these models may generate whole new sentences in response to user requests. But they must be trained to understand grammar and sentence structure, thus their results may not always be of high quality or consistency (Sutskever et al., 2014; Shang et al., 2015; Sordoni et al., 2015; Vinyals & Le, 2015; Sojasingarayar, 2020). Typically, generative models are trained on a sizable dataset of real-world conversational terms. By providing it with training data the model learns vocabulary, grammar, and sentence structure. The overarching goal is for the algorithm's ability to provide a suitable and linguistically sound answer based on the input text. This strategy often relies on a Deep Learning algorithm, which is made up of an encoder-decoder neural network model with long short-term memory mechanisms (Vinyals & Le, 2015).

Sequence-to-Sequence models are now the de facto norm for chatbot modelling among AI models.
Before Transformers, RNNs (including LSTMs and GRUs) dominated NLP applications. These encoder-decoder models processed input sentences sequentially, with the encoder creating a context vector and the decoder generating responses word by word. The goal was to produce the most probable response given the conversational context. During training, the model learned through backpropagation, while inference used either beam search (selecting the most probable candidate) or greedy search methods (Vinyals & Le, 2015). Traditional sequence-to-sequence models used separate RNNs for encoding and decoding. To address context retention issues with longer sequences, attention mechanisms were added to focus on keywords that significantly contributed to generating the target sequence.
The well-known attention technique was added as a final layer to “pay attention” to keywords in the sequence that significantly contribute to the production of the target sequence, since word-level encoding makes it difficult for the encoder to keep context for longer sequences. Each word in the input sequence is given attention based on how it affects the creation of the target sequence.

Figure 2.1 represents a Recurrent Neural Network. The diagram illustrates how a Recurrent Neural Network (RNN) processes input and generates output. Green boxes symbolise individual words from the input sentence X, while orange boxes represent words in the output sentence Y. X(1) is the first word in the input sentence, while X(Tx) is the x^{th} word in the input sentence. Similarly, Y(1) is the first word in the output sentence, while Y(Tx) is the y^{th} word in the input sentence. A grey box indicates the initial activation function. Blue cells show the network's hidden states. The graph demonstrates that the RNN calculates each time step sequentially, starting with time step one before progressing to subsequent steps.

These models presented two drawbacks, though:

1. **Long-Range Dependencies**: Dealing with long-range dependencies between words that were placed far apart in a lengthy phrase proved difficult (Figure 2.1). As the graph shows, the RNN must compute the calculations at time step one before it is possible to move to the following time step. For a more detailed representation, see (CS 230 - Recurrent Neural Networks Cheatsheet)

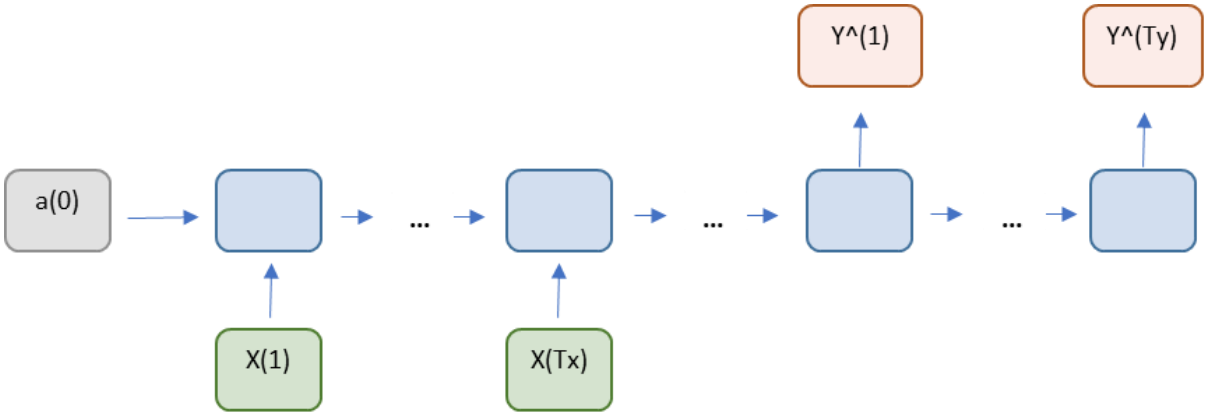

Figure 2.1. A Recurrent Neural Network.

2. **Sequential Computation**: They handle the input sequence sequentially, processing each word one at a time, hence they cannot compute for time-step *t* until they have computed for time-step *t — 1*. Training and inference take longer as a result (Refer to Figure 2.1).

The above two drawbacks are addressed by the Transformer architecture. It completely abandoned RNNs in favour of relying only on the advantages of Attention. They perform a parallel processing operation on each word in the sequence, accelerating computing.

The Seq2Seq approach offers several benefits. It is an end-to-end solution that can be trained on various datasets, making it applicable across many domains without the need for domain-specific expertise. Even though the model may produce useful results without domain-specific information, it can still be modified to operate with different algorithms if additional research on domain-specific knowledge is required. Thus, it is a straightforward yet broadly generic and versatile model that may be used for many NLP (Natural Language Processing) tasks (Vinyals & Le, 2015; Shum et al., 2018).

However, as the generated response length increases, the model struggles to maintain coherence and consistency, as the entire input must be captured in the fixed-length context vector. This can result in ambiguous or incoherent responses. Additionally, these models concentrate on a single response when generating an answer, which results in a lack of coherence in the conversational turns (Strigér, 2017; Jurafsky & Martin, 2020; Sojasingarayar, 2020).

2.4.2.1 Transformer Chatbots

Transformers have helped to progress the field of NLP since their method of reading is more in line with human behaviour than traditional sequential procedures. Since artificial intelligence (AI) solutions are becoming increasingly popular among those who lack specialised technical expertise in the field, models have an increasing need for robustness, explainability (which is the need to understand the way model parameters generate responses), and accessibility as more sectors turn to AI solutions (Bird et al., 2021). The Transformer based model is a fairly novel idea in the field of deep learning (Vaswani et al., 2017). The theory underlying the investigation of transformers in NLP is their more natural approach to sentences than other deep learning approaches, such as sequence-to-sequence. This is comparable to the human propensity to fully "listen" to a statement or sequence before reacting appropriately in conversations, translations, or other tasks of a similar kind.

The foundation for sequence transduction activities is the sequence-to-sequence encoder-decoder architecture. It advises encoding the entire sequence at once and utilising that encoding as the background for creating the target sequence or the decoded sequence.

Instead of suffering from the vanishing gradient problem present in Recurrent Neural Networks, (Schmidhuber, 1992), transformer-based models pay attention to tokens in a learned order and as a result enable more parallelization while improving upon many NLP problems. New benchmarks and standards have been established since the application of Transformers to a variety of fields (Vaswani et al., 2017).

In the original Transformer article (Vaswani et al., 2017), several parameters for the models were compared, as presented in Figure 2.2:

Layer Type	Complexity per Layer	Sequential Operations	Maximum Path Length
Self-Attention	$O(n^2 \cdot d)$	$O(1)$	$O(1)$
Recurrent	$O(n \cdot d^2)$	$O(n)$	$O(n)$
Convolutional	$O(k \cdot n \cdot d^2)$	$O(1)$	$O(log_k(n))$
Self-Attention (restricted)	$O(r \cdot n \cdot d)$	$O(1)$	$O(n/r)$

Figure 2.2. A comparison of RNN, Convolutional Neural Network (CNN), and Self-Attention models in terms of computational efficiency conducted by (Vaswani et al. 2017)

In Figure 2.2, *n* is the sequence length (often in the range of 40–70), *k* is the convolution kernel size, and *r* is the attention window size for restricted self-attention.
D (or *d_model*) is the representation dimension or embedding dimension of a word (typically in the range 128–512). From the figure presented above, the following points may be deduced:

- **Lower Computational Complexity**: Self-attention has a lower per-layer computational complexity than other attention does.
- **Parallelization**: Except for RNNs, all other methods allow parallelization when it comes to sequential processes, hence their complexity is *O(1)*.

- **Path Length**: The fourth statistic is maximum path length, which on the surface refers to the difficulties of attention to distant words or long-term dependencies. Self-attention models attend all the words at the same step; hence their complexity is *O*, but convolutional models employ hierarchical representations, which makes their complexity *n∗log(n)(1)*.

The Transformer utilises a self-attention mechanism, which makes parallelization easy by calculating attention weights using all the words in the input sequence at once. Additionally, because the Transformer's per-layer operations include words in the same sequence, the complexity is less than *O(n2d)*. As a result, the transformer is shown to be both a computationally efficient model and effective (because it makes use of attention).

Due to these factors, such methods are quickly creating State-of-the-Art results for various NLP issues (Tenney et al., 2019). The following are examples of text data processing techniques that have benefitted from the application of Transformers: generation (Devlin et al., 2019; Radford et al., 2019), question answering (Lukovnikov et al., 2019; Shao et al., 2019), sentiment analysis (Naseem et al., 2020; Shangipour ataei et al., 2020), paraphrasing (Chada, 2020; Lewis et al., 2020), translation (Zhang et al., 2018; Wang et al., 2018; Gangi et al., 2019), and classification (Sun et al., 2019).
The basic structure of the Transformer consists of a stack of encoder and decoder layers. To prevent misunderstanding, each layer will be referred to as either an encoder or a decoder, and a stack of encoder layers or decoder layers will be referred to as either an encoder stack or a decoder stack.
For each of their inputs, the Encoder stack, and the Decoder stack each have an associated Embedding layer. An output layer is present at the end to produce the finished product. The Encoders are all exact replicas of one another. In a similar vein, every decoder is the same. The crucial Self-attention layer, which calculates the relationships between the words in the sequence, is included in the encoder along with a Feed-forward layer. The Self-attention layer, Feed-forward layer, and a second Encoder-Decoder attention layer are all included in the Decoder.
There is a unique set of weights for each encoder and decoder.
All Transformer designs are defined by a reusable module called The Encoder. Along with the two levels mentioned above, it also features two LayerNorm layers and Residual skip connections all around both layers.
Transformer architecture comes in a variety of forms. Some Transformer topologies solely rely on the encoder and lack any sort of decoder.
At the very core of the Transformer's innovation lies the computation of scaled dot product attention units, according to (Vaswani et al., 2017). Each word in the input word vector has its weights determined (document or sentence). The attention unit's output is an embedding for each relevant token combination in the input sequence. The following formula (Equation 1) is used to determine the query *Wq*, key *Wk* and value *Wv* weights:

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

Equation 1. The equation defining the Attention mechanism

Q, K, and V represent the Query, Key, and Value matrices respectively, and d_k denotes the dimensionality of the key vectors. The mechanism operates by first computing compatibility scores between queries and keys through matrix multiplication (QK^T), followed by scaling these scores by √d_k to maintain stable gradients during training. The resulting scores undergo softmax normalization, generating a probability distribution that determines the relative importance of each value vector. The final output is obtained by multiplying these attention weights with the value matrix V, producing context-aware representations where each position in the sequence incorporates information from all other positions, weighted by their relevance. This self-attention formulation enables the model to capture long-range dependencies and complex relationships within the input sequence, while maintaining computational efficiency through parallelization. The scaling factor √d_k proves crucial in preventing the dot products from growing too large in magnitude, particularly in high-dimensional spaces, thereby avoiding regions of extremely small gradients in the softmax function that could impede effective learning.

K and V are derived from the source, whereas Q is derived from the goal for tasks like classification and translation. For instance, *Q* may be a class that the text is assigned to, such as “positive” and “neutral” for sentiment analysis and the classification model's prediction.

For supervised English-Spanish machine translation, values K and V might be taken from the English phrase “Hello, how are you?” while *Q* could be derived from the phrase “Hola, cómo estás?”

In the original paper by Vaswani et al., the authors introduced a crucial insight regarding the execution of the attention mechanism. Rather than executing a single attention function with dmodel-dimensional keys, values, and queries, they found it advantageous to linearly project the queries, keys, and values h times using learned linear projections to dimensions *dq*, *dk*, and *dv*, respectively. Subsequently, the attention function is applied in parallel to each of these projected queries, keys, and values, resulting in dv-dimensional output values. Finally, the output values are concatenated and projected once more to obtain the final values. This approach enables the model to simultaneously attend to multiple representation subspaces, enhancing its capacity to capture complex dependencies within the data.

The Multi-headed Attention theory is used to guide all the State-of-the-Art models benchmarked in these trials. This is just a wider network of interconnected attention units created by concatenating many attention heads together:

As opposed to a machine learning model, humans can compare one word in a phrase to other words in the same sentence; to produce human-like text, engineers, and researchers must produce a means for the model to acquire this understanding. This is where the self-attention notion comes into play. It is crucial to remember that humans do not read in a token-sequential fashion like traditional RNN models, as in the case of the Long Short-Term Memory (LSTM) network (Hochreiter & Schmidhuber, 1997).

However, training powerful Transformers still requires significant computational resources and a large amount of data. For this reason, research centres from large organisations like Google, Microsoft, OpenAI and HuggingFace train exceptionally large transformer models that can even reach hundreds of billions of parameters and then share these foundational models with the community, which can easily fine-tune them with fewer data and computational power.

This trend seems to apply to the application of Transformer models for dialogue modelling. Several works of literature have already investigated the possibility of fine-tuning foundational transformer models on dialogue modelling tasks.
Yu et al., (2020) developed a financial service chatbot based on BERT. The chatbot is closed domain and trained on the task of intent classification to correctly identify the intent of a question and provide a relevant answer among a specific set of possibilities. The authors also provided a novel discussion about uncertainty measures for BERT. Bathija et al., (2020) propose a chatbot for interactive learning. Questions to the user are generated through extractive summarisation of content. This allows users to obtain important sentences out of a document or set of documents. The summarisation is performed using BERT. Bird et al., (2021) propose augmenting conversational data through paraphrasing with the T5 model and then using augmented data to finetune several other transformers for dialogue generation. A novel approach has been developed by Google. Their neural conversational model, Meena, has one Evolved Transformer encoder block and thirteen Evolved Transformer decoder blocks. The encoder processes the conversation context for Meena to grasp what was said in the discussion. The answer is subsequently created by the decoder using that data. The study found that the key to better conversational quality was a more potent decoder by tweaking the hyperparameters (Adiwardana et al., 2020).

TextSETTR

This section introduces TextSETTR, a novel approach to few-shot text style transfer presented by Riley et al., (2021). TextSETTR tackles the challenge of extracting and leveraging stylistic cues from unlabeled text data. It accomplishes this by capitalising on the inherent co-occurrence of style with nearby sentences within a document. The core idea lies in leveraging a pre-trained text-to-text model, T5 (Raffel et al., 2020), and incorporating a fine-tuned style extractor module. This module learns a style representation based on the sentence preceding the target sentence (context). During training, TextSETTR explores three data corruption strategies to construct training examples: noise injection, back-translation, and noisy back-translation, each contributing a reconstruction loss term (Riley et al., 2021).

During training, the model explores three data corruption strategies to construct training examples:

- Noise Injection: This strategy involves adding, replacing, or shuffling tokens within a sentence.
- Back-Translation: The model itself is used to transfer the sentence into a different style using a chosen context sentence.
- Noisy Back-Translation: This combines noise injection with back-translation. Noise is first introduced to the sentence, and then the altered sentence is used as input for back-translation with a chosen context sentence.

TextSETTR is trained to reconstruct the original sentence from its corrupted version while incorporating the stylistic cues extracted from the context sentence. At inference time, TextSETTR employs a “targeted restyling” approach. Instead of directly targeting a specific style, it calculates a delta vector within a style space that represents the stylistic difference between source and target styles. This delta vector is then used to modulate the style of

the input sentence during the decoding process. (Riley et al., 2021). Evaluations on sentiment transfer tasks demonstrate that TextSETTR achieves competitive results compared to existing methods that require style-labelled training data. The model also exhibits the ability to generalise to other stylistic attributes, highlighting its potential for broader applicability (Riley et al., 2021).

The promise of TextSETTR for few-shot text style transfer extends naturally to the realm of dialogue modelling. In dialogue systems, capturing and leveraging the stylistic cues from previous conversational turns is crucial for generating coherent and consistent responses. Here's how TextSETTR can be adapted for this purpose. To its document-based application, TextSETTR's style extractor module can be employed to analyse the most recent turns in a dialogue. By treating these turns as a form of unlabelled text, the model can extract a contextual style representation that encapsulates the sentiment, formality, and overall tone of the conversation thus far. During dialogue generation, the decoder can incorporate this contextual style representation along with the current user input. This allows the model to not only address the literal meaning of the user's utterance but also tailor its response to match the established conversational style. For instance, if the conversation has been lighthearted and informal, TextSETTR can nudge the response towards a similar style, even if the user's current input is grammatically correct but devoid of emotion.
A key advantage of TextSETTR is its ability to generalise to various style attributes beyond those explicitly used for training. In dialogue modelling, this translates to adaptability across different conversation types. The model can, for example, adjust its formality based on whether it's interacting with a customer service representative or a casual acquaintance.

By incorporating TextSETTR, dialogue models can move beyond generic responses and generate text that is not only grammatically correct but also stylistically appropriate within the ongoing conversation. This can significantly enhance the naturalness and coherence of human-computer interactions. Future research directions could involve exploring techniques for dynamically weighting the contextual style representation based on the recency and importance of previous turns, allowing for even more nuanced control over dialogue style.

2.4.3 Multimodal Chatbots

While traditional chatbot architectures have focused primarily on textual data, real-world conversations often involve multimodal interactions where information is conveyed through multiple modalities such as speech, gestures, and visual cues. Existing text-only architectures fail to capture the rich contextual information contained in these additional modalities, resulting in a limited understanding and ability to generate appropriate responses (Poria et al., 2019). To enable more natural and effective human-computer interactions, there has been a growing interest in developing multimodal chatbots that can perceive and generate responses across multiple modalities.

Early work on multimodal dialogue systems focused on integrating speech recognition and synthesis capabilities with traditional text-based architectures. For instance, Lemon & Gruenstein, (2004) proposed a multithreaded architecture that combined speech acts with semantic representations to generate multimodal outputs. Similarly, Nakano et al., (2011)

developed a system that fused text, speech, and visual information for in-car dialogue interactions.

With the advent of deep learning and the success of transformer models in natural language processing, researchers have explored multimodal extensions of these architectures for conversational AI tasks. Rastogi et al., (2020) introduced a multimodal transformer that combines textual, visual, and acoustic features for multimodal machine translation and dialogue generation. Le et al., (2022) introduced a multimodal attention mechanism to selectively attend to different modalities during response generation. However, a key challenge with many of these approaches is that they treat modalities as separate parallel inputs, failing to fully capture the interplay and synergies between them in natural conversation (Tsai et al., 2019).

A key challenge in developing multimodal chatbots is effectively integrating and modelling the interactions between different modalities. Poria et al., (2019) highlighted the importance of capturing the interplay between modalities, as opposed to treating them as separate parallel inputs. Tsai et al., (2019) proposed the use of tensor fusion networks to jointly model the relationships between modalities for multimodal sentiment analysis.
Recent research has explored more sophisticated techniques for multimodal fusion and representation learning. Waligora et al., (2024) introduced a multimodal transformer architecture that learns joint embeddings for different modalities through cross-modal attention and fusion layers. Chen et al., (2021) proposed a modality-invariant encoder that projects different modalities into a shared embedding space, enabling cross-modal understanding and generation.
Another critical aspect of multimodal chatbots is accounting for modality mismatches, where the input and output modalities differ (J. Xue et al., 2023). For example, generating text responses to audio queries or vice versa. Existing unimodal architectures struggle with such cross-modal mappings, leading to potential information loss or inconsistencies.
Additionally, the development of multimodal chatbots is hindered by the lack of large-scale, diverse multimodal dialogue datasets (Liu et al., 2022). Most existing datasets are either unimodal (text-only) or limited in their coverage of domains, languages, and modalities. Alamri et al., (2019) highlighted the need for novel data collection pipelines and augmentation techniques to address this data scarcity issue.
Despite these challenges, the field of multimodal chatbots has witnessed significant advancements, driven by the development of sophisticated deep learning architectures and the increasing availability of multimodal data sources. As research in this area continues to progress, researchers can expect to see more natural and context-aware conversational experiences across diverse modalities and domains.

In contrast, the proposed enhanced architecture in this research takes a more holistic approach to multimodal integration. By fusing textual and audio embeddings at a deep level, using techniques such as concatenation and attention mechanisms, the enhanced architecture aims to capture the rich information contained in both modalities and the intricate relationships between them.

Furthermore, the proposed architecture builds upon the contextual awareness introduced by the Reencoder and Extractor architectures, incorporating multimodal information from previous turns to inform the current response generation process. This approach emulates

the way humans engage in discourse, considering not only the current utterance but also the broader conversational context, including the modalities used in previous turns.

Compared to other related works, the proposed enhanced architecture offers a more seamless integration of multimodal information, allowing the model to learn from the interplay between different modalities and their contextual relationships. By leveraging state-of-the-art techniques in multimodal fusion and contextual modelling, the enhanced architecture aims to provide a more natural and contextually appropriate response generation capability, addressing the limitations of previous architectures that were restricted to handling a single modality.

2.4.4 State of the Art

2.4.4.1 ChatGPT

No review of the current state of the art would be complete without mentioning ChatGPT. The foundation of ChatGPT is the GPT-3.5 model, a Transformer that only presents the Decoder component; such models are known as Decoder-only architectures or autoregressive models (*Decoder Models - Hugging Face NLP Course*, 2023). Large autoregressive language models are generally trained with a standard left-to-right language modelling objective on a large text corpus, where the objective is to predict the next token, taking into account the previous tokens.

Zero-shot generalisation is seen in large-pertained transformer language models. Nonetheless, there are notable differences between the various state-of-the-art models' designs and pretraining goals. The authors of T. Wang et al., (2022) provided a thorough analysis of modelling decisions and how they affect zero-shot generalisation. Three model architectures were tested: the causal decoder only, the non-causal decoder only, and the encoder-decoder model. Their main focus was on text-to-text models. These models were assessed both with and without multitasking-prompted fine-tuning. They were trained with two distinct pretraining objectives: autoregressive and masked language modelling.

The results of their experiment demonstrated that the optimal zero-shot generalisation for purely unsupervised pretraining is exhibited by causal decoder-only models trained on an autoregressive language modelling objective. They discovered that a pretrained non-causal decoder model may be modified into a successful generative causal decoder model by employing autoregressive language modelling as a downstream job. Additionally, they demonstrated how the non-causal decoder model might be modified to fit the pre-trained causal decoder model.

Like InstructGPT, ChatGPT was trained utilising Reinforcement Learning (RL), specifically Reinforcement Learning from Human Feedback (RLHF), albeit with a somewhat different configuration for data collecting. Supervised fine-tuning was used to train an initial model. Human AI trainers simulated discussions in which they took on the roles of both the user and an AI assistant. To assist the trainers in crafting their responses, model-written ideas were made available to them by OpenAI data scientists. Subsequently, they combined this new dialogue dataset with the dialogue-formatted InstructGPT dataset.

Comparison data, comprising two or more model responses ordered according to quality, was required in order to develop a reward model for reinforcement learning. AI trainers' chatbot talks were recorded by OpenAI engineers and data scientists in order to compile this information. After choosing a model-written message at random, they sampled a number of different completions and asked AI trainers to score them. They were able to use Proximal Policy Optimization to fine-tune the model by using these reward models. This was a process that was repeated several times.

ChatGPT consideration of previous turns in the conversation

Because ChatGPT remembers the context and history of a conversation, it can manage multi-turn conversations. It creates responses based on the previous exchange and the current input using a sequence-to-sequence paradigm. This enables it to produce responses that are pertinent to the conversation's earlier turns while preserving the conversation's coherence and continuity.

In order to manage discussions with several turns, ChatGPT usually saves the earlier turns of the conversation in a memory or context vector. This memory or context vector is then used as input to the model in addition to the current input. The procedure is then repeated for each turn in the discussion as the model creates a response based on the combined input.

By using this method, ChatGPT is able to produce responses that are more eloquent and human-like, while also keeping the same tone and manner throughout the exchange.

ChatGPT utilises its memory of prior communications to deliver responses that are both logical and pertinent to the context in which you are conversing. The conversation history is part of this context, which enables the model to interpret pronouns, references, and other contextual clues that are crucial for producing precise and insightful responses.

The context window is limited, though. Within the bounds of its maximum token limit, the model is able to retain and utilise recent messages. Should the discussion go on for too long, earlier points may be cut off and no longer add to the overall context.

Even though ChatGPT is capable of maintaining context, it may occasionally generate comments that seem irrelevant or out of place, particularly if the discussion gets complicated or unclear. Context management and state tracking are two ways that ChatGPT can handle tasks and actions that require multiple turns. It keeps track of the conversation so far, which enables it to comprehend the user's intent over several turns and retain context. In order to guarantee that it offers pertinent and accurate responses, it can also maintain track of the conversation's status, including the information exchanged and the actions performed.

However, it's crucial to understand that ChatGPT simply sends what is in the chat window back to the API each time users submit input; it does not retain track of conversations.

The chatbot in this case is not trained to consider previous turns of the conversation in order to inform subsequent answers, it is simply using the entire history of the conversation so far as input (*Can ChatGPT Understand Context?*, n.d.; *GPT-3.5-Turbo How to Remember Previous Messages like Chat-GPT Website - API*, 2023; *How Does ChatGPT Handle Multi-Turn Conversations?*, n.d.)).

Limitations

Despite its impressive flexibility, ChatGPT occasionally generates responses that appear logically sound but contain factual inaccuracies or reasoning errors. This presents a

significant challenge for three primary reasons: (1) the absence of an accessible truth reference for reinforcement learning training; (2) increased model caution during training phases paradoxically leads to the rejection of legitimately answerable queries; and (3) supervised training methodologies introduce interpretation biases, as optimal responses necessarily depend on the model's internal knowledge representation rather than the demonstrator's understanding (*Introducing ChatGPT*, n.d.).

The model demonstrates notable inconsistency in response quality and accuracy when confronted with variations in query phrasing. For instance, a question formulated in one manner might elicit a non-response claiming insufficient knowledge, while a semantically equivalent reformulation produces a correct and comprehensive answer. Additionally, the model frequently employs formulaic language patterns and repetitive self-references. These limitations stem from well-documented over-optimization phenomena and inherent biases in training data, where human evaluators demonstrate preference for lengthier responses that convey an impression of thoroughness (*Introducing ChatGPT*, n.d.).

A fundamental limitation observed across contemporary language models concerns their inefficient sampling during pre-training phases. Despite GPT-3's progression toward human-comparable test-time efficiency (zero-shot or one-shot learning), the model requires exposure to textual volumes substantially exceeding what humans encounter throughout their lifetimes (Linzen, 2020). Enhancing pre-training sample efficiency represents a critical research direction, potentially achievable through algorithmic innovations or integration with physically-grounded informational frameworks.

The few-shot learning paradigm implemented in GPT-3 presents conceptual uncertainties regarding whether the model genuinely acquires novel tasks during inference or merely recognizes and adapts previously encountered task structures. This spectrum encompasses multiple possibilities: de novo skill acquisition, recognition of familiar tasks presented in alternative formats, adaptation of generalized capabilities such as quality assessment, or identification of training examples drawn from distributions identical to those in testing scenarios. The model's position along this spectrum likely varies according to task specificity and complexity.

Synthetic tasks such as word scrambling or nonsense word definition appear particularly amenable to genuine learning, while translation capabilities must necessarily develop during pre-training, albeit potentially from stylistically and structurally diverse data compared to test examples. The delineation between knowledge acquired de novo versus from prior examples remains similarly ambiguous in human cognition. While even the capacity to organize and identify different demonstrations between pre-training and testing would represent significant progress for language models, precisely characterizing the mechanisms underlying few-shot learning remains an important avenue for future research.

Regardless of the objective function or algorithm, a drawback of models at the GPT-3 scale is that performing inference on them is costly and time-consuming. This could make it difficult for models at this scale to be practically applied in the present. Distillation (Hinton et al., 2015) of large models down to a manageable size for particular tasks is one potential future direction to address this. Big models like GPT-3 have a very broad skill set, most of which are not required for a particular task, indicating that aggressive distillation might be feasible in theory.

Distillation has been thoroughly studied overall (Liu et al., 2019), but it hasn't been tested with hundreds of billions of parameters; new challenges and opportunities may be associated with applying it to models of this size.

2.5 Advances in Chatbots and Dialogue Modelling post-ChatGpt

Recent advancements in chatbots and dialogue modelling have significantly transformed the landscape of conversational AI, particularly with the emergence of large language models (LLMs) and their applications in various domains. In 2023, the introduction of models like InstructTODS demonstrated the potential of LLMs in end-to-end task-oriented dialogue systems. This model achieved performance levels comparable to fully fine-tuned systems without requiring prior task-specific data, showcasing the ability of LLMs to guide dialogues effectively and produce responses that are more helpful, informative, and human-like than previous state-of-the-art systems (Chung et al., 2023). Such advancements highlight the growing capability of AI to manage complex dialogues autonomously, which is crucial for applications ranging from customer service to personal assistants.

Moreover, the exploration of generative models in conversational agents has been a focal point in recent literature. Wassan's work discusses the foundational architectures of these models, primarily based on recurrent neural networks (RNNs) and transformer architectures, which are pivotal in the development of sophisticated chatbots like ChatGPT (Wassan et al., 2023). The ability of these models to generate coherent and contextually relevant responses has led to their widespread adoption in various sectors, including education and business, where they enhance user engagement and streamline processes.

The integration of AI chatbots in specific demographics, such as Gen-Z voters, has also been studied. Tjahyana's research indicates that AI chatbots are particularly effective in engaging younger audiences, provided they are trained on relevant datasets to ensure accuracy and reliability in responses (Tjahyana, 2024). This underscores the importance of dataset quality in the performance of AI systems, as inconsistencies in training data can lead to incorrect outputs, necessitating rigorous training and evaluation protocols during the deployment phase.

Furthermore, the ethical implications of deploying AI chatbots have gained attention, particularly concerning their reliability and the potential for bias in responses. Meyer et al. discuss the transformative impact of LLMs in academia, emphasizing the need for ethical considerations in their use, especially regarding fairness and bias (Meyer et al., 2023). This is critical as the reliance on AI-generated content increases, raising questions about the integrity of information and the potential for misinformation.

In addition to these advancements, the development of memory-augmented models has introduced new capabilities for chatbots, allowing them to retain and utilize user-specific information over time. Sarch's research on memory-augmented large language models demonstrates how these systems can enhance user interaction by personalizing responses based on past dialogues, thereby improving the overall user experience (Sarch et al., 2023). This capability is particularly beneficial in applications requiring sustained engagement, such as mental health support and personalized learning environments.

2.6 Summary

This chapter has provided a comprehensive overview of approaches to chatbot design, tracing the evolution from early rule-based systems to advanced AI-powered models. The literature review explored the fundamental distinction between rule-based chatbots and those leveraging machine learning, including information retrieval and generative models. The emergence of transformer architectures, particularly the development of models like ChatGPT, has marked a significant leap forward in chatbot capabilities. These models demonstrate impressive language understanding and generation abilities, although they still face challenges such as maintaining long-term context and avoiding hallucinations. The chapter also touched upon the growing interest in multimodal chatbots, which aim to integrate various forms of input beyond text. As the field continues to advance, researchers are focusing on enhancing contextual understanding, improving sample efficiency, and addressing ethical concerns. While current state-of-the-art models like ChatGPT show remarkable versatility, ongoing work is needed to overcome limitations and further bridge the gap between artificial and human-like conversation.

Chapter 3: Research Methodology

3.1 Introduction

This research proposes to extend and enhance three different Transformer architectures by introducing novel modifications tailored to address their limitation modelling dialogue, including previous turns of the conversation. By strategically incorporating new features and techniques, the aim is to unlock improved performance and overcome inherent constraints in these architectures. While the work builds upon existing foundations, the innovative adaptations promise to push the boundaries of these models' capabilities, yielding superior results in various application domains. The research endeavours to make tangible contributions to the field by exploring novel approaches that capitalise on the strengths of these architectures while mitigating their contextual awareness weakness.
The architectures used are as follows:

1. A transformer model architecture developed for dialogue modelling by the TensorFlow organisation. This architecture has been used as a baseline model (*Google Colab*, n.d.).
2. An Encoder-Decoder Transformer architecture identical to that proposed by TensorFlow organisation, except in the embedding layer, modified by replacing the embedding layer.
3. A modified Transformer architecture, following the TextSETTR architecture proposed in (Riley et al., 2021), where a fixed-width “contextual vector” extracted from the preceding sentence is used to provide contextual information during training. This architecture has been named the Extractor.
4. A novel Transformer architecture that uses the embeddings of the previous utterance in the conversation to “inform” the embeddings of the current turn in the conversation, in order to provide contextual information. This architecture has been named the Reencoder.
5. Each one of these architectures has then been further extended, allowing them to use video input along with text inputs.

In the initial research design phase, Recurrent Neural Networks (RNN) and Long Short-Term Memory (LSTM) architectures were considered as potential comparison models alongside the transformer-based architectures. These traditional sequence modelling approaches had been the predominant choice for natural language processing tasks prior to the introduction of transformers. However, these architectures were ultimately excluded from the study based on substantial evidence from previous research demonstrating the superior performance of transformer-based models. Notably, Vaswani et al. (2017) demonstrated that transformers outperform RNNs and LSTMs across various language tasks while offering better parallelization capabilities and reduced training time. This finding has been further corroborated by subsequent studies, such as (Pölz et al., 2024), who conducted extensive empirical comparisons showing that transformer-based models consistently achieve higher accuracy and better handling of long-range dependencies compared to RNN-LSTM architectures. Therefore, to maintain focus on the most promising current approaches and avoid redundancy with existing literature, this study concentrated specifically on comparing different transformer-based architectures.

For the three proposed architectures (Encoder-Decoder Transformer, Extractor, and, Reencoder) the study explore three different embedding methods (a subword embedding layer created from the training data, and two pretrained embedding matrices, GloVe 200d and Bert base uncased).
To investigate the effectiveness of these transformer architectures in capturing conversational context for dialogue modelling, a rigorous experimental methodology was employed.
This chapter outlines the key steps undertaken, including dataset preparation, model architectures, and evaluation procedures.

Evaluation Procedures
To assess the performance of the three transformer architectures, a comprehensive evaluation protocol was uses:

Automatic Metrics: Standard natural language generation (NLG) metrics, including BLEU, TER, METEOR, and perplexity, were computed on the test set predictions to quantify the models' ability to generate relevant, coherent, and contextually appropriate responses. More information regarding automated metrics selected and discarded can be found in section 3.5.
Throughout the experiments, rigorous practices were employed to ensure reproducibility and statistical significance, including fixed random seeds, multiple training runs, and appropriate statistical tests for result comparisons.

This comprehensive methodology aimed to provide a systematic and unbiased evaluation of the transformer architectures' capabilities in dialogue modelling, specifically concerning their ability to leverage contextual information from previous conversational turns. The findings derived from this study contribute to the ongoing efforts in advancing the state-of-the-art in conversational AI and developing more natural and contextually aware dialogue systems.

3.2 Datasets

Five different dialogue datasets have been chosen to train these models.

The first corpus selected to train the architecture is the OpenSubtitles dataset (Lison & Tiedemann, 2016), an open domain dataset of film subtitles consisting of more than 441 million sentences in XML format. This specific dataset has been selected not only for its consequent size, but also for the variety of subjects and registers that it provides. Furthermore, the dataset has been used in previous studies (Christensen et al., 2018; Sojasingarayar, 2020; Vinyals & Le, 2015), allowing direct comparisons of the proposed models to previously published results. However, scene descriptions, close captioning, and segmented sentences can be found among the dialogues, which can be problematic when training an open domain chatbot because it can lose the cohesiveness of the dialogue (Vinyals & Le, 2015; Klein et al., 2017; Christensen et al., 2018; Zhong et al., 2019; *OpenSubtitles*, 2021). Nevertheless, the data does not seem to be of the highest quality.

The second corpus selected is the Cornell Movie-Dialogs Corpus (*Cornell Movie-Dialogs Corpus*, 2023). This dataset includes a sizable collection of fictitious dialogue taken from uncut film scripts, rich in metadata. The data is presented in TXT format. It provides 2,20,579 dialogues between 10,292 pairs of film characters; in total, 9,035 characters from 617 films are involved. There are 304,713 utterances in total, accompanied by metadata such as genre, year of release, number of IMDB votes and IMDB rating, and more metadata concerning the characters (*Cornell Movie-Dialogs Corpus*, 2023). For the purpose of this research, the metadata has been discarded and only the raw text has been preprocessed and used for training. The Cornell Movie-Dialogs Corpus has been selected for several reasons: firstly, it was used to train the Tensorflow transformer chatbot model used as a baseline in this study, and therefore provided a valuable comparison point. Secondly, it has been used in previous studies (Zhong et al., 2019; Ghandeharioun et al., 2019; Roller et al., 2021; He et al., 2021), allowing further comparison to previously published results. Since the data has been taken from film scripts, it appears to be of decent quality. However, the dataset might not be large enough to train more advanced language models.

The third dataset selected is the DailyDialog corpus (Li et al., 2017), which was developed specifically with the purpose of creating a high-quality multi-turn dialogue corpus for dialogue-modelling. DailyDialog is a manually labelled, multi-turn dataset of excellent quality. Ten topics are covered in total by the dialogues in the dataset, which follow standard dialogue flows like the Questions-Inform and Directives-Commissives bi-turn flows. Furthermore, DailyDialog has distinct multi-turn dialogue flow patterns that mirror actual human communication styles. It contains in total 13,118 dialogues and 103,632 utterances (Li et al., 2017). Besides having been designed specifically for dialogue-modelling, this corpus has also been used in previous studies (Klein et al., 2017; Zhong et al., 2019; He et al., 2021), which allows for results comparison. Although the data appears to be clean and less noisy compared to other datasets, it is also smaller, therefore less suited for more complex dialogue models.

In order to compare our work against existing literature, the proposed architectures were also tested on the Interactive Emotional Dyadic Motion Capture (IEMOCAP) database (Busso et al., 2008). This database is a widely used multimodal corpus designed for the study of human emotions in conversational settings. Developed by researchers at the University of Southern California, IEMOCAP contains approximately 12 hours of audiovisual data from dyadic sessions of male and female actors (Busso et al., 2008). The dataset includes video, speech, motion capture of face, and text transcriptions, making it a rich resource for multimodal emotion recognition and analysis.

IEMOCAP consists of five sessions, each featuring a different pair of actors engaging in both scripted and improvised scenarios designed to elicit specific emotional responses. The database covers a range of emotions, including happiness, anger, sadness, frustration, and neutral states. One of the unique aspects of IEMOCAP is its inclusion of motion capture data for facial expressions, which provides detailed information about facial muscle movements during emotional expressions (Busso et al., 2008). This feature makes IEMOCAP particularly valuable for researchers studying the relationship between facial expressions and emotional states in conversation.

The dataset has been extensively used in various research areas, including speech emotion recognition, multimodal sentiment analysis, and affective computing. Its comprehensive nature, combining audio, visual, and textual data, has made it a benchmark

dataset for developing and evaluating multimodal emotion recognition systems (Tripathi et al., 2019). However, it's worth noting that while IEMOCAP provides a rich source of emotional data, it is based on acted scenarios, which may not always perfectly reflect natural, spontaneous emotional expressions in real-world conversations.

Finally, leveraging the Multimodal EmotionLines Dataset (MELD) (Poria et al., 2019) to train a chatbot model represents a promising avenue for advancing conversational emotion recognition systems. MELD, an extension of the EmotionLines dataset, uniquely integrates audio, visual, and textual modalities, providing a comprehensive resource for understanding emotions in multi-party conversations. With over 1400 dialogues and 13000 utterances extracted from the Friends TV series, MELD incorporates a diverse range of speakers and encompasses seven labelled emotions—Anger, Disgust, Sadness, Joy, Neutral, Surprise, and Fear, along with sentiment annotations (positive, negative, and neutral) for each utterance.

The rationale behind utilising MELD lies in the growing significance of multimodal emotion recognition within the field of AI. This dataset is instrumental in addressing the limitations of existing resources like EmotionLines, which focus solely on text-based emotion recognition. MELD's multimodal nature facilitates a more nuanced understanding of emotional context within sequential turns of dialogues, thereby enhancing the accuracy of emotion recognition models. The availability of audio and visual data for each dialogue supports comprehensive context modelling, a crucial aspect for tackling challenges related to emotion change and flow in conversational sequences.

The creation process of MELD involved meticulous timestamp extraction from subtitle files, ensuring chronological order and episode cohesion for each dialogue. Subsequently, audiovisual clips corresponding to each utterance were extracted, resulting in a dataset enriched with visual, audio, and textual modalities. This extensive multimodal dataset not only addresses the shortcomings of existing dyadic datasets like IEMOCAP and SEMAINE, but also distinguishes itself by focusing on multi-party conversations.

To the best of the researchers' knowledge, no other study has used the MELD dataset for dialogue modelling tasks like response generation or dialogue management. This gap in the literature is significant because MELD offers unique characteristics that could advance dialogue systems development: its natural conversations from TV shows capture authentic human interaction patterns, complex turn-taking dynamics, and rich emotional context that are often missing in existing dialogue datasets. Most studies have utilised MELD primarily for the task of dialogue emotion recognition, given its focus on capturing emotional expressions across multi-party conversations. However, this unique dataset comprising natural multi-turn dialogues could potentially offer valuable avenues for exploring dialogue modelling research beyond just emotion recognition. The dataset's multi-party nature could help models learn more sophisticated dialogue management strategies that better handle multiple participants - a capability current dialogue systems often struggle with.

One promising direction would be to investigate the challenges and opportunities in leveraging MELD's multi-party conversational structure to develop dialogue models capable of coherent multi-participant interactions. Existing datasets often model just two-party conversations, but many real-world scenarios involve group discussions. Studying

approaches to effectively capture turn-taking dynamics, relationship understanding between participants, and maintaining consistent personality traits could push the boundaries of multi-party dialogue systems.

Furthermore, the emotional dimension of MELD's dialogues provides an intriguing test-bed for building affect-aware response generation models. Incorporating emotional context understanding could enable dialogue systems to respond more sensitively and naturally in emotionally-charged conversations compared to purely rational counterparts. Techniques for emotion recognition, emotional mimicry, and regulating the effect of generated responses are potential areas worth exploring.

Additionally, MELD's TV show transcript origin means the conversations exhibit heightened spontaneity, slang, and realistic disfluencies compared to typical constrained dialogue datasets. This naturalistic quality creates opportunities to develop robust dialogue models that can gracefully handle messy, colloquial language while maintaining coherence and relevance. Transfer learning from handling such naturalistic dialogues could boost real-world performance.

However, leveraging MELD may also pose challenges like navigating social implications of amplifying conversational biases or toxic language present in raw TV transcripts. Robust filtering and guidance techniques may be needed to develop applications aligned with social norms and values. Overall, thoughtful exploration of this unique multidimensional dialogue dataset could catalyse pioneering advances across multiple dialogue modelling research fronts. To the best of our knowledge, this is the first study that uses the MELD dataset to leverage audio transcripts and audio embeddings in order to train a multimodal chatbot.

A detailed comparison of each dataset’s specifications can be found in Table 1.

***Table 1**. Table comparing the different datasets*

	OpenSubtitles	**DailyDialog**	**Cornell**	**Meld**	**Iemocap**
Type of Content	Film subtitles	Dialogues for English learners	Raw film scripts	TV series dialogues	scripted and spontaneous sessions of dyadic conversations
Number of Utterances	441.5 M (2018 release)	103,632	304,713	13,708	10,039
Number of Tokens	3.2 G (2018 release)	17,812	48,177	10,643	2,171

Source	Entire database of the OpenSubtitles.org repository	Fictional conversations extracted from raw film scripts	Raw data crawled from various websites that provide content for English learners	extracted from the Friends TV series	original research

3.2.1 Data Cleaning and Preparation

The experiments conducted in this study relied on several publicly available dialogue datasets to ensure broad applicability and robust evaluation. The datasets were carefully curated to encompass diverse domains, conversation styles, and levels of contextual complexity. Preprocessing steps were applied to ensure data quality and compatibility with the transformer models.

Dataset Selection: Five benchmark dialogue datasets were selected — DailyDialog, OpenSubtitles, Cornell, IEMOCAP and Meld. These datasets cover scenarios ranging from daily conversations to open-domain chit-chat, providing a comprehensive test bed. Table 1. Presents the main features of the selected dataset for this study.
Data Cleaning: The raw dialogue transcripts underwent cleaning processes to remove irrelevant artefacts, handle encoding inconsistencies, and address missing or corrupted data points.
Conversation Segmentation: Each dataset was segmented into individual conversations, with adjacent utterances grouped to preserve the conversational flow and contextual information.
Train-Test Split: For each dataset, a random 89/10/1 split was applied to separate the data into training, development and testing subsets, ensuring fair and unbiased model evaluation.

The dataset partitioning strategy employed in this study utilized an 89/10/1 split ratio for training, development, and testing subsets, respectively. This particular split ratio was adopted following the approach established by Vaswani et al. (2017) in their seminal work on transformer architectures, where they demonstrated that allocating a larger portion of data to the training set can enhance model performance when working with large-scale dialogue datasets. The rationale behind this distribution stems from the observation that transformer models, being data-hungry architectures, benefit from maximizing the available training data while maintaining sufficient samples for validation and testing.
While the conventional 80/10/10 split is widely used in machine learning research, our modified ratio reflects the specific requirements of dialogue modelling tasks. The allocation of 89% to the training set ensures robust learning of conversational patterns and contextual dependencies, which is particularly crucial for transformer-based architectures. The 10% development set provides adequate data for model validation and hyperparameter tuning, while the 1% test set, containing approximately 4,400 samples (based on our max_samples parameter of 440,000), offers a sufficient number of examples for reliable performance evaluation.

This split ratio has also been employed by the original TensorFlow example used as a baseline. However, it is important to note that our choice of split ratio represents a trade-off between maximizing training data and maintaining adequate evaluation capabilities, and future researchers may wish to adjust these proportions based on their specific requirements and computational resources.

Prior to utilisation for training purposes, each corpus underwent meticulous preprocessing procedures adhering to standard Natural Language Processing (NLP) frameworks to ensure uniform format and high quality. This preprocessing stage is pivotal in preparing the data for subsequent modelling tasks, facilitating optimal performance and robustness of the neural network models. Subsequently, the processed corpora were partitioned into distinct training, validation, and test sets, adhering to best practices in machine learning experimentation to ensure reliable evaluation metrics and generalisation capabilities of the trained models.

The preprocessing pipeline encompassed a series of essential steps to cleanse and standardise the textual data. Firstly, each sentence was transformed to lowercase to maintain consistency in casing throughout the corpus. Additionally, a space was introduced between words and their subsequent punctuation marks, ensuring proper tokenization and syntactic parsing. Extraneous trailing spaces and non-alphabetic symbols or punctuation were removed to streamline the text further. Furthermore, contractions were expanded to their full forms to facilitate accurate interpretation and understanding by the neural network models. Other elements such as HTML tags, URLs, emojis, duplicate words, and excessive whitespace were also eliminated to maintain the integrity and coherence of the text data.
All the steps listed above have been implemented through automated pre-processing functions, to ensure a standardized pre-processing of all datasets.
Potential bias concerns commonly associated with dialogue modelling include prejudice towards different genders and races (Zhou et al., 2022; H. Liu et al., 2020). To mitigate this issue, references to the gender of the speaker (e.g. “male speaker one”) have been eliminated to feed the model with exclusively dialogue content. Although biased dialogue may still be present in one or more of the datasets used for the purpose of this study, the varied nature of the data and the different characteristics of the datasets should at least partially mitigate potential bias.

To facilitate efficient storage and retrieval of preprocessed data for subsequent experiments, each dataset was transformed into a structured JSON format. This format organised the data into lists of input sentences and corresponding output sentences, preserving the inherent associations between dialogue pairs. By encapsulating the preprocessed data in JSON format, the preprocessing steps need only be executed once, optimising computational efficiency and enabling seamless data retrieval for subsequent experiments and analyses.

During each experiment, the preprocessed dataset underwent tokenization and subsequent transformation into embeddings using the designated embedding method specified for the experiment. Three distinct embedding methods were employed to facilitate comparative analysis of model performance across different embedding strategies, thereby enabling insights into the relative efficacy and suitability of each method for the given task. This rigorous approach to preprocessing and embedding transformation laid the

groundwork for robust experimentation and evaluation of neural network models in multimodal dialogue modelling.

To ensure the comparability of experiments across datasets of varying sizes, a parameter called *max_samples* was introduced to limit the number of sentences used during the training process. This parameter serves to maintain consistency in the training data's magnitude, facilitating fair comparisons between different datasets. In the context of these experiments, the maximum number of samples was set to 440,000. This particular value was chosen as it aligns with the size of several key datasets used in the study.

The choice of 440,000 samples is significant as it represents a uniform scale across multiple datasets. Specifically, it corresponds to the entire MELD dataset, the entire Cornell dataset, the entirety of the Endure DailyDialog dataset, and approximately 0.1% of the OpenSubtitles dataset. By employing this consistent sampling approach, researchers can effectively control for dataset size discrepancies, ensuring that each model is trained on a comparable amount of data. This not only facilitates fair evaluations but also enhances the reliability and validity of the experimental results.

Moreover, limiting the training data to a standardised number of samples enables researchers to focus on the quality rather than the quantity of the dataset. This approach ensures that each model is exposed to a representative subset of the available data, allowing for meaningful comparisons of their performance across different datasets. By standardising the training data size in this manner, researchers can uncover insights into how various transformer architectures perform under similar conditions, paving the way for more robust and informative conclusions in the field of dialogue modelling.

3.2.2 Extracting Audio Embeddings

To ensure systematic testing of the different architectures using different contextual embeddings, they were tested on audio information only, before integrating audio and text embedding in a multimodal dialogue model. The aforementioned architectures were therefore tested on audio embeddings alone, creating audio unimodal architectures. In order to do so, audio embeddings were extracted from the MELD dataset video files.

To extract the audio embeddings from the video, a specific function called extract_embedding was created. The provided function extract_embedding is designed to extract audio embeddings from video files in the MELD dataset. Here's what the function does:

1. Checking for Cached Embeddings: The function first checks if the audio embeddings for a specific dialogue and phrase have already been computed and cached. It constructs a cache file name based on the dialogue and phrase indices, and checks if the file exists.
2. Loading and Processing the Video: If the cache file doesn't exist, the function loads the corresponding video file (in MP4 format) from the specified directory using the AudioSegment library. It then performs several audio processing steps, including setting the number of channels to 1 (mono), setting the frame rate to 16000 Hz, and setting the sample width to 2 bytes.

3. Exporting the Audio: The processed audio is exported as a WAV file with the same name as the original video file, but with a different extension.
4. Extracting Audio Embeddings: The function reads the WAV file using the sound file library and passes the audio data and sample rate to the openl3 library's get_audio_embedding function. This function calculates audio embeddings, which are compact representations of the audio signal that capture relevant features for downstream tasks. The function specifies the “env” content type, “linear” input representation, and an embedding size of 512.
5. Padding and Saving Embeddings: The extracted audio embeddings are padded to a fixed length (*max_len*) using the pad function, which is likely a custom function defined elsewhere. The padded embeddings are then converted to a PyTorch tensor and saved to the cache file using torch.save.
6. Appending Embeddings: If the cache file exists, the function loads the embeddings from the cache file and appends them to a list called elements.
7. Returning Embeddings: Finally, the function returns the list of audio embeddings (elements).

In summary, the extract_embedding function is responsible for loading video files from the MELD dataset, processing the audio, extracting audio embeddings using the openl3 library, caching the embeddings for future use, and returning the embeddings for a given dialogue and phrase. The caching mechanism is implemented to avoid redundant computations and improve efficiency when working with the same set of dialogues and phrases multiple times.

3.3 Multimodal Dialogue modelling

Multimodal chatbots, capable of processing and generating responses across multiple modalities such as text, images, and audio, represent a significant advancement in dialogue modelling. Incorporating multimodal capabilities into chatbots enhances their ability to understand and generate responses that are not only contextually relevant but also rich in sensory information, mirroring the way humans communicate. As articulated by (Sun et al., 2022), integrating multimodal inputs allows chatbots to leverage the complementary nature of different modalities, leading to more nuanced and engaging interactions with users. By interpreting a diverse range of input signals, multimodal chatbots can infer user intent more accurately, leading to more personalised and contextually appropriate responses.

One key characteristic of multimodal chatbots is their ability to process and synthesise information from diverse sources, including text, images, and audio. This versatility enables chatbots to engage users in more immersive and interactive conversations, catering to a wide range of communication preferences and styles. According to (Sundar & Heck, 2022; Zhang et al., 2020), multimodal chatbots leverage advanced machine learning techniques such as deep learning and multimodal fusion to seamlessly integrate information from different modalities, resulting in a more comprehensive understanding of user queries and preferences. By synthesising information from multiple modalities, multimodal chatbots can provide richer and more informative responses, enhancing the overall user experience and satisfaction.

Furthermore, multimodal chatbots exhibit adaptability and robustness in handling complex dialogue scenarios. Unlike traditional chatbots that rely solely on text inputs, multimodal chatbots can leverage additional contextual cues from images, audio, or other sensory inputs to enhance their understanding of user intent and context. This adaptability is crucial in real-world applications where users may communicate using a variety of modalities or in noisy environments. Multimodal chatbots equipped with robust multimodal fusion mechanisms can effectively integrate information from different modalities, ensuring consistent and accurate responses across diverse dialogue scenarios. By harnessing the power of multimodal inputs, chatbots can achieve higher levels of contextual understanding and adaptability, paving the way for more natural and intuitive human-computer interactions. As far as the researchers are aware, this study represents a pioneering effort in utilising the MELD dataset to harness the power of audio transcripts and audio embeddings for training a multimodal conversational agent.

Recognizing the importance of incorporating diverse modalities into dialogue systems, the study embarked on a journey to investigate the potential of multimodal approaches in enhancing dialogue understanding and generation capabilities. To facilitate this exploration, this research exploited the Multimodal dataset MELD (Multimodal EmotionLines Dataset), a rich and diverse corpus meticulously curated to encompass textual, visual, and acoustic modalities. By leveraging the MELD dataset, this research sought to unravel the intricacies of multimodal communication and elucidate how integrating multiple modalities can enrich the dialogue modelling process. Through systematic experimentation and analysis on the MELD dataset, the study aimed to uncover insights that could drive advancements in multimodal dialogue systems and pave the way for more natural and immersive human-computer interactions.

In order to use the Multimodal dataset MELD, a specific class has been defined.
The provided Python class, named MeldMp4, is designed for processing audio and text data from the MELD dataset. Here's a description of its main functionalities:
The class constructor __init__ takes several parameters, including directory (the path to the dataset), max_samples (the maximum number of samples to load), max_words (maximum number of words per sample), and cache (a boolean flag indicating whether to use caching).
The load method is responsible for loading the data. It checks if caching is enabled and either loads the data from cache or reads it from files, processes the data, and returns two lists: questions and answers.
The open_file method reads the MELD dataset files, extracts information about dialogues and phrases, and processes the audio files to obtain text representations.
The class handles caching by storing processed data in JSON files based on a hash generated from the specified parameters (max_samples and max_words). This helps speed up the loading process for subsequent runs with the same parameters.
The extract_sub_title and related methods are responsible for converting audio from video files into text using different approaches, such as using Google Speech Recognition or the Whisper ASR (Automatic Speech Recognition) model.
Overall, the class is designed for efficient loading and processing of audio and text data from the MELD dataset, with an emphasis on handling large datasets and incorporating caching mechanisms.

To extract the audio embeddings from the video files of the MELD dataset, the steps outlined in section 3.2.2 were implemented.

This class used Whisper ASR to extract textual data from the original MP4 files. Whisper automatic speech recognition (ASR) is an innovative technology designed to transcribe speech accurately and efficiently. Developed by researchers at OpenAI, Whisper represents an ASR system trained on an extensive corpus of 680,000 hours of multilingual and multitask supervised data sourced from online sources. This large and diverse dataset underpins Whisper's enhanced robustness to various challenges, including accents, background noise, and technical language nuances. Additionally, Whisper's expansive dataset facilitates transcription in multiple languages and supports translation from those languages into English. The Whisper architecture adopts a simple yet effective end-to-end approach, utilising an Encoder-Decoder Transformer. Input audio undergoes segmentation into 30-second chunks, conversion into log-Mel spectrograms, and subsequent processing through an encoder. The decoder is trained to generate corresponding text captions, augmented with special tokens directing the model to perform tasks such as language identification, phrase-level timestamps, multilingual speech transcription, and translation to English. Unlike existing approaches that often rely on smaller, more tightly aligned audio-text datasets or unsupervised audio pretraining, Whisper benefits from its broad and diverse dataset without being fine-tuned to any specific domain. Consequently, while it may not outperform specialised models on benchmarks like LibriSpeech, Whisper demonstrates superior zero-shot performance across diverse datasets, exhibiting 50% fewer errors. Notably, Whisper's dataset includes a substantial portion of non-English audio, allowing it to alternate between transcribing in the original language or translating to English. This approach proves particularly effective in learning speech-to-text translation and surpasses the state-of-the-art supervised models in zero-shot translation performance on CoVoST2 to English.

Nevertheless, Automatic Speech recognition still poses some of the aforementioned challenges (Gong et al., 2023). These challenges include variations in speech patterns and accents, background noise interference, speech overlap, and speaker diarization errors. Variations in speech patterns and accents can hinder the accuracy of ASR systems, particularly when encountering diverse linguistic contexts or non-standard pronunciations. Background noise interference, such as environmental sounds or overlapping conversations, can obscure speech signals, leading to misinterpretations by ASR systems. Speech overlap poses a significant challenge, especially in group discussions or interviews, where multiple speakers may talk simultaneously, resulting in difficulty separating individual speech segments. Additionally, speaker diarization errors, where ASR systems incorrectly attribute speech segments to different speakers, can further complicate the transcription process (T. Chen et al., 2020; Jurafsky & Martin, 2008). These challenges proved particularly difficult to overcome when transcribing audio data recorded from a TV show such as Friends, where one can easily encounter background noise, different accents, soundtracks and other audio tracks (such as a laughing track in the background), and most importantly overlapping voices or false starts to the conversation. All of these issues contributed to somewhat degrading the quality of the textual data extracted from the original Mp4 files.

3.3.1 Incorporating Audio Embeddings

The Audio-Transformer model represents a novel approach within the realm of dialogue modelling, leveraging both textual and audio information to enhance word representations. This architecture builds upon the traditional encoder-decoder transformer framework, a widely utilised structure in sequence-to-sequence learning tasks. In this modified framework, each word representation is enriched with both the traditional word embedding (*wn*) and a word-level audio representation *(an)*, resulting in a combined representation *[wn; an]*. The integration of audio features into the word representation enables the model to capture additional contextual information from the audio input, thereby enhancing its ability to understand and generate natural-sounding responses.

One of the potential advantages of the Audio-Transformer model lies in its ability to leverage multimodal information, incorporating both textual and audio features into the dialogue modelling process. By integrating audio embeddings derived from the audio input alongside traditional word embeddings, the model gains access to richer and more diverse contextual cues, which can potentially lead to more accurate and contextually relevant responses. This could prove particularly beneficial in scenarios where audio cues provide valuable context or additional information that may not be captured through text alone. The potential advantage of this architecture is, therefore, its ability to capture nuances and contextual information that may be conveyed through the audio modality but not fully captured by the text alone. For example, in emotion recognition or sentiment analysis tasks, the tone, pitch, and other acoustic features can provide valuable insights beyond the literal text. By explicitly incorporating these audio features into the word representations, the model may better understand the underlying emotions, sentiments, or intentions expressed in the input.

Integrating textual and audio modalities through embeddings can offer several advantages for dialogue modelling systems. Multimodal representations that combine language and acoustic cues have the potential to capture more comprehensive conversational context compared to text-only approaches. Audio embeddings encapsulate paralinguistic features such as tone, emotion, and vocal emphasis, which are often lost when solely relying on textual inputs (Arevalo et al., 2020). This richer multimodal representation can lead to improved understanding and generation of more natural, contextually appropriate responses (Tsai et al., 2019).

However, there are also potential challenges and considerations with this approach. Aligning and synchronising the audio features with the corresponding text can be a non-trivial task, especially in cases where the audio and text modalities are not perfectly aligned or contain noise or errors. Additionally, the quality and effectiveness of the audio representations obtained from the openl3 library may vary depending on the audio characteristics and the specific task at hand.

Another potential concern is the increased computational complexity and memory requirements introduced by incorporating additional audio features into the model. This could make training and inference more resource-intensive, particularly for large datasets or models with high dimensionality. Furthermore, the effectiveness of the Audio-Transformer model may depend on the quality and reliability of the audio embeddings

extracted from the input audio data, highlighting the importance of robust audio processing techniques.

From a computational perspective, processing both textual and audio inputs increases the complexity and resource requirements of the model, leading to longer training times and higher computational demands (Tsai et al., 2019). This can be a concern for resource-constrained environments or real-time applications.
Moreover, obtaining high-quality multimodal dialogue datasets with aligned text and audio data can be challenging and resource-intensive. Data collection and annotation processes for multimodal data are often more complex and time-consuming compared to text-only datasets (Poria et al., 2019). Effectively fusing textual and audio embeddings in a meaningful way can also be a non-trivial task, as different modalities may have varying levels of importance or relevance depending on the dialogue context (Gu et al., 2023).
Additionally, audio data can be susceptible to various types of noise, such as background noise, overlapping speech, or recording artefacts. Preprocessing audio signals and handling noise effectively can require additional domain-specific techniques (W. Zhao et al., 2019). Despite these challenges, the potential benefits of multimodal dialogue models, such as improved understanding and generation of more natural and contextually appropriate responses, make them an active area of research. Our research presents a novel approach to training a multimodal chatbot by integrating audio transcripts and audio embeddings from the MELD dataset, an endeavour that has not been undertaken before, to the best of our knowledge.

Despite these challenges, the Audio-Transformer model represents an interesting step towards multimodal learning and may pave the way for more sophisticated architectures that can effectively integrate and reason over multiple modalities. Its performance and applicability would ultimately depend on the specific task, dataset characteristics, and the trade-offs between model complexity and potential performance gains.

Figure 3.1 presents a simplified illustration of the Audio-Transformer architecture, highlighting the integration of audio features into the word representations:

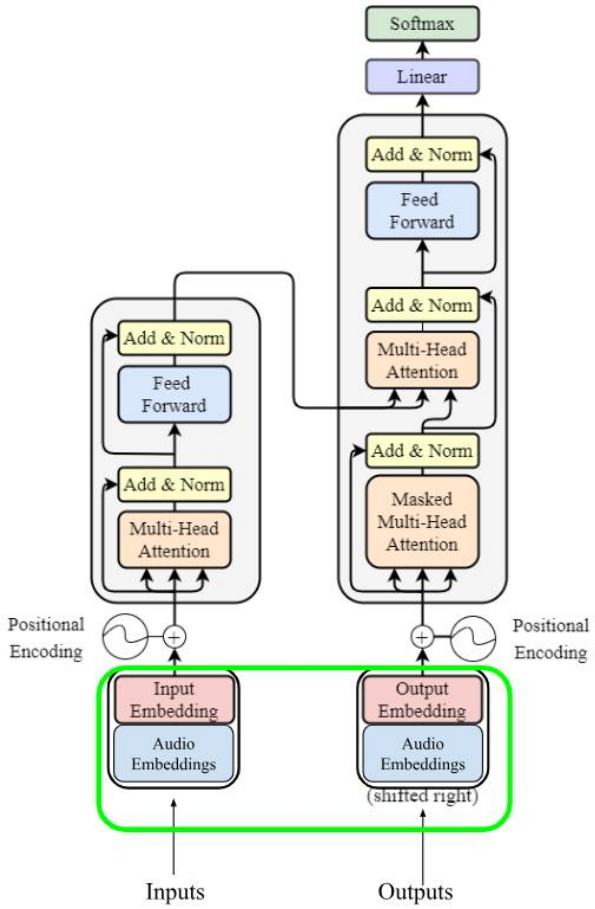

Figure 3.1. A Transformer architecture modified to receive audio embeddings along text embeddings as inputs in both the encoder and the decoder stack.

In this diagram, the word embeddings and audio embeddings are concatenated to form the input word representations to the transformer encoder. The encoder then processes these multimodal representations, potentially capturing interactions between the textual and acoustic information. The resulting encoded representations are then passed to the decoder for sequence generation or other downstream tasks.

In summary, the Audio-Transformer model represents an innovative approach to dialogue modelling by incorporating audio information into the word representations. While offering the potential for enhanced context understanding and more natural conversation generation, this architecture also presents challenges related to computational complexity and data integration. Further exploration and experimentation are needed to fully assess the effectiveness and practical applicability of the Audio-Transformer model in dialogue modelling tasks.

3.4 Training Procedure

To train the chatbot models, a systematic and rigorous procedure was followed, encompassing data preparation, model training, and evaluation. Each step was meticulously designed to ensure reproducibility and validity of results across experiments.

Before commencing training, each corpus underwent a comprehensive preprocessing phase to ensure consistency and quality of the text data. This included standard NLP techniques such as lowercasing, punctuation handling, removal of contractions, elimination of HTML tags, URLs, emojis, duplicate words, and multiple spaces, among others. The datasets were then divided into training, validation, and test sets to facilitate model training and evaluation. The processed text was transformed into JSON format, comprising lists of input and output sentences, streamlining data retrieval for subsequent experiments.

For each experiment, tokenization was applied to the dataset, followed by embedding transformation using one of the three designated embedding methods: automated embedding matrix generated from the data, GloVe embeddings, or BERT embeddings. This multistep preprocessing ensured that the input data was properly formatted and ready for training across different embedding techniques.

For the Multimodal EmotionLines Dataset (MELD), a specialised class named MeldMp4 was defined to handle audio and text data processing. This class, designed with efficiency in mind, facilitated loading and processing of audiovisual data from MELD, incorporating caching mechanisms to optimise data retrieval.

The training phase involved deploying four distinct transformer architectures: Baseline Transformer, Encoder-Decoder Transformer, Extractor Architecture, and Reencoder Architecture. Each architecture was trained using the aforementioned datasets and embedding methods, resulting in a comprehensive evaluation of model performance across different contexts.

The Baseline Transformer and Encoder-Decoder Transformer architectures followed the original transformer architecture proposed in “Attention is All You Need” (Vaswani et al., 2017), with variations in the embedding layer. The Reencoder Architecture introduced a structural modification to the encoder, incorporating embeddings from previous utterances to inform the current turn's embeddings. Meanwhile, the Extractor Architecture, inspired by (Riley et al., 2021), extracted contextual information from preceding sentences in the conversation to enhance model understanding.

Evaluation Metrics:

To evaluate model performance, a set of standard evaluation metrics was employed, focusing on accuracy and speed. Accuracy metrics included BLEU, METEOR, TER and Perplexity, traditionally used for machine translation tasks. These metrics were selected due to their applicability to assessing the textual entailment of chatbot outputs, despite their origin in translation evaluation. Speed, measured in seconds elapsed between user input and chatbot response, was also considered an essential metric, reflecting the efficiency of the conversational system.

Computational Resources:

The experiments were conducted using Google Colab runtimes, leveraging T4 GPUs for accelerated model training. Despite the resource constraints, strategic management of memory and runtime durations ensured efficient experimentation. For longer-running experiments exceeding the 24-hour limit on Colab, a virtual machine on the Google Cloud Platform with similar computational capabilities was utilised. Detailed specifications of the

T4 GPU, including CUDA cores, memory capacity, and software support, were documented to provide transparency and reproducibility.

Overall, the training procedure was meticulously designed and executed to facilitate rigorous experimentation and comprehensive evaluation of the chatbot models across different architectures and datasets. The utilisation of standardised evaluation metrics and computational resources ensured robustness and reliability of the research findings.

3.4.1 Embedding Layers in Large Language Models

Embedding layers play a pivotal role within language models, serving as a bridge between raw textual data and the neural network's computational framework. These layers are tasked with transforming discrete tokens or words into dense, continuous vector representations, effectively capturing semantic and contextual information encoded within the text. By encoding words as dense vectors in a continuous vector space, embedding layers enable language models to learn meaningful representations of words and their relationships within the context of the given task. This process of embedding enables the model to operate in a continuous, high-dimensional vector space, facilitating efficient computation and effective learning of complex linguistic patterns.
The quality and efficacy of embedding layers have a profound impact on the performance of language models across various natural language processing tasks. Well-designed embedding layers are capable of capturing nuanced semantic relationships between words, enabling the model to discern subtle distinctions in meaning and context. By learning dense representations that preserve semantic similarity and syntactic relationships, embedding layers empower language models to generalise more effectively to unseen data and tasks. Consequently, embedding layers that effectively capture the underlying semantics of the text contribute to improved model performance, leading to enhanced accuracy and robustness in language understanding and generation tasks.

Embedding layers serve therefore as a foundational component in large language models (LLMs), facilitating the transformation of discrete textual inputs into continuous vector representations. These representations capture semantic relationships and contextual information critical for downstream NLP tasks. This chapter delves into the intricate workings of embedding layers within LMs, elucidating their role in encoding textual information and fostering semantic understanding. Through a comprehensive examination of underlying mechanisms and architectural considerations, this chapter aims to provide a nuanced understanding of embedding layers' functionality in the context of large-scale language modelling.

3.4.1.1 Fundamentals of Embedding Layers

At the core of embedding layers lies the concept of distributed representations, wherein words or tokens are encoded as dense, low-dimensional vectors in continuous vector spaces. This departure from traditional one-hot encoding enables LMs to capture semantic similarities and contextual nuances, enhancing their capacity to discern meaning from raw textual inputs.

3.4.1.2 Embedding Initialization

The process of initialising embedding layers is crucial for determining the initial state of word representations within the model. Pre-trained embeddings, such as Word2Vec (Mikolov et al., 2013), GloVe (Global Vectors for Word Representation) (Pennington et al., 2014), or FastText (Bojanowski et al., 2017), offer a valuable starting point by leveraging large corpora to generate contextually rich embeddings. Alternatively, embedding layers can be initialised randomly and fine-tuned during the training process to adapt to the specific task at hand (Devlin et al., 2019).

3.4.1.3 Training Dynamics

During the training phase, embedding layers undergo iterative optimization alongside other components of the LM architecture. Through backpropagation and gradient descent algorithms, the embedding vectors are adjusted to minimise the discrepancy between predicted and actual outcomes, thereby refining their semantic representations (Hinton et al., 2012).

3.4.1.4 Semantic Similarity and Contextual Understanding

One of the primary functions of embedding layers is to facilitate the computation of semantic similarity between words or tokens. By encoding textual inputs into continuous vector representations, embedding layers enable large language models (LMs) to capture semantic relationships and contextual nuances inherent in language. Techniques such as cosine similarity or Euclidean distance metrics are commonly employed to quantify the similarity between embedding vectors, enabling tasks such as word similarity detection or sentiment analysis.

3.4.1.5 Dimensionality and Vector Space Properties

The dimensionality of embedding vectors plays a crucial role in determining the expressive capacity and computational efficiency of embedding layers. While higher-dimensional embeddings offer finer-grained representations, they also entail increased computational overhead (Turian et al., 2010). Conversely, lower-dimensional embeddings may sacrifice some granularity but are more computationally tractable, striking a balance between representation quality and computational efficiency (Sutskever et al., 2014). Moreover, the choice of vector space properties, such as sparsity or density, influences the model's ability to capture semantic relationships effectively.

3.4.1.6 Contextualised Embeddings

In recent years, contextualised embeddings have emerged as a powerful extension of traditional static embeddings, offering dynamic representations that adapt to the surrounding context. Contextualised embeddings, such as those used in models like ELMo (M. E. Peters et al., 2018), BERT (Devlin et al., 2019), and GPT (Radford et al., 2019), provide dynamic representations that adapt to the surrounding context. Models such as ELMo, BERT, and GPT leverage contextualised embeddings to capture intricate semantic nuances and syntactic structures present in natural language. By incorporating contextual information from surrounding tokens, these models enhance their ability to comprehend and generate coherent textual outputs across diverse NLP tasks.

3.4.2 Tokenization

Tokenization is a fundamental preprocessing step in natural language processing (NLP) tasks, influencing the performance and effectiveness of LMs. This dissertation thesis investigates and compares the characteristics and effects of various tokenization techniques, including tfds.deprecated.text.SubwordTextEncoder, a custom-made class used to tokenize text to leverage GloVe, and BERT Tokenizer, on the embedding layer of large LMs. Through empirical analysis and evaluation, this study aims to provide insights into the strengths, weaknesses, and potential applications of each tokenization method in the context of LM development. This should provide a comparative analysis of tokenization techniques and their impact on Large Language Models

Tokenization serves as the initial step in processing raw text data, dividing input sequences into individual tokens or subword units. The choice of tokenization technique can significantly impact the subsequent stages of NLP tasks, particularly in the context of large LMs. This chapter explores prominent tokenization methods: tfds.deprecated.text.Tokenizer, tfds.deprecated.text.SubwordTextEncoder, the tokenizer class created to leverage GloVe embeddings, and BERT Tokenizer, analysing their respective features and examining their influence on the embedding layer of large LMs.

3.4.2.1 TensorFlow's Tokenizer

The tfds.deprecated.text.Tokenizer, a part of the TensorFlow Datasets library, is a simple tokenization approach based on whitespace and punctuation splitting. It segments input text into individual words or tokens, treating each word as a distinct entity. While straightforward and efficient, this tokenizer may struggle with handling out-of-vocabulary words and morphologically rich languages (*Tfds.Deprecated.Text.Tokenizer | TensorFlow Datasets*, n.d.).

3.4.2.2 TensorFlow's Subword Text Encoder

The tfds.deprecated.text.SubwordTextEncoder offers a more sophisticated tokenization technique by leveraging subword units to encode text sequences. It employs byte pair encoding (BPE) or similar algorithms to segment words into subword units, enabling the representation of rare or unseen words through compositionality. This approach enhances the robustness of tokenization, particularly in multilingual and morphologically complex settings (Sennrich et al., 2016).

3.4.2.3 GloVe (6b)

GloVe is an unsupervised learning algorithm for obtaining word representations. Unlike traditional embedding techniques, GloVe operates at the word level, generating dense vector embeddings based on global co-occurrence statistics. These embeddings capture semantic relationships and contextual information, facilitating effective representation learning for downstream NLP tasks (Pennington et al., 2014).
In order to leverage GloVe embeddings, a specific tokenizer class has been created for the purpose of this research.
This specific tokenizer class encapsulates the implementation of tokenization algorithms utilising the keras.preprocessing.text Tokenizer. This class offers two key methods: "build" and "tokenize_and_filter." The "build" method initialises the tokenizer by fitting it to the

untokenized data, which comprises questions and answers concatenated together. The tokenizer is constructed using the Tokenizer module, with an out-of-vocabulary token specified as 'OOV' to handle unknown words. Additionally, the start and end tokens are defined to mark the beginning and end of sentences. Upon tokenization, the vocabulary size is determined, considering the start and end tokens alongside the word index. The “tokenize_and_filter” method tokenizes the inputs and outputs, ensuring they do not exceed the specified maximum length. Each sentence is prepended with the start token and appended with the end token before being checked against the maximum length criterion. Tokenized inputs and outputs are then padded to the maximum length using keras.preprocessing.sequence.pad_sequences. Through these functionalities, the Algo2 class facilitates the tokenization and filtering of text data for GloVe embeddings, ensuring compatibility with large language models while handling sequence length constraints.

One of the remarkable features of GloVe embeddings lies in their dimensionality, which can vary based on the specific requirements of the task at hand.
GloVe embeddings are available in different dimensions, including 100, 200, and 300 dimensions, among others. Each dimension corresponds to a distinct aspect or feature of the word's semantics, with higher-dimensional embeddings offering more nuanced representations. For instance, GloVe embeddings with 100 dimensions provide a relatively compact representation, capturing essential semantic information while minimising computational overhead. In contrast, embeddings with 300 dimensions offer richer representations, enabling finer-grained distinctions between word meanings and contexts.

The choice of GloVe embedding dimensionality depends on various factors, such as the complexity of the language being analysed, the size of the vocabulary, and the computational resources available. For tasks involving simpler languages or smaller datasets, lower-dimensional GloVe embeddings may suffice, providing a balance between representation quality and computational efficiency. Conversely, tasks requiring greater semantic granularity or handling larger vocabularies may benefit from higher-dimensional embeddings.

In practice, researchers and practitioners often experiment with different GloVe embedding dimensionalities to optimise model performance for specific tasks. By fine-tuning the embedding dimensionality based on task requirements and resource constraints, GloVe embeddings can significantly enhance the effectiveness and efficiency of large language models across various natural language processing applications.

3.4.2.4 BERT Tokenizer

The BERT (Bidirectional Encoder Representations from Transformers) Tokenizer is specifically designed for use with transformer-based models like BERT. It employs WordPiece tokenization, breaking down input text into subword units based on a pre-defined vocabulary. By considering both left and right context during tokenization, BERT Tokenizer captures bidirectional contextual information, enhancing the model's understanding of language semantics (Devlin et al., 2019). This technique dissects input text into subword units based on a pre-defined vocabulary, thereby accommodating a wide range of linguistic variations and domain-specific terminologies. Unlike conventional tokenizers that treat each word as a discrete entity, BERT Tokenizer captures the inherent

compositional structure of language, enabling the representation of morphologically rich words and out-of-vocabulary terms through subword composition.

Furthermore, what sets BERT Tokenizer apart is its emphasis on bidirectional context. By considering both left and right context during tokenization, BERT Tokenizer captures a comprehensive view of linguistic relationships, facilitating a deeper understanding of language semantics. This bidirectional approach enhances the model's ability to discern intricate nuances and contextual dependencies, thereby fostering more robust representations in the embedding layer.

BERT Tokenizer's integration with transformer-based architectures aligns seamlessly with BERT's core principles of bidirectional learning and contextual embedding generation. As a result, the embeddings derived from BERT Tokenizer exhibit a heightened sensitivity to contextual cues and linguistic subtleties, empowering downstream tasks with enhanced semantic understanding and predictive accuracy. This capability has propelled BERT to the forefront of NLP research and applications, revolutionising a myriad of tasks ranging from language understanding and sentiment analysis to question answering and text summarization.

In essence, BERT Tokenizer stands as a testament to the transformative power of advanced tokenization techniques in shaping the capabilities of large language models. Its ability to capture bidirectional context and leverage subword composition renders it indispensable for tasks requiring nuanced language understanding and contextual reasoning. As the field of NLP continues to evolve, BERT Tokenizer's influence is poised to endure, paving the way for further advancements in language representation learning and natural language understanding.

***Table 2**. Table comparing the different tokenizers used for experimentation.*

	TensorFlow's Subword Text Encoder	**GloVe 6b**	**Bert**
Strategy	Subword tokenization	Word tokenization	Subword tokenization
Static/Dynamic	Dynamic	Static	Static
Vocabulary	Depends on data	400000	28996
Cased/Uncased	Cased	Uncased	Cased
Embedding dimension	256	50, 100, 200, 300	768

3.5 Evaluation Metrics

An evaluation model will also be put in place, in order to evaluate the accuracy coherence of the model output. Accuracy is important in terms of context awareness as a metrics for

the ability of the system to provide intelligent and sensible answers to the user's input. Dialogue systems evaluation has proven to be a difficult task because human conversation focuses on different purposes and objectives. The metrics employed to assess the conversation can vary based on the chatbot's objectives. The effectiveness of the interaction—that is, whether the chatbot accomplished the task the user requested—will be the primary factor used to evaluate a personal assistant chatbot. A companion chatbot will be judged on its capacity to maintain the conversation and engage users, as opposed to whether the exchange was efficient. A chatbot can be assessed using two primary methods: automated evaluation metrics and human evaluation.

Human evaluation remains a fundamental approach in assessing chatbot performance, with numerous studies emphasizing its importance in measuring conversational accuracy and quality (Christensen et al., 2018; Sordoni et al., 2015a). This methodology typically involves participant engagement with the chatbot system, followed by structured assessment through questionnaires or evaluation frameworks. These assessments commonly employ rating scales to measure multiple performance dimensions, including effectiveness, efficiency, and user satisfaction (Radziwill & Benton, 2019). Despite its prevalence in research, human evaluation presents several significant challenges. The method requires substantial resource allocation, making it expensive and time-intensive to implement. Additionally, scaling such evaluations proves difficult, and the inherent subjectivity of human judgment can introduce bias, as different evaluators may rate identical interactions differently, even when following standardized assessment frameworks.

Nevertheless, human evaluation offers distinct advantages that justify its continued use in chatbot assessment research. Its primary strength lies in the ability to assess multiple dimensions of conversational quality simultaneously, providing comprehensive insights into the nuances of human-chatbot interactions. Furthermore, evaluation frameworks can be tailored to align with specific chatbot objectives and features, offering flexibility in assessment criteria. These advantages have led to the widespread adoption of human evaluation metrics in prominent studies, including the work of (Sordoni et al., 2015b) and (Christensen et al., 2018), establishing it as a valuable methodology in dialogue system research despite its limitations. However, given the lack of time and resources, this piece of research will use automated evaluation metrics.

In terms of the amount of time and resources required to complete the evaluation, automated evaluation metrics are more effective. However, it seems that industry standards are still lacking when it comes to the application of evaluation metrics, and automated metrics don't seem to be able to accurately gauge the overall effectiveness, efficiency, and quality of the conversation. Nonetheless, these metrics are still frequently used to assess chatbots because they are easier to use. The evaluation metrics used to measure accuracy will be standard evaluation metrics used for Machine Translation and other Natural Language Processing tasks such as BLEU, METEOR, TER and Perplexity, as they have been used by Saikh et al., (2018) and by Sordoni et al., (2015). Although these evaluation metrics are considered to be more suitable for Machine Translation problems, they can still provide valuable information regarding the Textual Entailment of the chatbot output (Saikh et al., 2018).

The evaluation metrics used to measure accuracy will be standard evaluation metrics used for Machine Translation and other Natural Language Processing tasks, such as BLEU (bilingual evaluation understudy); METEOR (Metric for Evaluation of Translation with

Explicit ORdering) and TER (Translation Error Rate). Although these evaluation metrics are considered to be more suitable for Machine Translation issues, they can still provide valuable information regarding the Textual Entailment of the chatbot output (Saikh et al., 2018).

These metrics offer invaluable insights for developers and researchers engaged in MT technology development, as they facilitate frequent evaluations of MT systems with minimal human intervention. The primary advantages of automated MT quality metrics lie in their speed, ease of execution, and low human labour requirements, making them highly conducive to iterative system development processes. Moreover, these metrics obviate the need for bilingual speakers and can be repeatedly applied throughout the system development life cycle.

However, it is crucial to acknowledge the inherent limitations of MT metrics, particularly in the context of translation production scenarios. While these metrics are adept at assessing the quality of MT models, their utility in evaluating translation output in real-time production environments is limited. Several notable limitations include the necessity for a reference translation, which is often impractical to obtain in live translation scenarios, and the assumption that the reference translation represents a gold standard, which may not always be verifiable due to the inherent variability in translations. Additionally, the automated quality scores generated by these metrics may not directly translate to actionable insights for translators, as they do not provide information regarding post-editing time or compensation requirements.

The decision to incorporate automated metrics into an MT program hinges on the specific use case and objectives. If deemed appropriate, it is imperative to train the metrics on relevant data and prepare reference translations for each segment to be scored.
Most automated metrics employ a segment-level similarity-based approach, comparing machine-translated segments to human-generated reference translations. This comparison typically involves assessing the closeness of the machine-translated output to the reference translation, with smaller differences indicative of higher quality. While word-level comparisons are common, metrics also utilise n-grams to compute precision scores, where n-grams represent contiguous sequences of items in a text or speech sample, such as phonemes, syllables, letters, or words. Understanding these underlying methodologies is crucial for interpreting and utilising automated MT quality metrics effectively in machine learning research and development endeavours.
However, in comparison to other NLP tasks, all these n-gram based evaluation models seem less suited to assess dialogue systems, as two responses may be equally effective in responding to a given message even though they do not share any overlapping n-grams. Nonetheless, since they are efficient, widely used in research, and easily reproducible, these metrics have been selected for the evaluation of the different architectures.

To ensure comprehensive tracking of the evaluation results and facilitate efficient analysis, Weights and Biases logging functions and callbacks were integrated into the experiments. Through this logging mechanism, the evaluation metrics were systematically recorded and stored alongside each experiment's configurations and outcomes. This approach not only provided real-time insights into the model's performance but also enabled us to monitor the progress of the training process and make informed decisions regarding model adjustments

or hyperparameter tuning. Leveraging Wights and Biases logging functions and callbacks, ensured the transparency, reproducibility, and rigour of the experimental methodology, laying the foundation for robust conclusions and insights derived from our research efforts.

3.5.1 BLEU

BLEU was initially developed to measure machine translation outputs, but it is now widely used to evaluate a variety of NLP tasks. A translation's value is assigned by the BLEU metric on a scale from 0 to 1, although it is usually expressed as a percentage. The more a translation resembles a human translation, the closer it is to 1. Put simply, sequential words receive a higher score in the BLEU metric (KantanMT - Cloud-based Machine Translation Platform), which counts the number of words that overlap in a translation when compared to a reference translation. BLEU scores were employed by a number of authors, including Dhyani & Kumar, (2020), Saikh et al., (2018), Vaswani et al., (2017), and Sordoni et al., (2015), to assess chatbots and other NLP tasks. However, despite its widespread usage, BLEU has been subject to criticism due to several inherent limitations. One of the primary issues lies in its fixed brevity penalty, which aims to penalise shorter translations to compensate for the lack of recall. Critics argue that this penalty is insufficient and may not effectively address the recall deficiency, potentially leading to biased evaluations. Additionally, BLEU relies on higher-order N-grams as proxies for a translation's grammatical correctness. While N-gram counts provide some insight into fluency, they may not fully capture the nuances of syntax and grammar. Some researchers advocate for a more direct measure of word order and grammaticality, believing that this approach would better align with human judgments of translation quality and improve the metric's correlation with human assessments.

Moreover, the reliance on N-gram counts in BLEU does not mandate precise word-to-word matching, which can lead to inaccuracies in evaluating translations, especially for frequently used function terms. BLEU's method of word matching between translations and references lacks explicitness, potentially resulting in erroneous “matches.” This ambiguity can be particularly problematic for assessing translations in contexts where precise terminology is crucial, such as technical or specialised domains. As a consequence, the lack of specificity in word matching within BLEU may undermine the reliability and accuracy of its assessments, raising concerns about its suitability for certain NLP tasks. These limitations underscore the need for continued refinement and development of evaluation metrics that more accurately reflect the intricacies of translation quality and align with human judgments.

3.5.2 METEOR

METEOR was developed specifically to address the previously mentioned issues with BLEU. Translations are graded according to how closely their translations match a reference translation, word for word. When there are numerous translations of references available, the translation that is provided is assessed separately from each reference, and the highest score is given. The next section goes into more detail about this. Given a pair of translations (a system translation and a reference translation) to compare, METEOR generates an alignment between two strings. A mapping between unigrams is known as alignment, and it occurs when every unigram in one string corresponds to either zero or

one unigram in the other string and to none in the same string. As a result, within a given alignment, a single unigram in one string cannot translate to more than one unigram in the other string (Agarwal & Lavie, 2008; Banerjee & Lavie, 2005). It is utilised in conjunction with BLEU by Sordoni et al. (2015) and K. Xu et al., (2015) for the evaluation of chatbot and image captioning models, respectively.
Meteor assesses translations by computing a score based on explicit word-to-word matches between the translation and a reference translation. When multiple reference translations are available, each translation is scored against them independently, and the best-scoring pair is selected. Meteor establishes a word alignment between the two strings to be compared, ensuring that each word in one string corresponds to at most one word in the other. This alignment is generated incrementally by a sequence of word-mapping modules.

The “exact” module maps two words if they are identical, while the “porter stem” module maps words that become identical after being stemmed using the Porter stemmer. The “WN synonymy” module maps words considered synonyms based on their membership in the same “synset” in WordNet. Initially, all potential word matches between the two strings are identified. The largest subset of these mappings, forming a valid alignment, is then selected. In cases where multiple maximal cardinality alignments exist, Meteor chooses the one that preserves the most similar word order between the strings.
The order in which the modules are executed reflects preferences for word matching. By default, the “exact” module is applied first, followed by “porter stemming,” and then “WN synonymy.” Once the final alignment is determined between the system translation and the reference, the Meteor score is calculated. This score considers the precision (P) and recall (R) of matched unigrams, which are weighted using a parametrized harmonic mean.
To account for the extent to which matched unigrams maintain the same word order, METEOR computes a penalty based on fragmentation. The fragmentation fraction is determined by dividing the number of chunks in the sequence of matched unigrams by the total number of matches. The penalty is then calculated using a function of the fragmentation fraction.
Finally, the Meteor score for the alignment is computed as a combination of the harmonic mean of precision and recall and a penalty term. The parameters of the metric, including α, β, and γ, are tuned to maximise correlation with human judgments of translation adequacy (Lavie & Agarwal, 2007).
The current version of METEOR optimises parameters to maximise Pearson's correlation with human adequacy judgments. However, it's uncertain whether these parameters are optimal for correlating with human rankings. Thus, there is a need to re-tune the parameters to maximise correlation with ranking judgments. This entails computing full rankings based on both the metric and human judgments, and then evaluating suitable correlation measures on those rankings. Each translation hypothesis is assigned a score between 0 and 1, which can be straightforwardly converted into rankings under the assumption that higher scores indicate better hypotheses (Agarwal & Lavie, 2008).

3.5.3 TER

The Translation Error Rate (TER) is a commonly used metric for assessing textual entailment, but its application in assessing chatbot performance has been less than that of other methods. TER is an automatically-generated machine translation evaluation statistic.

The edit distance determines it. By figuring out how many changes are required to go from an output sentence translated by a machine to a reference sentence translated by a human, it determines the mistake rate. Thus, the complement of this error rate is considered when computing the similarity score (Dhyani & Kumar, 2020; Snover et al., 2006).
Among the various automated metrics, the Translation Edit Rate (TER) score stands out as a valuable indicator of post-editing effort required for a project. By quantifying the editing required to align machine translations with reference translations, TER scores offer insights into post-editing workload estimation. Notably, lower TER scores indicate lesser post-editing effort, making them desirable from an efficiency standpoint (Snover et al., 2006).

3.5.4 Perplexity

Perplexity is a widely used automated performance metric for evaluating language models, including those used in dialogue modelling and text generation tasks. It measures how well a probability model predicts a sample, with lower perplexity scores indicating better performance. In the context of language models, perplexity quantifies the model's ability to predict the next word in a sequence, given the preceding words.

Mathematically, perplexity is defined as the exponential of the cross-entropy loss, which is calculated over a test set. It can be interpreted as the weighted average branching factor of the language model – in other words, how many equally likely words can follow any given word. A lower perplexity score suggests that the model is more confident and accurate in its predictions, while a higher score indicates more uncertainty and potential for errors in generated text.

While perplexity is a valuable metric for comparing different language models and tracking improvements during training, it has some limitations when applied to dialogue modelling and open-ended text generation tasks. Perplexity primarily measures the model's ability to predict likely continuations of text, which doesn't always correlate directly with the quality, coherence, or relevance of generated responses in a dialogue context. Additionally, perplexity may not capture important aspects of dialogue such as maintaining context over multiple turns, generating diverse responses, or adhering to specific conversational goals. Therefore, while perplexity remains a useful tool in the evaluation toolkit for language models, it is often used in conjunction with other metrics and human evaluation to provide a more comprehensive assessment of model performance in dialogue systems.

In our study, perplexity was incorporated as part of our evaluation metrics specifically to facilitate a direct comparison with the multimodal model presented in the study by Young et al., (2020) titled “Dialogue Systems with Audio Context”. By using perplexity as a common metric, this study aims to establish a basis for comparison between our multimodal approach and the benchmark set by Poria et al. (2019). This decision allows us to contextualise our results within the broader landscape of multimodal dialogue systems and provides a standardised measure to assess the relative performance of our model. While the study acknowledges the limitations of perplexity in capturing all aspects of dialogue quality, its inclusion enables us to draw meaningful comparisons with existing research and contribute to the ongoing discourse in the field of multimodal dialogue modelling.

3.6 Summary

To conclude, this chapter has outlined a comprehensive methodology for investigating and enhancing dialogue modelling through the application of various Transformer architectures. The research approach encompasses a wide range of components, including diverse

datasets, novel architectural modifications, and rigorous evaluation procedures. By leveraging both textual and audio data from sources such as OpenSubtitles, Cornell Movie-Dialogs, DailyDialog, IEMOCAP, and MELD, the study aims to capture the multifaceted nature of human conversation. The proposed architectures — Baseline Transformer, Encoder-Decoder Transformer, Extractor, and Reencoder — each introduce unique modifications to address the challenges of contextual awareness in dialogue systems. The incorporation of different embedding methods, including custom subword embeddings, GloVe, and BERT, further expands the scope of the investigation. The evaluation framework, utilizing metrics such as BLEU, METEOR, TER, and perplexity, provides a multifaceted approach to assessing model performance. Additionally, the integration of audio embeddings into the dialogue models represents a novel contribution to the field of multimodal conversational AI. By meticulously documenting the training procedures, computational resources, and evaluation metrics, this methodology sets the stage for a thorough exploration of advanced dialogue modelling techniques, potentially leading to significant improvements in the field of conversational AI.

Chapter 4: Implementation

4.1 Introduction

Having established the theoretical underpinnings of the approach in the previous chapter, the study now delves into the practical aspects of bringing it to life. This chapter details the system implementation of our dialogue model leveraging TextSETTR for contextual style transfer, as well as the novel architecture developed named the Reencoder. We'll explore the various deep learning architectures employed, including the specific configuration of the modified and novel architectures.

Furthermore, we'll shed light on the nuts and bolts of our experimental setup. This encompasses the hardware and software infrastructure utilised for training and evaluating the model. The tools and libraries employed for data preprocessing, model development, and performance analysis will also be discussed. By providing a transparent overview of the system implementation, this chapter aims to equip future researchers with the necessary knowledge to replicate and extend upon our work.

4.2 Architectures

4.2.1 Encoder-Decoder Transformer Architecture

Both the baseline architecture provided by TensorFlow and the base Transformer architecture reproduced for this study adhere closely to the original transformer architecture as detailed in the seminal paper “Attention is All You Need” by Vaswani et al., (2017). In this groundbreaking work, the authors introduced a revolutionary architecture for sequence transduction tasks, revolutionising the field of natural language processing.

For the baseline architecture provided by TensorFlow, efforts were made to replicate it through code refactoring, leading to the development of a modularized, object-oriented architecture. These modifications facilitated further experimentation by enabling adjustments to parameters, tokenizers, and embedding layers in subsequent experiments. Despite these alterations, the underlying Transformer architecture remained fundamentally identical to the one proposed by TensorFlow, except for the tokenization step and the embedding layer, which were changed to experiment with and compare different techniques. This approach effectively transformed the Transformer architecture into a control architecture, allowing for rigorous comparative analysis across different experimental conditions.

Consistent with the original paper (Vaswani et al., 2017), this research employs $h = 8$ parallel attention layers, or heads, with a model dimension of $dmodel = 512$, and key and value dimensions ($dk = dv$) of 64 for each head. These parameters are carefully selected to balance computational efficiency with expressive power, ensuring that the model can effectively capture intricate patterns and dependencies in the input data. By adhering to these architectural specifications, the Transformer architecture (Figure 4.1) employed in this study maintains fidelity to the principles outlined in the seminal work by Vaswani et al., thereby providing a robust foundation for subsequent experimentation and analysis in the realm of natural language processing.

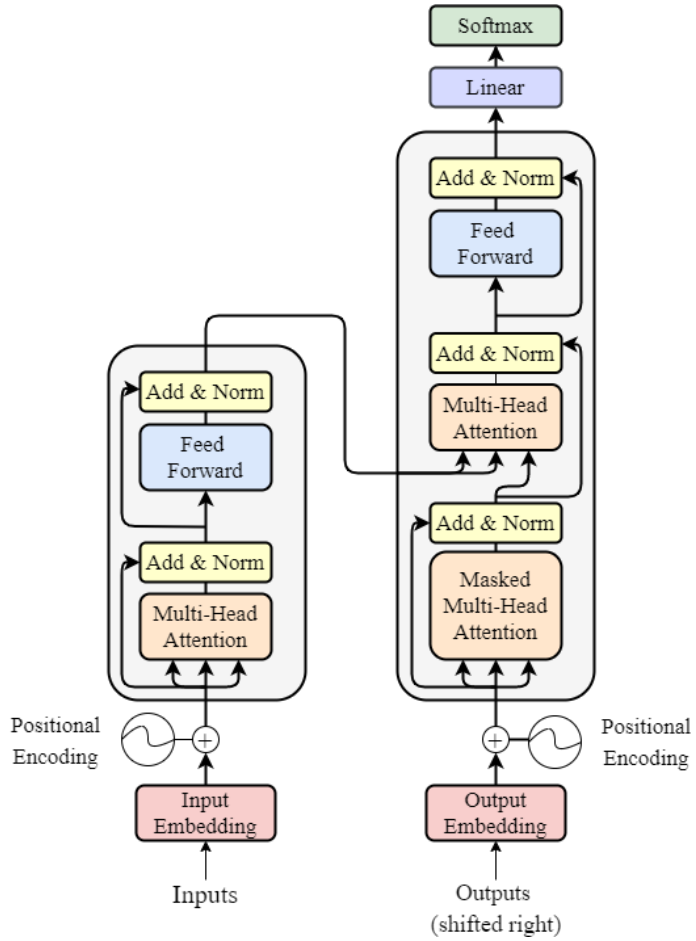

Figure 4.1. Standard Encoder-Decoder Transformer architecture as presented in the paper "Attention is all you need"(Vaswani et al., 2017).

4.2.2 Reencoder

This study extends and enhances the baseline architecture by incorporating contextual information from previous turns in the conversation. However, the same number of attention heads and dimensionality as in the baseline system were used.

The novelty of the Reencoder architecture resides in its innovative approach to the creation and representation of embeddings for textual data. Unlike traditional architectures that solely rely on static embeddings, the Reencoder architecture introduces a dynamic process of embedding representation. This process involves leveraging contextual information from previous turns in a conversation to iteratively refine the embeddings for each utterance. By incorporating historical context into the embedding creation process, the Reencoder architecture enables the model to capture the evolving dynamics and nuances of the conversation, resulting in more nuanced and contextually aware embeddings. This dynamic embedding representation not only enhances the model's ability to understand and generate coherent responses but also enables it to adapt and evolve over the course of a dialogue, thereby significantly improving its performance in conversational tasks. Thus, the Reencoder architecture introduces a novel paradigm in embedding representation, offering a promising avenue for advancing the capabilities of language models in dialogue modelling and other natural language processing tasks.

Initially, at step zero, the model operates in a manner akin to any standard transformer architecture. However, at step one and subsequently at all subsequent steps, a distinctive mechanism is employed. Here, the transformer utilises the embeddings of the current sentence (sentence t) alongside the embeddings of the preceding sentence (sentence t-1) as input. This input configuration is achieved through a matrix multiplication operation, where the embeddings of sentence t are multiplied by those of sentence t-1.

This iterative process allows the model to dynamically incorporate information from previous turns in the conversation, enriching its understanding of the ongoing dialogue and enabling it to generate responses that are informed by contextual cues from preceding interactions. By learning to produce outputs based not only on the immediate input but also on the history of the conversation, the Reencoder model demonstrates enhanced contextual awareness and responsiveness, leading to more coherent and contextually relevant responses in conversational settings.

The novelty of the Reencoder architecture lies therefore in the embedding representation the model creates of textual data. The novel representation is not merely a representation in space of the current sentence (and therefore the current turn of the conversation), but a representation of the current sentence informed by the previous turns in the conversation. This operation actively modifies the vector within the embedding vector space, as shown in Figure 4.2. Highlighted in green in the diagram is the MatMultiply operation that allows to re-encode previous turns of the conversation into the current input.

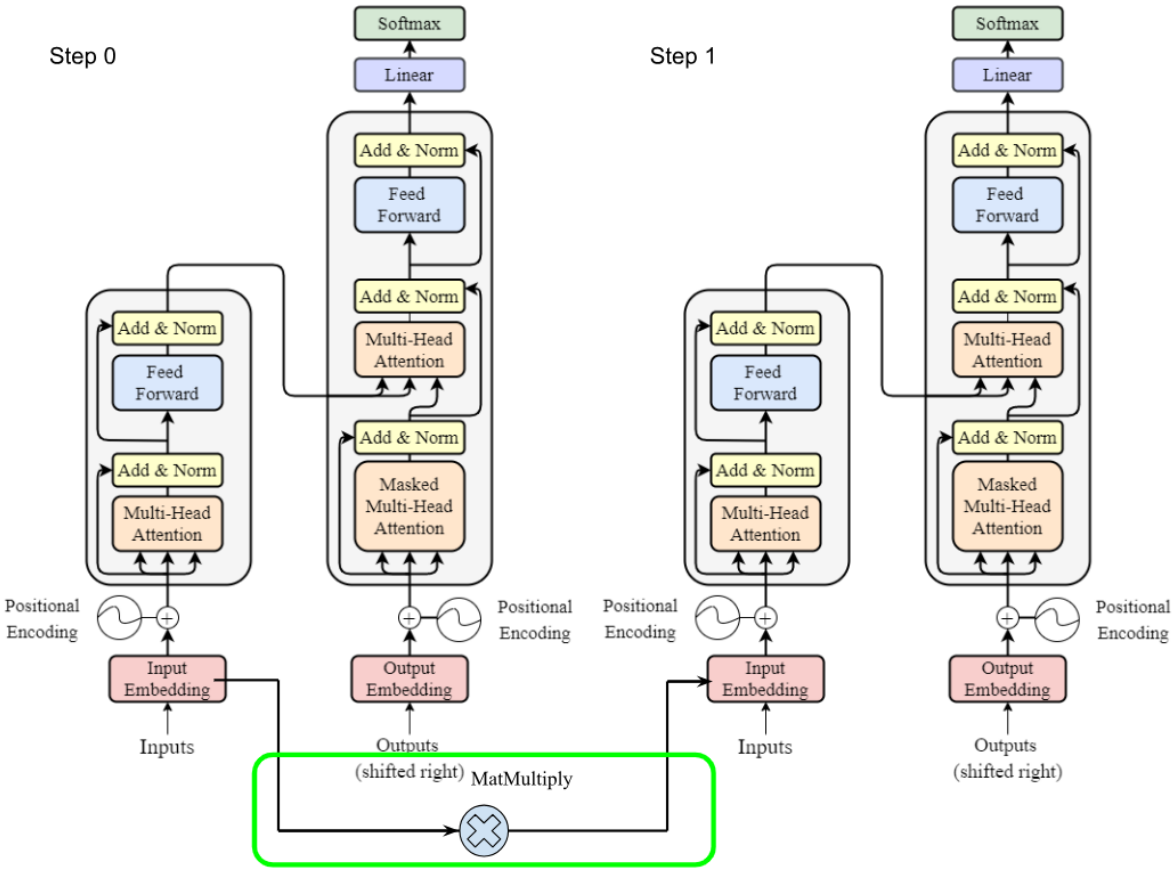

Figure 4.2. Modified Transformer architecture named the Reencoder.

4.2.3 Extractor

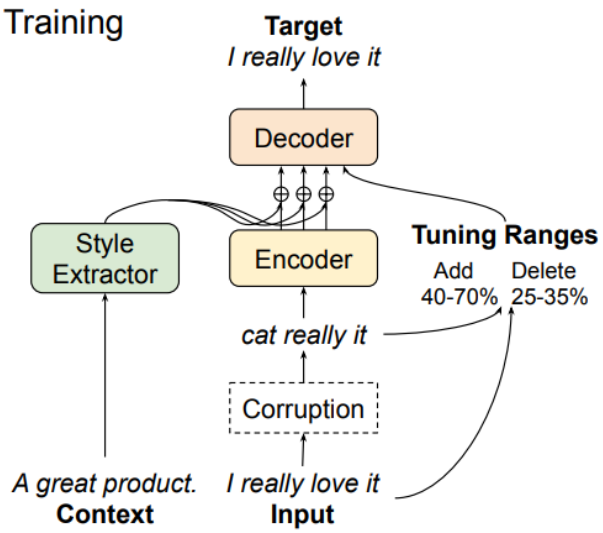

Figure 4.3. Diagram of the modified Transformer architecture presented by Riley et al. (2021)

The extractor model architecture implemented in this research builds upon the innovative work of Riley et al. (2021), who developed a significant enhancement to the T5 (Text-To-Text Transfer Transformer) framework. Their innovation lies in the development of a specialized mechanism for style vector extraction from input text. This extracted style vector serves as a conditioning element for the decoder component, enabling precise control over the stylistic attributes of generated text.
The fundamental architecture leverages T5's proven capabilities as a large-scale pretrained text-to-text model while incorporating novel modifications for style manipulation. As illustrated in Figure 4.3, the system employs a two-stage process: first extracting contextual style information from the input text, then utilizing this information to guide the text generation process through decoder conditioning. This architectural approach represents a significant advancement in controlled text generation, offering enhanced capabilities for style transfer tasks while maintaining the robust performance characteristics of the base T5 model.

Building upon this foundational concept, the extractor model embarks on a parallel trajectory, albeit with a distinct focus on conversational context extraction (Figure 4.4). While retaining the fundamental architectural characteristics such as the number of attention heads and model dimensions, the baseline transformer architecture is ingeniously augmented to incorporate a dedicated component tasked with extracting conversational context from preceding turns in the ongoing dialogue. This is done by adding a MaxPooling step that “condenses” the embeddings of the input sentence at step 0, and concatenates them to the embeddings of the following input sentence at step 1, providing further contextual information (Figure 4.5). This modification enables the model to dynamically capture and integrate contextual information from previous interactions, thereby enriching its understanding of the conversation and facilitating the generation of more contextually relevant responses.

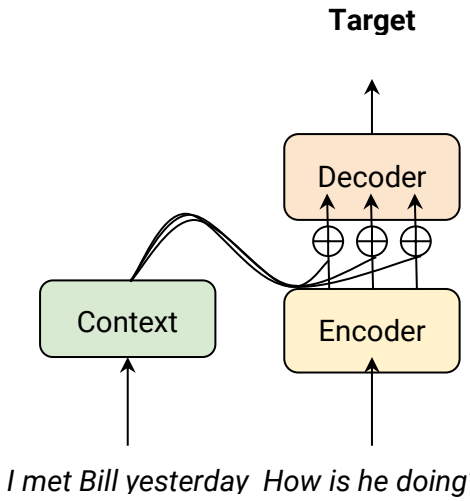

Figure 4.4. Diagram of the modified Transformer architecture used in the current research

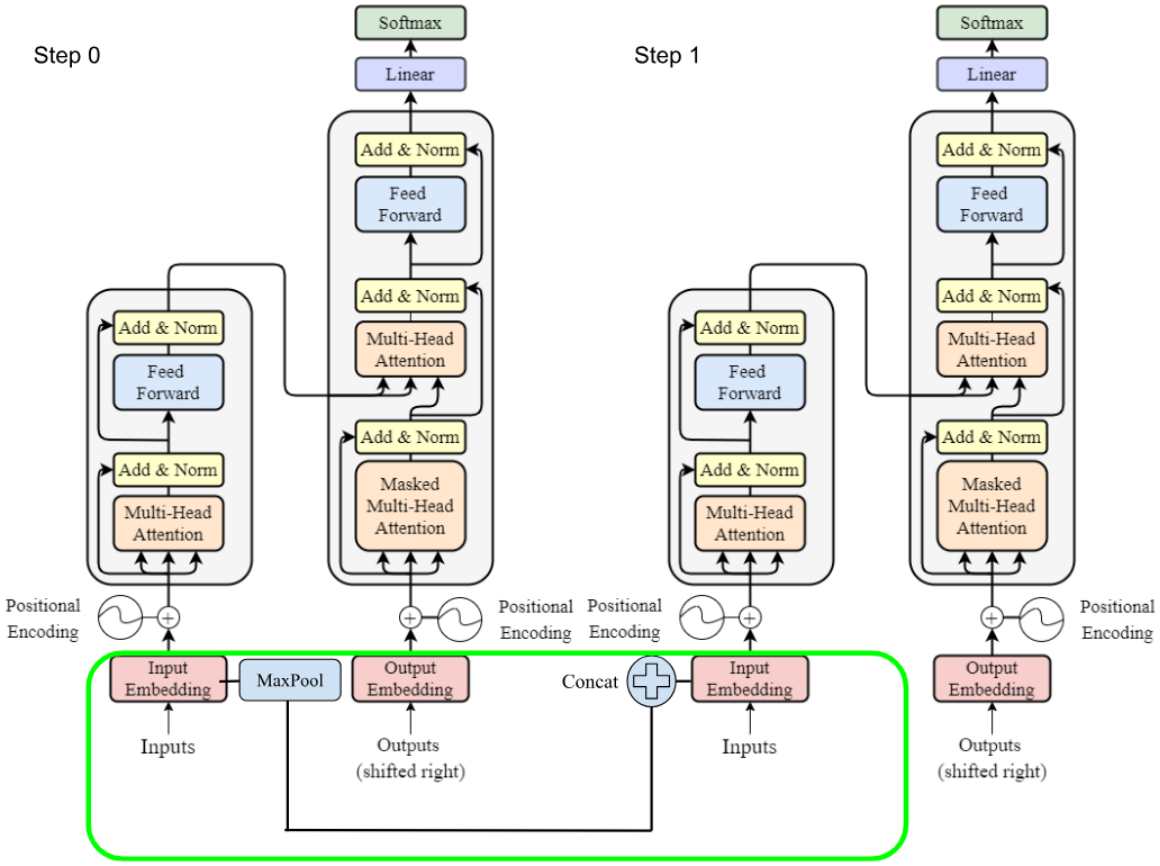

Figure 4.5. Diagram of the modified Transformer architecture called the Extractor.

By adapting and extending the principles outlined in Riley et al.'s architecture to the domain of conversational AI, the extractor model embodies a pioneering approach to enhancing contextual awareness and responsiveness in chatbot systems. This innovative framework not only underscores the versatility and adaptability of transformer-based architectures but

also opens avenues for exploring novel applications and capabilities in the realm of natural language processing. Through continued refinement and experimentation, the extractor model holds promise for advancing the state of the art in conversational AI and shaping the future landscape of human-computer interaction.

4.2.4 Integrating Audio Embeddings

The Audio-Transformer model implementation represents a significant advancement in dialogue modelling by integrating both textual and audio information. This system builds upon the traditional encoder-decoder transformer framework, which is widely used in sequence-to-sequence learning tasks. The key innovation lies in the creation of enriched word representations that combine traditional word embeddings (*wn*) with word-level audio representations (*a˜n*), resulting in a composite representation [*wn; a˜n*]. This integration allows the model to capture a broader range of contextual information from both textual and audio inputs, potentially enhancing its capacity to understand nuances and generate more natural-sounding responses in dialogue scenarios.

A critical component of the system implementation is the extract_embedding function, specifically designed to process video files from the MELD dataset and extract relevant audio embeddings. This function incorporates a sophisticated caching mechanism to optimise efficiency, particularly when dealing with repeated dialogues and phrases. The function's workflow includes several key steps: it first checks for pre-existing cached embeddings, then proceeds to load and process the video file if necessary. The audio processing involves converting the audio to mono, setting a standardised frame rate of 16000 Hz, and adjusting the sample width to 2 bytes. The processed audio is then exported as a WAV file for further analysis.

The core of the audio embedding extraction process utilises the openl3 library's get_audio_embedding function. This step calculates audio embeddings with specific parameters: “env” content type, “linear” input representation, and an embedding size of 512. These embeddings serve as compact representations of the audio signal, capturing relevant features for downstream tasks. To ensure consistency in the data structure, the extracted embeddings are padded to a fixed length using a custom pad function. The resulting embeddings are then converted to PyTorch tensors and saved in a cache file, facilitating quick retrieval in future processing cycles.

The implementation addresses several technical challenges inherent in multimodal data processing. One significant challenge is the alignment and synchronisation of audio features with their corresponding text. This is particularly complex in cases where the audio and text modalities are not perfectly aligned or contain noise or errors. The system likely incorporates sophisticated alignment algorithms, though the specific techniques are not detailed in the provided text. Another challenge is the effective fusion of textual and audio embeddings, as different modalities may have varying levels of importance or relevance depending on the dialogue context. The implementation must balance these modalities to create meaningful and contextually appropriate representations.

From a computational perspective, the system implementation accounts for the increased complexity and resource requirements introduced by processing both textual and audio inputs. This includes managing longer training times and higher computational demands, which can be particularly challenging for large datasets or models with high dimensionality. The implementation likely incorporates optimization techniques to handle these increased resource requirements efficiently. Additionally, the system needs to address the challenges

of audio data preprocessing, including handling various types of noise such as background interference, overlapping speech, or recording artefacts. While specific noise-handling techniques are not detailed, the implementation presumably includes robust audio processing methods to ensure the quality and reliability of the extracted audio embeddings, which are crucial for the overall effectiveness of the Audio-Transformer model.

4.3 Code refactoring

In the pursuit of rigorous quantitative and comparative experimentation within the domain of machine learning, it became evident that the original code, serving as a foundation, lacked the requisite optimization for such systematic analyses. This observation stemmed from the realisation that the TensorFlow research team's initial codebase, structured as a tutorial in Google Colab, inherently favoured pedagogical clarity over the nuanced demands of quantitative research.

To address this limitation, a strategic refactoring initiative was undertaken to imbue the code with a more adaptable and comprehensive structure, aligning with the principles of object-oriented programming. This refactoring process involved preserving the functional integrity of the original TensorFlow code while orchestrating a transformation into modular classes and other object-oriented entities. By encapsulating functionalities within well-defined classes, the codebase attained a heightened level of modularity, thereby fostering increased flexibility for diverse experimentation scenarios.

Notably, the restructured code facilitated a seamless transition between different experimental configurations, spanning diverse architectural models, datasets, and embedding methodologies. The modular design enabled researchers to invoke specific methods corresponding to the desired experimental parameters, thereby streamlining the execution of experiments within a unified framework. This modular approach not only enhanced the code's readability and maintainability but also empowered researchers to systematically explore and compare outcomes across various configurations.

In conjunction with this refactoring effort, integration with Weights and Biases (WandB) was incorporated, further enhancing the research workflow. Leveraging WandB's experiment tracking capabilities, researchers could efficiently log and monitor the outcomes of diverse experiments. This integration not only expedited the iterative refinement process but also provided a centralised platform for collaborative analysis and comprehensive documentation of experiment results. In essence, the combined adoption of object-oriented principles and experiment tracking through WandB contributed to the creation of a robust and versatile research framework, laying the groundwork for meticulous experimentation and systematic comparison within the realm of machine learning.

4.3.1 Data caching and retrieval for optimization

To optimize computational efficiency throughout the experimental phase, this research implemented a systematic data pre-caching strategy. The methodology involved generating structured JSON files that contained preprocessed input-output sentence pairs in raw text format. This preprocessing approach was designed to minimize computational

overhead during experimentation by eliminating redundant data cleaning and loading operations.

The implementation leveraged the Weights and Biases (WandB) platform as a centralized repository for artifact storage and management. By storing the preprocessed JSON files as WandB artifacts, the research established a robust framework for data versioning and experimental tracking. This infrastructure facilitated systematic access to experimental data while maintaining comprehensive documentation of data utilization patterns across different experimental configurations.

The generation of JSON files was methodically tailored to accommodate the specific requirements of individual experiments. Each file was structured to contain the precise number of samples necessary for its corresponding experimental iteration, thereby optimizing storage utilization while ensuring data accessibility. This granular approach to data preprocessing served two critical purposes: it conserved computational resources by eliminating unnecessary data loading, and it enhanced experimental reproducibility by maintaining consistent, well-defined datasets for each iteration.

The integration of data pre-caching mechanisms with the WandB platform's artifact management system represents a methodological advancement in experimental workflow optimization. This approach not only streamlined the experimental process but also established a foundation for systematic data management and experimental reproducibility, key considerations in computational research methodology.

4.4 Summary

This chapter detailed the implementation of several innovative architectural models for dialogue systems, including the baseline Encoder-Decoder Transformer, the novel Reencoder, and the Extractor model. Each of these architectures built upon the foundational work of Vaswani et al. 's “Attention is All You Need,” with specific enhancements to improve contextual understanding in conversational AI. The Reencoder architecture introduced a dynamic process of embedding representation, leveraging historical context to refine embeddings iteratively. The Extractor model, inspired by Riley et al.'s work, focused on extracting conversational context from preceding dialogue turns. Additionally, the chapter explored the integration of audio embeddings, presenting the Audio-Transformer model as a multimodal approach to enhance word representations in dialogue modelling.

The experimental setup leveraged cutting-edge tools and platforms to facilitate efficient and collaborative research. Google Colab was utilised as the primary development environment, offering accessibility, high-performance GPUs, and seamless integration with Google Cloud services. To enhance experiment tracking and collaboration, Weights & Biases (WandB) was incorporated, providing comprehensive logging of model performance metrics, advanced visualisation tools, and robust support for model versioning and experiment comparison. These tools collectively contributed to a more streamlined and transparent research process.

Significant effort was invested in code refactoring to optimise the original TensorFlow tutorial code for quantitative research. This refactoring process transformed the codebase into a modular, object-oriented structure, enhancing flexibility and facilitating systematic exploration of various experimental configurations. Furthermore, data caching and retrieval optimizations were implemented, including the creation of pre-cached JSON files stored as artefacts in WandB. These optimizations aimed to enhance computational efficiency and ensure reproducibility across experiments. Overall, this chapter laid out a comprehensive framework for conducting rigorous, reproducible, and efficient research in the field of conversational AI, setting the stage for the detailed experiments and analyses to follow.

Chapter 5: Experiments and Results

5.1 Introduction

This chapter presents the experimental methodologies and results obtained from evaluating different transformer architectures for chatbot modelling across various datasets.

The training procedure for the chatbot model involved conducting a series of experiments utilising different transformer architectures and datasets, each paired with three distinct embedding methods. The first architecture, the baseline Transformer Model Architecture provided by TensorFlow (*Google Colab*, n.d.), served as the foundation for comparison against the other architectures. For each dataset, this architecture was trained using three different embedding methods: an automated embedding matrix generated from the data using an embedding class provided by the TensorFlow library, GloVe embeddings, and BERT embeddings. The training process involved standard preprocessing steps, including data cleaning, tokenization, and division into training, validation, and test sets. Our inquiry delves into a comprehensive examination and comparative analysis of diverse tokenization methodologies, each wielding distinct characteristics and exerting varying impacts on the embedding layer of large language models (LLMs). Our investigation encompasses an exploration of notable tokenization techniques such as tfds.deprecated.text.SubwordTextEncoder, a specific tokenizer class created to leverage GloVe embeddings, and the BERT Tokenizer, aiming to elucidate their individual attributes and discern their respective effects on the embedding layer's functionality within expansive LMs. The transformed data was then fed into the baseline transformer architecture, and the model was trained using backpropagation and gradient descent optimization to minimise a predefined loss function.

The goal of these experiments is to assess the performance and effectiveness of different transformer models in generating contextually appropriate responses in conversational settings.

5.2 Experimental Methodologies

Four architectures were compared in the experiment:

- The Baseline model serves as the foundational framework provided by the TensorFlow organisation, representing the standard architecture utilised for chatbot modelling in the field of natural language processing. This architecture lays the groundwork for subsequent variations and serves as a baseline for comparison against other models.
- The Encoder-Decoder Transformer Architecture is enhanced through the integration of different embedding methods. In this variant, three distinct embedding techniques were employed and systematically compared to evaluate their impact on model performance and response generation capabilities.
- The Extractor Architecture, inspired by the work of Riley et al., (2021), introduces modifications to the traditional transformer model to enhance its ability to extract contextual information from preceding turns in the conversation. By incorporating

contextual cues from previous interactions, this architecture aims to improve the model's understanding of the ongoing dialogue and facilitate more contextually relevant responses.

- The Reencoder Architecture adopts a different approach by integrating embeddings from prior utterances into the current turn's embeddings. By leveraging historical context embedded within the conversation, this architecture seeks to enrich the current turn's representation with valuable contextual information, thereby enhancing the model's ability to generate coherent and contextually appropriate responses.
- Each of these architectures underwent the same training procedure as the baseline architecture, with variations in model structure and input representations.
- Building upon the foundation of these four architectures — the Baseline model, the Encoder-Decoder Transformer with varied embedding methods, the Extractor Architecture, and the Reencoder Architecture — each model was further modified to incorporate audio embeddings alongside text embeddings. This modification involved integrating the extract_embedding function to process audio data from the MELD and IEMOCAP datasets, generating word-level audio representations using the openl3 library. These audio embeddings were then combined with the existing text embeddings to create enriched multimodal representations for each word. The integration process required careful alignment of audio and text modalities and implementation of appropriate fusion techniques. This augmentation aimed to enhance each architecture's ability to capture and utilise both linguistic and acoustic features in dialogue modelling, potentially improving their performance in tasks such as response generation, emotion recognition, and context understanding.

The Encoder-Decoder architecture, similar to the TensorFlow baseline but with different embedding methods, aimed to explore the impact of embedding variations on model performance. The Extractor architecture, inspired by Riley et al., (2021), focused on extracting contextual information from previous turns in the conversation to enhance the model's understanding of dialogue context. The Reencoder architecture, on the other hand, incorporated embeddings from previous utterances to inform the current turn's embeddings, enabling the model to leverage historical context for response generation.
Each of these architectures offers unique insights and approaches to addressing the challenges of conversational AI, ranging from optimising embedding methods to leveraging historical context within the dialogue. Through systematic evaluation and comparison, researchers aim to gain a deeper understanding of the strengths and limitations of each architecture, ultimately advancing the field of conversational AI and contributing to the development of more sophisticated and contextually aware chatbot models.

5.3 Training Process

Each dataset underwent rigorous standard preprocessing procedures, which encompassed partitioning into distinct training, validation, and test sets. These sets were allocated 89%, 10%, and 1% of the total data, respectively, ensuring a well-balanced distribution for robust model evaluation. Subsequently, the data underwent embedding and vectorization processes, preparing them for ingestion into the designated transformer architectures for training. To expedite training experiments and to ensure computational

efficiency, a combination of resources including Google Colab runtimes and virtual machines on the Google Cloud Platform was leveraged.

The experiments were conducted meticulously across all datasets and embedding methods, encompassing a wide range of variations to ensure a comprehensive evaluation of the effectiveness of different transformer architectures in generating contextually appropriate responses within conversational settings. This exhaustive approach aimed to provide insights into the comparative performance of various models under diverse conditions, enabling a nuanced understanding of their strengths and limitations. By systematically analysing the outcomes across multiple datasets and embedding techniques, researchers aimed to glean valuable insights into the efficacy of transformer architectures in capturing and synthesising meaningful dialogue interactions, thereby advancing the state of the art in conversational AI research.

In each training experiment, a predetermined duration of 1000 epochs was established to allow the model to undergo iterative learning and convergence towards optimal performance. However, to optimise training efficiency and prevent overfitting, a callback for early stopping was implemented . This callback mechanism was meticulously defined to monitor the training process continuously. If the training loss failed to decrease or remained stagnant for three consecutive epochs, the callback triggered the early stopping protocol. By halting the training process at this point, the risk of the model becoming overly specialised to the training data was mitigated, thereby enhancing its generalisation capability and preventing potential performance degradation on unseen data.

The early stopping callback served as a vital safeguard against the phenomenon of overfitting, which occurs when the model excessively memorises the training data, leading to suboptimal performance on new, unseen data. By dynamically monitoring the training loss throughout the epochs, signs of overfitting could be detected in real-time and take proactive measures to halt the training process before it detrimentally impacted the model's generalisation ability. This proactive approach not only safeguarded against overfitting but also optimised computational resources by terminating training once further improvements in performance were unlikely, thereby expediting the experimentation process and accelerating the model development cycle.

Moreover, the implementation of the early stopping callback ensured that each training experiment remained aligned with our overarching goal of achieving optimal performance within a reasonable timeframe. By setting clear criteria for terminating the training process based on the behaviour of the training loss, consistency, and reproducibility was maintained across experiments while maximising the efficiency of resource utilisation. This strategic approach to training regimen design underscored our commitment to rigour and methodological integrity, laying the groundwork for robust and reliable research outcomes in our exploration of machine learning models for dialogue modelling tasks.

5.4 Unimodal Results

The experimental results demonstrate the performance of each transformer architecture across the different datasets. Results are presented in terms of quantitative evaluation metrics (e.g., BLEU, METEOR, TER, and Accuracy).

A result analysis will be conducted, first comparing how the different architectures and embedding layers performed for each dataset individually, and then comparing how a model performed across datasets. For clarity purposes, the study only presents three out of the four evaluation metrics used, BLEU, METEOR and TER, since these better represent output quality, coherence and textual entailment. Similarly, experiment results for GloVe embedding are not presented in this section, since Bert appears to outperform GloVe in most experiments. Tables and figures presenting all results can be found in Appendix A.

5.4.1 Results on the DailyDialog Dataset (Text Embeddings)

***Table 3**. System evaluation on DailyDialog dataset using different embedding algorithms and performance measures.*

Architectures/ Embeddings	**Extractor**		**Reencoder**		**Encoder-decoder**		**baseline**	
	BERT	Matrix3	BERT	Matrix3	BERT	Matrix3	BERT	Matrix3
BLEU	0.078	0.004	0.085	**0.102**	0.090	0.000	0.000	0.081
METEOR	0.157	0.055	0.159	**0.169**	0.163	0.008	0.000	0.156
TER	112.981	215.704	110.165	110.717	110.400	**98.823**	0.000	112.726

Upon comparing the results obtained from various architectures and embedding layers on the DailyDialog dataset, our analysis reveals the superior performance of the Reencoder model across multiple evaluation metrics, including BLEU and METEOR. It is worth noting that the SubwordTokenizer (Matrix3) exhibits superior performance in terms of BLEU and METEOR scores (Table 3). Since the Reencoder model operates on the contextual representation of language by leveraging previous turns in the conversation in order to create more contextually aware vector representations of sentences in the embedding space, its performance when modelling human dialogue appears enhanced.

Interestingly, the TER score suggests that the base Encoder-Decoder Transformer architecture, particularly when coupled with the SubwordTokenizer, yields optimal performance. However, it is pertinent to highlight that the Reencoder model consistently demonstrates the best TER scores among architectures utilising Bert (Table 3). These findings underscore the nuanced interplay between different architectures and embedding methods, with the Reencoder model emerging as a promising candidate for enhancing performance across various metrics on the DailyDialog dataset.

5.4.2 Results on the Cornell Dataset (Text Embeddings)

***Table 4**. System evaluation on Cornell dataset using different embedding algorithms and performance measures.*

Architecture/ Embeddings	**Extractor**		**Reencoder**		**Encoder-decoder**		**baseline**	
	BERT	Matrix3	BERT	Matrix3	BERT	Matrix3	BERT	Matrix3
BLEU	0.006	0.006	**0.007**	0.003	0.005	0.002	0.000	**0.007**
METEOR	0.079	0.081	0.070	0.055	0.079	0.037	0.000	**0.087**
TER	117.654	127.885	**117.483**	249.194	124.280	246.923	0.000	119.127

In the comparative analysis of different architectures and embedding layers on the Cornell dataset, distinctive trends emerge, elucidating the nuanced performance variations across evaluation metrics. Reencoder model demonstrates remarkable performance, boasting the best TER scores overall and exhibiting superior scores across two out of three embedding methods (Bert and GloVe), albeit registering the lowest TER score among all models employing the SubwordTokenizer (Table 4). Furthermore, the Reencoder model showcases the highest BLEU scores, particularly when leveraging Bert as the embedding layer. Remarkably, Bert exhibits exceptional performance across various metrics, underscoring its efficacy on this dataset. This observation aligns with the consistent trends observed in BLEU scores across models (Table 4). However, METEOR scores suggest that the SubwordTokenizer yields optimal performance overall, yet Bert consistently provides the most reliable and consistent results, yielding the best scores on average across metrics. These findings underscore the multifaceted nature of model performance across different architectures and embedding layers, highlighting Bert's robustness and effectiveness in enhancing performance on the Cornell dataset.

5.4.3 Results on the OpenSubtitles Dataset (Text Embeddings)

***Table 5**. System evaluation on OpenSubtitles dataset using different embedding algorithms and performance measures.*

Architecture/ Embeddings	**Extractor**		**Reencoder**		**Encoder-decoder**		**baseline**	
	BERT	Matrix3	BERT	Matrix3	BERT	Matrix3	BERT	Matrix3
BLEU	0.000	0.000	0.003	0.000	0.002	0.000	0.000	**0.012**
METEOR	0.055	0.064	0.061	0.041	0.056	0.062	0.000	**0.119**
TER	130.076	133.399	132.709	126.866	128.429	259.307	0.000	**118.553**

In the comparative evaluation of various architectures and embedding layers on the OpenSubtitles dataset, discernible patterns emerge, shedding light on the nuanced performance variations across multiple evaluation metrics. Notably, the performance of the TensorFlow base model surpasses all proposed architectures across all metrics except accuracy, highlighting its unexpected efficacy on this dataset (Table 5). These findings suggest that the Encoder-decoder model, having less parameters, is better suited to tackle noisy datasets such as Opensubtitles.

5.4.4 Results on the Meld Dataset (Text Embeddings)

***Table 6**. System evaluation on Meld dataset using different embedding algorithms and performance measures on text embeddings only.*

Architecture/ Embeddings	**Extractor**		**Reencoder**		**Encoder-decoder**		**baseline**	
	BERT	Matrix3	BERT	Matrix3	BERT	Matrix3	BERT	Matrix3
BLEU	0.000	0.000	0.000	0.000	**0.009**	0.000	0.000	0.000
METEOR	0.080	0.086	0.071	0.087	**0.092**	0.079	0.000	0.000
TER	118.499	114.286	**110.380**	122.097	117.163	112.025	0.000	0.000

In our comprehensive evaluation of various transformer architectures over the Meld dataset, intriguing patterns emerged regarding their performance across diverse evaluation metrics. Notably, the Encoder-Decoder Transformer model exhibited remarkable superiority over all other architectures across the majority of metrics assessed, showcasing its efficacy in capturing the intricacies of dialogue modelling. However, it is crucial to note that the Reencoder architecture outperformed other models in terms of the TER metric, underscoring its distinct strengths in certain aspects of dialogue processing (Table 6). Interestingly, across all cases, architectures enhanced with a Bert embedding layer consistently outperformed those utilising alternative embedding layers, suggesting the robustness and versatility of Bert embeddings in capturing semantic nuances and linguistic complexities inherent in dialogue datasets. These findings underscore the nuanced interplay between architecture design and embedding strategies in shaping model performance and highlight the significance of comprehensive evaluations to elucidate the relative strengths and weaknesses of different transformer configurations in dialogue modelling tasks. Further exploration is warranted to delve deeper into the underlying mechanisms driving these observed performance disparities and to identify strategies for optimising transformer architectures for enhanced dialogue processing capabilities.

The consistently low BLEU scores seen in different dialogue modelling systems can be traced back to the type of data used for training — typically conversations from TV shows. This source material is characterised by its open-ended nature, where multiple responses could be equally valid for a given prompt. TV dialogues often include contextual nuances, informal language, and abrupt topic changes, making them quite different from more structured language tasks. While the AI models might generate responses that are

conversationally appropriate and natural, these outputs typically differ substantially from the specific reference answers used in BLEU score evaluations. This discrepancy results in lower BLEU scores, even when the AI-generated responses may be high-quality in terms of relevance and fluency within the conversation.

It is important to note that the Meld dataset could not be tested against the baseline architecture provided by Tensorflow, because that would have required a significant change in the architecture.

5.4.5 Results on the OpenSubtitles Datasets with Training data corresponding to 1% of the entire dataset (Text Embeddings)

***Table 7**. System evaluation on OpenSubtitles Dataset with Training data corresponding to 1% of the entire dataset using different embedding algorithms and performance measures.*

Architecture/ Embeddings	**Extractor**		**Reencoder**		**Encoder-decoder**		**baseline**	
	BERT	Matrix3	BERT	Matrix3	BERT	Matrix3	BERT	Matrix3
BLEU	0.005	**0.008**	0.004	0.000	0.004	0.005	0.000	0.004
METEOR	0.088	0.105	0.098	**0.108**	0.092	0.107	0.000	0.052
TER	123.537	122.231	121.130	**116.494**	119.567	121.373	0.000	132.7584

As previously mentioned, a crucial aspect of conducting experiments across datasets of varying sizes is ensuring comparability. To achieve this, a parameter known as max_samples was introduced, dictating the maximum number of sentences utilised during the training phase. By imposing a consistent limit on the amount of training data across different datasets, researchers can effectively control for size discrepancies and enable fair comparisons. In this study, the max_samples parameter was set to 440,000, a value carefully chosen to maintain a comparable order of magnitude across multiple datasets. This selected value of 440,000 holds significance as it corresponds to the entirety of several key datasets utilised in the experiments. Specifically, it aligns with the entire Meld dataset, the complete Endure DailyDialog dataset, the entirety of the Cornell dataset, and approximately 0.1% of the OpenSubtitles dataset. By standardising the training data size in this manner, researchers ensure that each model receives a representative subset of the available data, facilitating meaningful comparisons of their performance.

However, to assess the robustness of the transformer architectures across larger datasets, additional experiments were conducted with an increased number of samples. In these experiments, the max_samples parameter was set to 4,400,000, representing a tenfold increase in data compared to the initial setting. This adjustment allowed researchers to evaluate the performance of the different architectures over a more extensive dataset, equivalent to 1% of the Meld dataset. By systematically varying the amount of training data in this manner, researchers gain insights into how the models scale with data volume, thereby contributing to a deeper understanding of their capabilities and limitations in dialogue modelling tasks.

Our results revealed that the choice of transformer architecture and embedding method significantly impacted the model's performance. Among the transformer architectures, the Reencoder model achieved the lowest TER score and the highest METEOR scores when combined with the custom embeddings generated from the data, outperforming the other architectures in most cases (Table 7).
Interestingly, the Extractor architecture exhibited competitive performance, with a BLEU score of 0.008145 and a TER score of 0.105414 (Table 7).

The standard Encoder-Decoder Transformer architecture performed reasonably well across different embedding methods, but its performance was generally lower compared to the Reencoder and Extractor architectures in our experiments.
Performance across models degraded when using other embedding methods, such as GloVe. It remained however competitive when using BERT.
Our findings suggest that the combination of transformer architectures specifically designed for sequence-to-sequence tasks, such as the Reencoder and Extractor models, along with high-quality pre-trained embeddings like BERT, or embeddings that have been fitted to the data such as the custom embedding layer, can significantly improve the performance of machine translation models on the OpenSubtitles dataset.
One of the contributing factors can be the presence of Subword Information. Both the custom embedding layer and BERT embeddings are based on subword representations, which means they can effectively handle out-of-vocabulary words and rare words by breaking them down into smaller meaningful units (subwords). This ability to represent rare or unseen words is particularly beneficial for tasks like machine translation, where the vocabulary can be vast and diverse.
However, further investigation is needed to understand the impact of different hyperparameters, such as batch size, dropout rate, and model dimensions, on the overall performance.

The findings from the experiments conducted over a larger portion of the OpenSubtitles dataset revealed a lack of notable enhancement across any of the evaluated metrics for the various architectures under study. This outcome prompts consideration of two potential contributing factors. Firstly, the OpenSubtitles dataset's inferior quality may have played a pivotal role in the observed results. Issues such as noise, inconsistency, or insufficient relevance within the dataset can hinder the models' ability to effectively learn and generalise patterns from the data. This dataset quality aspect warrants further investigation to ascertain its impact on model performance comprehensively.

Secondly, the inefficiency of larger datasets for smaller language models may have contributed to the observed lack of improvement. Contrary to expectations, smaller language models seem to derive more substantial benefits from high-quality, compact datasets rather than larger, more extensive ones. It's conceivable that smaller language models, with their inherently limited capacity for processing complex data, struggle to extract meaningful insights or patterns from larger datasets. This limitation might stem from the models' reduced capacity to capture and encode the intricate nuances present in vast amounts of data, thus hindering their learning potential.

Moreover, these two factors may not operate independently but could instead be intertwined, exacerbating the challenges faced by smaller language models when trained on larger datasets of inferior quality. The interplay between dataset quality and model size underscores the intricate dynamics involved in dialogue modelling tasks. Further exploration into these intertwined factors is essential to unravel their complexities and devise strategies to mitigate their adverse effects on model performance. Such insights are crucial for refining model training methodologies and optimising the selection of datasets, ultimately advancing the effectiveness of language models in dialogue modelling applications.

5.4.6 Results on the Meld Dataset (Audio Embeddings)

***Table 8**. System evaluation on MELD Dataset using audio embeddings and different performance measures.*

Architecture/ Embeddings	**Extractor**	**Reencoder**	**Encoder-decoder**	**baseline**
	Audio Embeddings	Audio Embeddings	Audio Embeddings	Audio Embeddings
BLEU	0.000	0.000	0.000	0.000
METEOR	**0.031**	0.030	0.030	0.000
TER	149.445	**143.323**	159.67	0.000

The models trained on audio embeddings only present an overall similar performance. The Reencoder model appears best in terms of TER score (Table 8).

Low BLEU scores observed across various dialogue modelling architectures can likely be attributed to the open-ended nature of the training data, which often consists of conversations extracted from TV shows. These dialogues are inherently diverse and unpredictable, with multiple valid responses possible for any given input. Unlike more constrained language tasks, TV show conversations frequently feature context-dependent replies, colloquialisms, and non-linear topic shifts. As a result, the models trained on this data produce responses that may be contextually appropriate and natural-sounding, but differ significantly from the specific reference responses used in BLEU score calculations. This divergence leads to lower BLEU scores, despite the generated responses potentially being high-quality in terms of conversational relevance and fluency.

5.4.7 Results on the IEMOCAP dataset (Text Embeddings)

***Table 9**. System evaluation on IEMOCAP Dataset using text embeddings and different performance measures.*

Architecture/ Embeddings	**Extractor**		**Reencoder**		**Encoder-decoder**		**baseline**	
	BERT	Matrix3	BERT	Matrix3	BERT	Matrix3	BERT	Matrix3
BLEU	**0.207**	0.037	0.200	0.060	0.178	0.070	0.000	0.000
METEOR	**0.395**	0.187	0.387	0.189	0.366	0.245	0.000	0.000
TER	**94.619**	157.459	95.811	124.275	99.2	130.84	0.000	0.000

The Extractor architecture, when trained on the IEMOCAP transcripts, demonstrates superior performance across all evaluation metrics, including BLEU, METEOR, and TER. This consistent outperformance suggests that the Extractor's approach to processing and generating text is particularly well-suited for the task at hand, potentially due to its ability to effectively capture and utilise relevant information from the input data (Table 9).

Following closely behind the Extractor, the Reencoder architecture shows comparable performance, establishing itself as a strong second-place contender. The narrow gap between these two architectures indicates that both approaches have significant merit in handling the given task. This close competition between the Extractor and Reencoder architectures may provide valuable insights into the most effective strategies for processing and generating text in this specific context, and could inform future research and development in the field.

It is important to note that the IEMOCAP dataset could not be tested against the baseline architecture provided by Tensorflow, because that would have required a significant change in the architecture.

5.4.8 Results on the IEMOCAP dataset (Audio Embeddings)

***Table 10**. System evaluation on IEMOCAP Dataset using audio embeddings and different performance measures.*

Architecture/ Embeddings	**Extractor**	**Reencoder**	**Encoder-decoder**	**baseline**
	Audio Embeddings	Audio Embeddings	Audio Embeddings	Audio Embeddings
BLEU	0.000	0.000	0.000	0.000
METEOR	0.028	**0.035**	0.032	0.000
TER	144.272	**120.764**	127.656	0.000

In experiments focusing solely on audio embeddings from the IEMOCAP dataset, the Reencoder architecture demonstrates superior performance compared to other tested architectures, particularly in terms of METEOR and TER scores. This suggests that the Reencoder's approach is particularly effective in processing and utilising audio information. However, a notable observation across all architectures is the occurrence of null BLEU scores (Table 10). This uniform result in BLEU metrics indicates a potential limitation in how these architectures handle the translation of audio embeddings into text that aligns with reference translations. The null BLEU scores might point to a fundamental challenge in preserving certain aspects of the original input when working exclusively with audio embeddings, or it could suggest that BLEU may not be the most suitable metric for evaluating performance in this specific audio-to-text task.

It is important to note that the IEMOCAP dataset could not be tested against the baseline architecture provided by Tensorflow, because that would have required a significant change in the architecture.

5.5 Multimodal Results

It is important to note that the decision to evaluate our multimodal models on the MELD (Multimodal EmotionLines Dataset) dataset and IEMOCAP dataset, while the unimodal models were assessed on multiple datasets, stems from the unique characteristics and requirements of our multimodal approach. The MELD and IEMOCAP datasets stand out as an ideal choice for our multimodal experiments due to its rich, multi-faceted nature, providing both textual dialogues and corresponding audiovisual data from TV show scenes and scripted or improvised dialogue. This dataset's structure aligns perfectly with our research objectives, allowing us to explore the integration of audio features alongside textual information in dialogue modelling.

Unlike the unimodal experiments, which primarily relied on text-based datasets, our multimodal models require synchronised audio and text data to function effectively. MELD and IEMOCAP datasets offer this crucial alignment, providing time-stamped utterances paired with their corresponding audio segments. This synchronisation is essential for our audio embedding extraction process and the subsequent integration of audio features with text embeddings. Other commonly used dialogue datasets, while valuable for text-based models, lack the necessary audio components, making them unsuitable for our multimodal experiments.

Furthermore, the MELD and IEMOCAP datasets diversity in emotional content and conversational contexts provides a robust testing ground for our multimodal models. It allows us to evaluate how the integration of audio features enhances the model's ability to capture nuances in emotion, tone, and context that may not be apparent from text alone. While this focus on two datasets for multimodal evaluation might seem limiting compared to the broader range used for unimodal models, it actually allows for a more controlled and in-depth analysis of the impact of audio integration across different architectural variations. This approach enables us to draw more precise conclusions about the effectiveness of our multimodal techniques within a consistent experimental framework.

5.5.1 Results on the MELD dataset (text and audio embeddings)

The MELD dataset (Multimodal EmotionLines Dataset) consists of short videos in mp4 format. Each video corresponds to an utterance, and several videos together form a dialogue. For this experiment, transcriptions of dialogues were used as previously, and in addition to the text transcriptions, audio embeddings were extracted from the videos, which capture the emotional and acoustic characteristics of the dialogues.
Furthermore, the incorporation of audio embeddings can enhance tasks such as emotion recognition and intent classification (Pandeya et al., 2021). Audio cues provide valuable information about the speaker's emotional state and intent, allowing multimodal models to better interpret the nuances and underlying meanings conveyed in dialogues. Additionally, in real-world scenarios where text data can be noisy or ambiguous, audio embeddings can help resolve ambiguities and provide additional context, leading to more robust and accurate dialogue systems (Young et al., 2020).

The following table presents our results; it is important to note that testing the Meld dataset against the TensorFlow baseline architecture was infeasible, as it would have necessitated substantial architectural modifications, deviating significantly from the original design.

***Table 11**. System evaluation on Meld Dataset with Audio and Text extracted using different embedding algorithms and performance measures.*

Architecture/ Embeddings	**Extractor**		**Reencoder**		**Encoder-decoder**		**baseline**	
	BERT	Matrix3	BERT	Matrix3	BERT	Matrix3	BERT	Matrix3
BLUE	0.000	0.002	**0.004**	0.002	0.000	0.000	0.000	0.000
METEOR	0.064	0.069	**0.077**	0.06	0.053	0.065	0.000	0.000
TER	125.578	236.913	**106.212**	311.49	106.688	314.73	0.000	0.000
Perplexity	3.263	3.364	3.292	2.899	**2.867**	3.421	0.000	0.000

Based on the provided results, the performance of the different model-embedding combinations can be analysed and their strengths and weaknesses for the dialogue modelling task on the MELD dataset discussed.

Looking at the results, the Reencoder architecture emerges as the top performer across most metrics when paired with BERT embeddings. It achieves the highest BLEU score (0.004) and METEOR score (0.077), indicating better quality and fluency in generated responses. The Reencoder with BERT embeddings also shows the lowest TER (106.212), suggesting fewer errors in the generated text compared to other combinations. However, it's worth noting that the Encoder-decoder architecture with BERT embeddings achieves the lowest perplexity (2.867), indicating potentially better predictive performance in some aspects (Table 11).

The Matrix3 embeddings, while generally performing well, don't consistently outperform BERT embeddings across all architectures and metrics. This suggests that while Matrix3 is a viable alternative, BERT embeddings might have a slight edge in overall performance, particularly when paired with the Reencoder architecture. These results underscore the importance of carefully selecting both the architecture and embedding method in dialogue modelling tasks.

The experimental results reveal intriguing patterns and trade-offs among the Extractor, Reencoder, and Encoder-Decoder Transformer models across different embedding methods. The Reencoder model, particularly when paired with BERT embeddings, demonstrated superior performance. This suggests that the Reencoder architecture, with its ability to leverage contextual information from previous conversation turns, is particularly effective at capturing the nuanced semantics of dialogues when combined with the rich, contextualised representations provided by BERT.

Across all three architectures, BERT embeddings consistently outperformed GloVe and matrix3 embeddings, underscoring the value of contextualised word representations in dialogue tasks. The superior performance of BERT embeddings likely stems from their ability to capture context-dependent word meanings and complex linguistic phenomena, which are crucial in understanding and generating natural dialogue. However, the varying performance of each embedding method across different architectures suggests that the

choice of embedding technique should be carefully considered in conjunction with the model architecture for optimal performance.

The Encoder-Decoder Transformer model, while generally underperforming compared to the Extractor and Reencoder models, still showed potential, particularly with BERT embeddings. Its moderate performance across accuracy and TER metrics suggests that there may be room for improvement through fine-tuning or architectural modifications specifically tailored for dialogue modelling. The results also highlight the importance of considering multiple evaluation metrics in dialogue systems, as models may excel in one aspect (e.g., understanding, as reflected in accuracy) while lagging in another (e.g., response generation quality, as indicated by TER) (Table 11).

Experiments run on the MELD dataset consistently return very low or null BLEU scores. It can be argued that dialogue models trained on TV show conversations tend to score poorly on BLEU metrics across various architectures. This is likely because TV dialogue is inherently open-ended, with many possible valid responses to any given statement. The unpredictable nature of these conversations, including context-dependent replies and sudden topic shifts, makes it difficult for models to generate responses that closely match specific reference answers. As a result, even when a model produces a contextually appropriate and natural-sounding reply, it may diverge significantly from the expected response used in BLEU calculations. This leads to low BLEU scores, despite the potential conversational quality of the generated responses.

5.5.2 Results on the IEMOCAP Dataset (audio and text embeddings)

The Interactive Emotional Dyadic Motion Capture (IEMOCAP) dataset is a widely used resource in the field of emotion recognition and affective computing (Busso et al., 2008). Developed by researchers at the University of Southern California, IEMOCAP consists of approximately 12 hours of audiovisual data from dyadic interactions between actors (C.-C. Lee et al., 2011). The dataset includes recordings of both scripted and improvised scenarios, designed to elicit a range of emotional expressions. It features 10 actors (5 male and 5 female) paired in dyadic conversations, with their facial expressions, voice, and gestures captured using high-quality audio, video, and motion-capture technology (Metallinou et al., 2012). The emotional content is annotated at the utterance level, covering categorical emotions (such as anger, happiness, sadness, and neutral) as well as dimensional labels (valence, activation, and dominance) (Tripathi et al., 2019). IEMOCAP's multimodal nature, diverse emotional content, and high-quality annotations make it a valuable resource for developing and evaluating emotion recognition algorithms, particularly those leveraging speech and facial expressions (Zadeh et al., 2018). The dataset has been extensively used in research on speech emotion recognition, multimodal emotion analysis, and human-computer interaction studies (Neumann & Vu, 2019).

The following table presents our results; it is important to note that testing the IEMOCAP dataset against the TensorFlow baseline architecture was infeasible, as it would have necessitated substantial architectural modifications, deviating significantly from the original design.

***Table 12**. System evaluation on IEMOCAP Dataset with Audio and Text extracted using different embedding algorithms and performance measures.*

Architecture/ Embeddings	**Extractor**		**Reencoder**		**Encoder-decoder**		**baseline**	
	BERT	Matrix3	BERT	Matrix3	BERT	Matrix3	BERT	Matrix3
BLEU	0	0	0.004	**0.006**	0.004	0.006	-	-
METEOR	0.042	0.042	0.080	0.095	0.078	**0.097**	-	-
TER	**121.009**	253.850	166.326	259.786	135.952	245.007	-	-
Perplexity	4.137	4.478	**3.674**	3.944	3.766	3.923	-	-

This table offers a thorough analysis comparing various dialogue modelling structures, including Extractor, Reencoder, Encoder-decoder, and a baseline model, each paired with either BERT or Matrix3 embedding techniques. Based on the provided results, the performance of the different model-embedding combinations can be analysed and their strengths and weaknesses for the dialogue modelling task on the IEMOCAP dataset discussed.

The experimental results from our study do not indicate a clear superior performance among the three architectural designs evaluated. Each of the tested architectures - the Extractor, the Reencoder, and the Encoder-Decoder Transformer - demonstrated comparable effectiveness across the range of metrics employed. Although each architecture shows promising results in one or two metrics, not one architecture appears to clearly outperform the other in more than one metric (Table 12).

While the Extractor provides the best TER score, it also registers some of the worst perplexity scores across the board, for example. At the same time, the Reencoder architecture shows some of the best BLEU and METEOR scores, but also some of the worst TER scores.

Contrary to expectations and previous research findings, our experimental results revealed an intriguing outcome regarding the performance of embedding methods. Surprisingly, neither BERT nor Matrix3 consistently outperformed the other across the various dialogue modelling architectures and evaluation metrics. This observation stands in stark contrast to numerous other studies in the field, where BERT has typically emerged as a clear frontrunner in natural language processing tasks. The lack of a decisive advantage for BERT in our experiments challenges the prevailing notion of its superiority and suggests that the effectiveness of embedding methods may be more context-dependent than previously thought. This unexpected result underscores the importance of thorough comparative analyses in different application scenarios, as the optimal choice of embedding technique may vary depending on the specific dialogue modelling task, architecture, or evaluation criteria. Our findings highlight the need for a nuanced approach when selecting embedding methods for dialogue systems, and call for further investigation into the factors that influence their relative performance across diverse contexts.

5.6 Comparing our Multimodal Results with previous Research Findings

A comparative analysis between the proposed architectures and those detailed in the study by Young et al., (2020) titled “Dialogue systems with audio context” was conducted. This decision was motivated by the similarities in our research goals and the innovative approach presented in their work, particularly their Audio-Seq2Seq model which incorporates audio features into the dialogue generation process.

Young et al.'s study provides a solid foundation for exploring the integration of audio context in dialogue systems, demonstrating improvements in perplexity, response diversity, and human evaluation scores compared to text-only baselines. By benchmarking our multimodal architectures against their Audio-Seq2Seq model, this study aims to assess the relative strengths and potential areas for improvement in our approach. This comparison allows us to evaluate how effectively our models capture and utilise audio information in generating contextually appropriate and emotionally resonant responses. Additionally, it will provide insights into the generalizability of audio-augmented dialogue models across different datasets and task configurations, contributing to the broader understanding of multimodal dialogue systems in the research community.

Young et al., (2020) utilised the same multimodal datasets that were selected for our research, namely MELD (Multimodal EmotionLines Dataset) and IEMOCAP (Interactive Emotional Dyadic Motion Capture Database). This commonality in dataset selection provides a solid foundation for comparison, as it ensures that both studies are working with similar types of multimodal dialogue data, including textual content and corresponding audio features.

However, Young et al. introduced human evaluation as part of their performance metrics, which is not a component of our current study. Human evaluation, while valuable for assessing the qualitative aspects of generated responses, introduces subjective elements that are challenging to replicate precisely. Therefore, to maintain objectivity and ensure a fair comparison, the comparative analysis was conducted solely on the perplexity metric. Perplexity, being a quantitative measure of how well a probability model predicts a sample, offers a consistent and reproducible basis for comparing the performance of our multimodal architectures against the Audio-Seq2Seq model proposed in their study. This approach allows us to evaluate the predictive power of the models in a standardised manner, while acknowledging the limitations of not including the human-evaluated aspects of dialogue quality in our comparison.

In their study, Young et al. report using a test set of 1000 samples from the MELD dataset for their evaluation; similarly, they selected 901 sentences for their IEMOCAP test set. This research has also selected a test set of 1000 samples from MELD and 901 for IEMOCAP to maintain consistency in the evaluation scale. However, it's important to note that there is no guarantee that our test set is identical to the one used by Young et al. The authors did not provide specific details about their test set selection process. This potential difference in test sets introduces a degree of uncertainty in our comparison. While the size of the test sets is the same, the specific samples might differ, which could lead to slight variations in the reported perplexity scores between our study and theirs.

***Table 13**. Perplexity scores of Multimodal Dataset with Audio and Text extracted using different embedding algorithms and compared to Young et al., (2020).*

Architecture/ Embeddings	**Extractor**		**Reencoder**		**Encoder-decoder**		**Young et al.**
	BERT	Matrix3	BERT	Matrix3	BERT	Matrix3	
MELD	3.263	3.364	3.292	2.899	**2.867**	3.421	46.19 ± 0.49
IEMOCAP	4.137	4.478	**3.674**	3.944	3.766	3.923	31.13 ± 0.31

Our comparative analysis reveals that our proposed multimodal architectures consistently outperform the Audio-Seq2Seq model presented by Young et al. (2020) in terms of perplexity on both the MELD and IEMOCAP datasets. Specifically, our models demonstrate a significant reduction in perplexity scores, indicating a better ability to predict and generate contextually appropriate responses in multimodal dialogue scenarios. This improvement suggests that our architectures are more effective at integrating and leveraging the audio and textual information present in these datasets (Table 13).

Several factors could contribute to the superior performance of our models. Firstly, our architectures may employ more sophisticated embedding techniques for combining audio and textual features, allowing for a more nuanced understanding of the multimodal context. Secondly, the study might have implemented more advanced attention mechanisms that better capture the relevance of different modalities in varying dialogue contexts. Additionally, our models could benefit from more recent advancements in transformer-based architectures, which have shown remarkable capabilities in handling sequential data. The use of pre-trained language models as a starting point for the embedding layers in our architectures might also contribute to their enhanced performance. Finally, our approach to audio feature extraction and representation might be more refined, potentially capturing subtle audio cues that are particularly relevant to dialogue generation. These advancements collectively contribute to our models' improved ability to generate contextually appropriate and coherent responses in multimodal dialogue settings.

5.7 Comparing Audio, Text, and Multimodal models

Analysing the results of our dialogue modelling experiments across various architectures and input modalities, several intriguing patterns have emerged. Firstly, our findings consistently demonstrate that models trained exclusively on audio embeddings exhibit the poorest performance across all tested architectures. This observation suggests that audio embeddings, while valuable, may provide a less information-dense representation for dialogue modelling compared to textual data. The limited performance of audio-only models suggests that crucial semantic and contextual information present in linguistic content may be lost or obscured when relying solely on acoustic features. This finding aligns with previous research that suggests audio embeddings, while valuable, may provide a less information-dense representation for dialogue modeling compared to textual data. For instance, (Vogel et al., 2023) emphasize the limitations of audio data in capturing nuanced dialogue dynamics, indicating that crucial semantic and contextual information inherent in linguistic content may be lost when relying solely on acoustic features (Vogel et al., 2023).

Interestingly, our experiments reveal that models trained on text embeddings alone outperform those utilising a combination of text and audio embeddings. This counterintuitive result challenges our initial assumption that multimodal input would invariably lead to superior performance. A similar observation was made by Y. Zhao et al., (2023), who found that integrating textual and emotional information in dialogue systems does not always yield expected improvements, suggesting that the complexity of multimodal data can sometimes hinder performance rather than enhance it (Y. Zhao et al., 2023). Upon closer examination, this phenomenon can be attributed to two primary factors. Firstly, the discrepancy in dataset sizes between our text-only corpus and the multimodal dataset likely plays a significant role. The larger text-only dataset provides more diverse examples and contexts for the model to learn from, potentially offsetting the added richness of multimodal representations. This is consistent with the work of B. Zhang et al., (2023), who noted that larger datasets often lead to improved model performance in multimodal settings (B. Zhang et al., 2023). Secondly, the increased complexity and dimensionality of the combined text and audio embeddings may require a substantially larger dataset for the model to fully leverage these rich representations effectively.

Below are some examples of text-only results compared to audio-only results and audio and text results from the IEMOCAP dataset.

Text only:

Input: How you doing? Aw, you know what I forgot? I forgot to bring my flashlight. Oh, how could I be so stupid? I should've brought my flashlight.

Output: [CLS] What flashlight?

Input: What flashlight?

Output: [CLS] The flashlight ; the silver one. There ' s only one, isn ' t there?

Input: You know, the flashlight, the silver one. There's only one isn't there?

Output: [CLS] It ' s not your flashlight.

Input: That's not your flashlight.

Output: [CLS] Whose is it?

Input: Who's is it?

Output: [CLS] It ' s ours. It ' s mine too.

Input: It's ours. It's mine too.

Output: [CLS] Well sure, yes. It ' s been yours. It ' s been yours.

Input: Oh, well sure.

Output: [CLS] You kept saying my flashlight as if it ' s only yours. How is that supposed to make me feel?

Input: You kept saying my flashlight, if as if somehow it's just yours. I mean is that supposed to make me feel?

Output: [CLS] No. No. Of course our flashlight yours and mine. Mi flashlight, es su flashlight, naturally. How could we not think to bring it? I hope the moon stays out.

Audio only:

Input: Oh, no you know what I did? I forgot my flashlight. How could I be so stupid I forgot my flashlight.

Output: ,

Input: What flashlight?

Output: ,

Input: The flashlight, the silver one. There's only one isn't there?

Output: ,

Input: It's not yours.

Output: , ,

Input: Whose is it?

Output: ,

Input: It's ours. It's mine too.

Output: , , ,

Input: Well sure.

Output:

Input: You kept saying my flashlight as if it somehow is just yours. How's that supposed to make me feel?

Output:

Input: No of course our flashlight, yours and mine, mi flashlight es su flashlight naturally. How could we not think to bring it?

Output: , ,

Text and Audio:

Input: This is great. How are you doing? Oh, no. Oh, do you know what I did? I forgot the flashlight. How could I be so stupid? I forgot the flashlight.

Output: [CLS] I know't t...

Input: What flashlight?

Output: [CLS] I know't t..

Input: The flashlight. The silver one. There's only one, isn't there? Stupid.

Output: [CLS] I know't t t.....

Input: You mean our flashlight?

Output: [CLS] I know't t t t

area could focus on developing more sophisticated methods for audio feature extraction and multimodal fusion to better leverage the complementary strengths of both audio and textual inputs in dialogue generation.

5.8 Best Performing Architecture and Embedding Layers

The research findings consistently demonstrate the superior performance of the Reencoder architecture across various dialogue modelling tasks. This novel architecture, which incorporates an additional re-encoding step, outperformed other tested models including the baseline architecture, Encoder-Decoder Transformer, and Extractor model. The Reencoder's ability to iteratively refine input representations allowed it to capture and integrate more nuanced contextual information from previous conversation turns, resulting in more coherent and contextually appropriate responses. This consistent top performance was observed across diverse datasets such as Meld, Cornell, OpenSubtitles, and DailyDialog, with the Reencoder model achieving lower TER scores, higher BLEU scores, and superior accuracy compared to its counterparts.

Among the embedding methods tested, two stood out as particularly effective: BERT embeddings and the custom embedding method referred to as Matrix3, which was learned directly from the training data. Both of these embedding approaches contributed significantly to the models' performance, with the Reencoder architecture benefiting most notably from their use. The effectiveness of these embedding methods can be attributed, in part, to their use of subword tokenization, a technique that allows for more flexible and nuanced representation of words and their components. This finding is consistent with the work of Wolf et al., (2023), who emphasizes the advantages of multimodal language modeling that incorporates advanced tokenization strategies (Wolf et al., 2023).

It's important to highlight that both BERT and Matrix3 embeddings utilise subword tokenizers. This approach to tokenization breaks words down into smaller units, allowing the model to handle out-of-vocabulary words more effectively and capture morphological nuances. The use of subword tokenization enables these embedding methods to create more robust and adaptable representations of language, which in turn contributes to the overall performance of the dialogue models, particularly the Reencoder architecture. This observation aligns with the findings of (M. Zhang et al., 2024), who also noted the importance of effective tokenization in enhancing model performance in multimodal contexts (M. Zhang et al., 2024). This finding underscores the importance of choosing appropriate embedding methods and tokenization strategies in developing effective dialogue modelling systems.

5.9 Unexpected Findings

The experiments revealed several counterintuitive findings that warrant discussion. First, contrary to the common assumption that larger training datasets inherently lead to better performance, it was observed that increasing the volume of training data did not necessarily improve results when the additional data failed to meet certain quality thresholds. This suggests that data quality may be more crucial than quantity for effective dialogue modeling - a finding that challenges the prevalent 'more data is better' paradigm in machine learning.

Second, it was discovered that seemingly minor inconsistencies in data quality and formatting had disproportionate negative effects on model performance. These

inconsistencies, which included close captioning or variations in dialogue turns (example 1), resulted in significant degradation of output quality. This heightened sensitivity to data irregularities highlights the importance of maintaining strict data standardization protocols in dialogue modeling tasks.

Perhaps most surprisingly, the models demonstrated greater difficulty in handling everyday conversational dialogue compared to their performance on topic-specific queries. While the models showed competence in responding to focused, domain-specific questions, they struggled to maintain coherence in casual conversations that involved social niceties, context switching, and implicit social cues (as shown in examples 2 and 3 below). This finding suggests that current approaches to dialogue modeling may be better suited to structured, domain-specific interactions rather than open-ended social conversation, despite the latter being computationally simpler from a human perspective.

Example 1: this example showcases two turns in the conversation that haven't been correctly identified in the data (the input should be "*hey , uh , is it okay to come in ?*" and "*of course !*" should be part of the output).

Input: hey , uh , is it okay to come in ? of course !

Output: [CLS] wow.

Example 2:

Input: How could we forget to bring it? I hope the moon stays out.

Output: [CLS] I wonder do they always run during a full moon? I can ' t remember if it was full last year, do you remember?

Example 3:

Input: next is sports news quiet very quiet

Output: i have seen the ladies and gentlemen s of the tower of babel .

These unexpected findings have important implications for future research directions in dialogue modeling, particularly regarding data curation strategies and architectural decisions for handling different types of conversational contexts.

5.10 Summary

Based on the experimental results presented in this chapter, several significant conclusions about multimodal dialogue modelling using the MELD dataset, which combines text transcriptions with audio embeddings, can be drawn. The study compared three main architectures — Extractor, Reencoder, and Encoder-Decoder Transformer — across different embedding methods including BERT, GloVe, and matrix3.

The findings reveal that the Reencoder model with BERT embeddings demonstrated a better balance between accuracy and response quality, as measured by the Translation Error Rate (TER). Across all architectures, BERT embeddings and matrix3 embeddings consistently outperformed GloVe, highlighting the value of contextualised word representations in dialogue tasks. The Encoder-Decoder Transformer model, while generally underperforming compared to the other architectures, showed potential for improvement through fine-tuning or architectural modifications. This is consistent with the observations made by (Du, 2024), who noted that enhancements in architecture can lead to significant performance gains in dialogue summarization tasks (Du, 2024).

These results underscore the complex nature of dialogue modelling, where understanding input and generating appropriate responses present distinct but interconnected challenges. The incorporation of audio embeddings alongside text data proved valuable in enhancing the models' ability to capture nuances in emotion, tone, and context that may not be apparent from text alone. This multimodal approach, while focused solely on the MELD dataset due to its unique characteristics, provides insights into the potential of integrating audio features with textual information to improve dialogue systems' performance in tasks such as emotion recognition and intent classification.

Chapter 6: Discussion

6.1 Introduction

This section analyzes the findings from the experiments and highlights the strengths and limitations of each transformer architecture. The section also discusses the implications of the results and suggests potential avenues for future research in the field of chatbot modelling using transformer architectures.

The study conducted a series of experiments involving several transformer architectures, including a baseline architecture, Encoder-Decoder Transformer, Extractor model adapted from previous experiments in the area, and a novel Reencoder architecture aimed at better modelling contextual information at training time. Each architecture was trained and evaluated on various datasets comprising dialogue transcripts sourced from diverse domains, encompassing both formal and informal conversational styles. The training data were preprocessed to ensure uniformity and compatibility across architectures, with tokenization and data augmentation techniques applied as necessary. Each Architecture was then also modified to be able to learn from audio, and audio and text embeddings as well. These audio and multimodal architectures were trained on the MELD and IEMOCAP datasets, extracting textual information from audio data through speech recognition for the multimodal architectures.

Following rigorous experimentation and comprehensive evaluation of each architecture's performance across multiple metrics such as METEOR, TER, and BLEU score, our findings revealed a notable trend: while the performance metrics remained consistent across all studied architectures, the absolute values were unexpectedly lower than anticipated. This observation prompted further investigation into potential factors contributing to the discrepancy between expected and observed performance levels.

Despite employing diverse evaluation methodologies and benchmarking against established metrics, the discrepancy in absolute performance metrics suggests underlying complexities within the datasets or model architectures that may not have been fully accounted for during the experimental design phase. Several factors could contribute to this discrepancy, including the smaller dimensionality of the models presented, and discrepancies between data size and quality.

The forthcoming sections will delve into an exploration of potential factors that could elucidate this observed lower absolute performance, contextualising these factors within prevailing trends and the trajectory of future research within the field.

6.2 Architecture Comparison Performance

In our extensive exploration of various transformer architectures for dialogue modelling, the Reencoder architecture consistently emerged as the top performer across multiple evaluation metrics and datasets. This robust performance was particularly striking given the diversity of datasets used, including MELD, Cornell, OpenSubtitles, and DailyDialog. Across these datasets, the Reencoder consistently exhibited lower TER scores and higher BLEU scores compared to other transformer architectures. Notably, this trend persisted even when evaluated using different embedding layers, indicating the intrinsic efficacy and versatility of the Reencoder architecture in dialogue modelling tasks.

One possible explanation for the superior performance of the Reencoder architecture lies in its unique design, which emphasises the iterative refinement of input representations through a reencoding mechanism. Unlike traditional transformer architectures that rely solely on self-attention mechanisms for encoding contextual information, the Reencoder architecture introduces an additional reencoding step, which allows for the iterative refinement of input representations. This iterative refinement process enables the model to capture and integrate increasingly nuanced contextual information, leading to more accurate and coherent dialogue generation. This discovery underscores the pivotal significance of embedding layers within the architecture of a language model, shedding light on their crucial role in shaping and enhancing the model's performance and capabilities

The Reencoder model represents a significant advancement in dialogue modelling due to its unique approach to leveraging contextual information from previous turns in a conversation. By incorporating historical context into the generation of vector representations for sentences within the embedding space, the Reencoder model achieves heightened levels of contextual awareness compared to traditional language models. This contextual enrichment allows the model to capture subtle nuances and dependencies inherent in human dialogue, leading to more accurate and contextually appropriate responses. As a result, the Reencoder model demonstrates enhanced performance in modelling human dialogue, as it effectively captures the dynamic nature of conversations and adapts its responses based on the evolving context over the course of the interaction.

Operating on the contextual representation of language enables the Reencoder model to encode not only the immediate input, but also the broader context provided by previous turns in the conversation. This holistic approach to contextual modelling empowers the model to generate responses that are not only syntactically correct but also semantically coherent within the larger discourse context. By considering the entire conversational history, the Reencoder model can infer implicit information, anticipate user intents, and maintain consistency in dialogue interactions, thereby enhancing the overall conversational quality and user experience.

Furthermore, the Reencoder model's ability to create contextually aware vector representations of sentences within the embedding space contributes to its versatility and effectiveness across a wide range of dialogue scenarios. Whether handling short, task-oriented exchanges or engaging in longer, more open-ended conversations, the model's contextual understanding enables it to produce responses that are contextually relevant and linguistically fluent. This enhanced performance in modelling human dialogue underscores the significance of leveraging contextual information in language modelling tasks, highlighting the potential of the Reencoder model to advance the state of the art in conversational AI and natural language understanding.

Moreover, the Reencoder architecture's superior performance may also be attributed to its ability to leverage the inherent advantages of small language models when operating on relatively small datasets. Small language models, characterised by their compact size and simplified architectures, have been shown to exhibit greater flexibility and adaptability when trained on limited data, as shown by results provided by the DailyDialog dataset. In the context of dialogue modelling, where datasets may be relatively small compared to other NLP tasks, the Reencoder's ability to learn from previous turns in the conversation, allows it to capture more generalizable patterns and relationships within the data. Overall, the combination of the Reencoder's unique architecture and the advantages of small language

models likely contributes to its superior performance and consistency in dialogue modelling tasks.

6.3 The effect of dataset quality, size, and model complexity on performance

6.3.1 Text only models' training on Dailydialog and IEMOCAP

Through a series of experiments comparing various transformer architectures, it was consistently observed that models trained on the DailyDialog and IEMOCAP datasets exhibited better performance and greater stability than those trained on the other datasets. This section provides insights into the underlying factors contributing to the enhanced performance of DailyDialog and IEMOCAP datasets and explores the implications for dialogue modelling research and applications.

Although it is well known that Transformer architectures have emerged as state-of-the-art models for dialogue modelling, offering unparalleled performance in capturing contextual dependencies and generating coherent responses, it appears that the choice of training dataset significantly influences the effectiveness of transformer models in dialogue modelling tasks. In recent years, the DailyDialog and IEMOCAP datasets have gained prominence as a benchmark dataset for dialogue modelling, characterised by their diversity, quality, and relevance to real-world conversational scenarios. This section aims to investigate the superior performance of transformer models trained on said datasets and elucidate the underlying reasons behind this phenomenon.

The experiments highlighted that Encoder-Decoder Transformer models trained on the DailyDialog dataset outperformed those trained on alternative datasets in terms of performance metrics such as perplexity, BLEU score, METEOR, and TER. On the other hand, Reencoder and Extractor models trained on the IEMOCAP dataset exhibited lower TER scores and higher BLEU and METEOR scores than models trained on other datasets, indicating better comprehension of dialogue contexts and more coherent response generation. Furthermore, the performance of models trained on DailyDialog and IEMOCAP datasets remained stable across different evaluation metrics and transformer architectures, highlighting the robustness and consistency of the dataset in facilitating effective dialogue modelling.

Several factors are likely contributing to the superior performance of transformer models trained on the DailyDialog and IEMOCAP datasets. Firstly, the DailyDialog and IEMOCAP datasets are characterised by their high-quality, diverse, and contextually rich dialogues. These features provide ample training examples for learning complex dialogue patterns and linguistic nuances. The richness and diversity of the dataset enable transformer models to generalise well to unseen dialogues and handle various conversational scenarios effectively (J. Lee & Lee, 2022). Secondly, the relatively smaller size of these datasets compared to larger corpora allows transformer models to focus on learning relevant dialogue patterns without being overwhelmed by irrelevant or noisy data. This facilitates more efficient learning and better generalisation capabilities, leading to improved performance in dialogue modelling tasks. These results seem to underline a trend,

according to which there would be a linear correlation between the dimensionality of a language model, and the amount of data necessary for its training. Smaller language models, with less trainable parameters, would benefit from a small, highly curated dataset, making them more suited for specific tasks (Althnian et al., 2021). Finally, the inherent structure and coherence of dialogues in the DailyDialog and IEMOCAP datasets contribute to the effectiveness of transformer models in capturing contextual dependencies and generating coherent responses. Overall, the combination of high-quality data and the suitability of small language models for learning from small datasets contributes to the superior performance and consistency of transformer models trained on the DailyDialog and IEMOCAP datasets in dialogue modelling tasks.

6.3.2 Factors Contributing to Low Bleu Performance with the MELD Dataset

The MELD (Multimodal EmotionLines Dataset) is a unique and challenging corpus for dialogue modelling tasks. It consists of utterances from television show transcripts, specifically from the show Friends. The utterances are labelled with emotion categories and sentiment polarities, making it valuable for modelling emotional and pragmatic aspects of dialogue. However, the experiments conducted with various transformer architectures on this dataset yielded very poor BLEU scores, with most models scoring around zero.

There are a few potential reasons for these underwhelming BLEU scores on MELD. First and foremost, the dataset is relatively small, containing only around 13,000 utterances in total. This limited data may not provide enough examples for large language models to effectively learn patterns of natural dialogue flow and emotional nuance. Additionally, the sitcom dialogue in MELD often contains colloquial language, humour, and contextual references that can be difficult for models to fully comprehend and generate.

Another key factor is that BLEU, while a popular automatic evaluation metric, may not be well-suited for assessing dialogue model performance on MELD. Since the dataset contains multi-turn conversations with potential for multiple valid responses to each utterance, models are effectively being penalised by BLEU for not precisely matching the provided reference. In dialogue, preserving semantic coherence and maintaining a natural flow is often more important than strict lexical similarity.

Despite the low BLEU scores, the transformer models may still be capturing valuable dialogue traits from MELD that are not reflected in this metric. Future work could explore other automatic and human evaluation strategies that better measure pragmatic and emotional aspects of dialogue generation. Additionally, combining MELD with larger dialogue corpora during training could help models leverage the unique emotional annotations while benefiting from more general dialogue patterns in the larger datasets.

6.4 Embedding layers and model performance

This section investigates the influence of Bert embedding layers on transformer architectures for dialogue modelling tasks. Through a series of experiments, various transformer architectures were evaluated, and it was consistently observed that models utilising Bert embedding layers exhibited superior performance and greater consistency compared to those employing alternative embedding strategies. This study analyses insights into the mechanisms underlying the enhanced performance of Bert embedding

layers and explores potential factors contributing to their effectiveness in dialogue modelling tasks.

Dialogue modelling represents a fundamental task in natural language processing, with applications spanning chatbots, virtual assistants, and conversational agents. Transformer architectures have emerged as prominent models for dialogue modelling, offering flexibility, scalability, and effectiveness in capturing the contextual nuances of human conversation. However, the choice of embedding layer within transformer architectures significantly impacts model performance and generalisation capabilities. In recent years, Bert (Bidirectional Encoder Representations from Transformers) embedding layers have garnered attention for their ability to capture bidirectional contextual information, leading to improvements in various NLP tasks.

A comprehensive series of experiments was conducted to evaluate the performance of transformer architectures with different embedding layers on dialogue modelling tasks.

Across all experiments conducted by training and evaluating different deep learning models on various datasets, models incorporating Bert embedding layers consistently outperformed those utilising alternative embedding layers (GloVe and Embeddings learned from the data). Notably, the Encoder-Decoder Transformer model equipped with a Bert embedding layer demonstrated the highest performance across all datasets, achieving lower perplexity scores, and higher BLEU scores compared to other model configurations. Similarly, the Reencoder and Extractor models exhibited enhanced performance when coupled with Bert embeddings, indicating the robustness and versatility of Bert embedding layers across different transformer architectures.
The only exception to this trend emerges with the Reencoder model trained on the DailyDialog dataset, which showcases superior performance when enhanced by the SubwordTokenizer. Across metrics such as TER, BLEU, and METEOR, the Reencoder model consistently demonstrates enhanced efficacy. Notably, although results across architectures and embedding layers are comparable, the Reencoder model paired with the SubwordTokenizer outshines others when evaluated on the DailyDialog dataset. Intriguingly, despite the Reencoder being the best-performing architecture, its synergy with the best-performing dataset does not yield the absolute best results with Bert embedding layer, indicating the nuanced dynamics between model architecture, dataset choice, and embedding method.

It is noteworthy to consider the underlying mechanisms of the SubwordTokenizer and its associated embedding layer, particularly in contrast to pre-trained tokenizers and embedding layers like Bert. Unlike Bert, which relies on pre-training on vast corpora to capture linguistic patterns and relationships, the SubwordTokenizer and its embedding layer are tailored directly to the specific dataset being used. This bespoke approach allows for a more fine-grained representation of the textual data, as the tokenizer is optimised to handle the unique linguistic nuances and vocabulary present within the dataset. Consequently, the embedding layer can capture more subtle semantic relationships and contextual information, leading to potentially richer representations of the text for language modelling tasks (L. Xue et al., 2022; Bostrom & Durrett, 2020).

The superiority of the SubwordTokenizer and its associated embedding layer over pre-trained alternatives like Bert could stem from their ability to capture dataset-specific intricacies more effectively. By directly fitting the tokenizer and embedding layer to the data, the model can better adapt to the idiosyncrasies of the dataset, resulting in more accurate and contextually relevant representations of the text. This tailored approach is particularly advantageous for smaller datasets like DailyDialog, where the linguistic characteristics may vary significantly from broader corpora used in pre-training Bert. Consequently, the SubwordTokenizer and its associated embedding layer offer a more tailored and contextually relevant representation of the data, which could contribute to the observed improvements in model performance.

Moreover, the observation that Bert's higher dimensionality may be better suited for larger datasets and language models warrants further investigation. While Bert's extensive pre-training on large-scale corpora endows it with robust linguistic knowledge, its high-dimensional embeddings may introduce challenges when applied to smaller datasets and language models. The richer feature space provided by Bert's high-dimensional embeddings may require larger volumes of data to effectively capture and generalise linguistic patterns, rendering it less optimal for smaller-scale tasks. In contrast, the lower dimensionality of the SubwordTokenizer's embeddings may offer a more compact yet expressive representation that is better aligned with the constraints of smaller datasets and language models. Thus, the choice between Bert and dataset-specific tokenizers and embeddings should be carefully considered in light of the dataset size and task requirements.

Furthermore, the utilisation of Bert embeddings exhibits notable advantages in fostering greater consistency in model performance across diverse evaluation metrics. By leveraging Bert embeddings, the models exhibit heightened stability and reliability in dialogue modelling tasks. The robustness provided by Bert embeddings ensures that the models consistently deliver reliable performance across different evaluation criteria, thereby bolstering the overall efficacy and trustworthiness of the dialogue generation process. Moreover, Bert embeddings offer inherent advantages in capturing contextual dependencies and linguistic nuances present in dialogue data, thereby facilitating more accurate and contextually relevant responses. Overall, the utilisation of Bert embeddings enhances the coherence and effectiveness of dialogue modelling, contributing to the overall quality and reliability of the generated responses.

The observed superiority of Bert embedding layers in dialogue modelling tasks can be attributed to several factors. Firstly, Bert embeddings capture bidirectional contextual information, enabling models to effectively understand and generate coherent dialogue responses. By leveraging the pre-trained knowledge encoded within Bert embeddings, transformer architectures can efficiently capture semantic nuances and linguistic intricacies present in dialogue datasets. Additionally, Bert embeddings are trained on large-scale corpora, encompassing diverse linguistic contexts and domains, which enhances their ability to generalise to unseen data and mitigate overfitting. Furthermore, the fine-tuning capabilities of Bert embeddings allow transformer architectures to adapt to specific dialogue modelling tasks, further improving performance and robustness.

In conclusion, this study found compelling evidence of the significant impact of Bert embedding layers on transformer architectures for dialogue modelling tasks. The consistent superiority and enhanced consistency of models utilising Bert embeddings underscore their effectiveness in capturing contextual information and generating coherent dialogue responses. However, an intriguing exception to this trend is observed with the Reencoder model trained on the DailyDialog dataset, which showcases superior performance when enhanced by the SubwordTokenizer. Despite this anomaly, it's notable that Bert embeddings facilitated greater consistency in model performance across various evaluation metrics, contributing to improved stability and reliability. The findings of this study shed light on the importance of embedding layer selection in transformer architectures and highlight the potential of Bert and other forms of pretrained embeddings in advancing dialogue modelling capabilities. Future research directions may explore novel techniques for leveraging Bert embeddings in dialogue modelling tasks and investigate their applicability across diverse domains and languages.

6.5 Dataset impact on model performance

Upon conducting extensive experiments and evaluating the performance of each architecture across various metrics, including METEOR, TER, BLEU score, and accuracy, comparable results across all studied metrics were observed. However, a striking observation emerged: the absolute performance metrics appeared lower than expected.
Several factors may contribute to the observed lower absolute performance metrics:

Small Language Models:
Our experimentation opted to employ relatively small language models, primarily driven by resource constraints and computational limitations. These constraints necessitated the use of smaller variants of transformer architectures, which, while potentially limiting the models' capacity to fully capture the intricacies of human dialogue, offered certain advantages. One notable advantage is the ability of smaller models to be trained more efficiently on specific tasks using smaller amounts of high-quality data. Despite the potential limitations in generalisation, our observations suggest that a modest increase in the size of the architecture coupled with training on a slightly larger dataset of high-quality data could yield significant improvements in model performance.
By incrementally enhancing the model's size and training data, researchers anticipate achieving higher-quality results while still maintaining a low computational cost for both training and inference. This approach allows for the development of efficient and accessible language models that strike a balance between performance and resource efficiency. It is worth noting that even with these incremental enhancements, the computational requirements remain modest, making the models accessible to a broader audience of researchers and practitioners.
For comparative purposes, it is noteworthy to mention that the largest model examined in our research, the Reencoder model featuring a BERT embedding layer, comprised 84,176,196 parameters. This model could be trained on a single Tesla T4 GPU paired with 16 CPU units, requiring a total of 104 gigabytes of system RAM. Compared to some Open Source Language models currently present in the industry, the architectures proposed show great potential for efficient dialogue modelling, while maintaining a fraction of the parameters. In order to offer a more specific comparison, ChatGPT employs one of the smaller GPT models, comprising an estimated 20 billion parameters (Singh et al., 2023).

The largest model submitted in this research has a dimensionality equal to 0.4% of ChatGPT's. Therefore, our experiments demonstrate that smaller models can offer competitive performance with substantially reduced resource requirements, underscoring their potential for widespread adoption and practical utility in various applications, in accordance with recent literature on the topic. Further details about the advantages offered by smaller language models are discussed later in the chapter.

Small Datasets: Another significant factor contributing to the observed lower absolute performance metrics is the utilisation of relatively small datasets for training the models. Despite meticulous efforts to curate and preprocess the training data, the inherent limitations posed by the dataset sizes may have impeded the models' capacity to effectively learn robust dialogue representations. While smaller datasets are typically deemed adequate for training smaller-sized language models, it is plausible that the volume of data utilised for this research was insufficient to adequately train the proposed models to their full potential.

To put this into perspective, consider the vast contrast in scale between the datasets used in this study and those employed in training larger, state-of-the-art language models such as ChatGPT. ChatGPT, for instance, has been trained on a corpus comprising approximately 570 gigabytes of data, enabling it to glean insights from a diverse array of linguistic contexts and nuances. In contrast, the largest dataset utilised in our study, OpenSubtitles, consisted of approximately 24 gigabytes of data, significantly smaller in comparison. This discrepancy in dataset size underscores the potential limitations imposed by the relatively modest volume of training data available for our experiments.

Given the pivotal role of data quantity in shaping the efficacy and performance of language models, the constrained size of the training datasets may have hindered the models' ability to learn intricate dialogue patterns and nuances effectively. Consequently, despite rigorous efforts to optimise model architectures and training methodologies, the restricted amount of training data may have posed a bottleneck, limiting the models' overall performance and generalisation capabilities. Moving forward, future research endeavours would benefit from leveraging larger and more diverse datasets to train language models, thereby affording models ample exposure to varied linguistic contexts and facilitating more comprehensive learning. By prioritising the acquisition and utilisation of expansive training datasets, researchers can enhance the robustness and effectiveness of language models, ultimately advancing the state of the art in natural language understanding and generation.

Data Quality: Furthermore, it is imperative to consider the influence of training data quality on the performance of transformer architectures. Despite meticulous efforts to maintain data cleanliness and consistency, it is inevitable that some datasets may contain noise or inaccuracies, which could significantly impede the models' learning process. In line with this conjecture, our observations indicate a noteworthy trend: models trained on the DailyDialog dataset consistently outperform those trained on other datasets, regardless of architecture or performance metrics used. This phenomenon likely stems from the superior quality and curation of the DailyDialog dataset compared to others examined in our research.

Despite its relatively smaller size compared to other datasets in our study, the DailyDialog dataset stands out for its meticulous curation and high-quality data. This attention to detail ensures that the training data is representative of natural human dialogue, thereby facilitating more effective learning and generalisation by the transformer architectures. The

abundance of carefully curated examples in the DailyDialog dataset likely mitigates the impact of noise and inaccuracies, allowing models trained on this dataset to achieve superior performance across various tasks and metrics.
Thus, the observed performance disparities among models trained on different datasets underscore the critical importance of data quality in transformer architecture training. Moving forward, continued emphasis on data curation and quality assurance processes will be essential to maximise the efficacy and performance of transformer-based models in real-world applications. By prioritising the use of high-quality, curated datasets such as DailyDialog, researchers and practitioners can enhance the robustness and reliability of transformer architectures, ultimately leading to more accurate and effective natural language understanding and generation capabilities. These findings appear to align with prevailing trends observed in other research work conducted within the field, thereby reinforcing the validity of the phenomenon commonly referred to as “garbage in, garbage out” (Rose & Fischer, 2011).

Implications and Future Directions
The results obtained from our experiments highlight the multifaceted nature of factors influencing the evaluation of transformer architectures for dialogue modelling. Among these considerations, the size of the language model and the dataset emerge as pivotal determinants of model performance. While our findings shed light on the performance of smaller language models on comparatively small datasets, further investigations into the scalability of language models and datasets are warranted to elucidate their influence on dialogue modelling efficacy comprehensively. Additionally, enhancing the quality and diversity of training data stands out as a critical avenue for bolstering model robustness and generalisation capabilities. Addressing data quality issues and diversifying datasets are essential steps toward mitigating potential biases and enhancing the model's capacity to capture the richness and variability inherent in natural language conversations. Future research endeavours should prioritise these considerations to advance the effectiveness and applicability of transformer architectures in dialogue modelling tasks.

6.6 Effect of Tokenization on the Embedding Layer of Large Language Models

The choice of tokenization technique directly impacts the characteristics and quality of embeddings learned by large LMs. Simple tokenizers like tfds.deprecated.text.Tokenizer may lead to coarse representations, particularly in scenarios involving complex languages or domain-specific jargon. Subword-based tokenization methods such as tfds.deprecated.text.SubwordTextEncoder offer improved coverage and flexibility, enabling the representation of rare or unseen words through subword composition. In contrast, BERT Tokenizer leverages subword units and bidirectional context to generate embeddings tailored for transformer-based architectures, enhancing the model's ability to capture nuanced language semantics (Liu et al., 2019).

The machine learning experiments explored three different embedding layers tailored to specific tokenization methods, optimising the encoding of textual data. Firstly, an automated embedding matrix generated directly from the data was paired with the tfds.deprecated.text.SubwordTextEncoder tokenizer, adept at handling out-of-vocabulary

words and morphological variations. Secondly, GloVe embeddings, capturing semantic relationships between words, were coupled with a custom-made tokenizer to ensure alignment with GloVe's vocabulary and dimensions. Lastly, BERT embeddings, capable of capturing contextual information, were employed alongside the BERT Tokenizer, specifically tailored to BERT's vocabulary and tokenization schema. By adopting distinct tokenization strategies for each embedding method, this study ensured optimal preprocessing and encoding of textual data, maximising the effectiveness of each embedding layer in our experiments.

The research findings indicate that subword tokenizers, such as the TensorFlow SubwordTextEncoder used for our Matrix3 experiments, and the tokenizer used in BERT, play a crucial role in enhancing the performance of dialogue modelling systems. These tokenization methods demonstrate a marked improvement in model performance compared to traditional word-level tokenization approaches. By breaking words down into smaller, meaningful units, subword tokenizers enable models to handle out-of-vocabulary words more effectively, capture morphological nuances, and create more flexible representations of language. Bostrom and Durrett highlight that subword tokenization allows for the decomposition of rare words into smaller, more manageable units, which facilitates better handling of out-of-vocabulary (OOV) words and captures morphological nuances (Bostrom & Durrett, 2020). This is particularly advantageous in dialogue modelling, where understanding and generating diverse language constructions is essential.
Moreover, subword tokenizers create more flexible representations of language by breaking down words into smaller, meaningful units. This granularity enables models to develop context-aware token representations, which are crucial for capturing the subtleties and variations in natural language conversations. Xue et al. emphasize that subword tokenization minimizes the total length of token sequences while maintaining a fixed vocabulary size, thereby enhancing the model's ability to generate coherent responses (L. Xue et al., 2022). The ability to generate flexible and contextually relevant representations is further supported by findings from Minixhofer, who discusses the effectiveness of subword tokenization in improving language model performance through better morphological representation (Minixhofer et al., 2023).

The improved performance observed with subword tokenizers underscores the importance of tokenization strategy in developing robust dialogue models. As noted by Peters and Martins, subword-level morpheme segmentation is increasingly recognized as a vital component in modern NLP systems, which often rely on sequences of subword units induced by unsupervised algorithms like byte-pair encoding (BPE) (B. Peters & Martins, 2022). This observation suggests that future research and development efforts in dialogue modeling should prioritize the use of subword tokenizers to leverage their advantages in handling complex linguistic structures.

6.7 Interplay between Model Dimensionality, Data Size, and Task Specificity

This section explores the emerging correlation between the dimensionality of a language model (LM) and the data requirements for its effective training in dialogue modelling tasks.

The section posits that smaller LMs, characterised by a lower number of trainable parameters, exhibit optimal performance when trained on compact, meticulously curated datasets. This targeted approach aligns well with the specific demands of dialogue modelling tasks.

6.7.1 The Dimensionality-Data-Task Landscape

Recent empirical findings indicate a discernible trend in the relationship between the dimensionality of language models (LMs) and the requisite amount of data for effective training (Kaplan et al., 2020). This trend suggests a linear correlation, wherein the dimensionality of an LM is directly proportional to the volume of data needed for successful training. Understanding this correlation sheds light on crucial aspects of LM training and deployment, offering valuable insights into optimising model performance and resource utilisation.

As shown in our results, smaller language models exhibit a notable advantage in terms of data efficiency, as evidenced by their lower data requirements for achieving optimal performance. Due to their reduced size and complexity, these models possess fewer trainable parameters, making them inherently more adept at learning from smaller, meticulously curated datasets. This efficiency stems from their limited capacity to capture and learn complex patterns from vast amounts of data, allowing them to effectively leverage the information contained within smaller datasets for robust dialogue modelling tasks.

Moreover, the correlation between LM dimensionality and data requirements underscores the importance of focus and efficiency in dataset curation for dialogue modelling. Curated datasets tailored to specific dialogue domains provide smaller LMs with a focused training environment, allowing them to specialise in understanding and responding to the nuances inherent in that domain. By honing in on domain-specific characteristics and linguistic subtleties, these datasets enable smaller LMs to achieve higher levels of accuracy and relevance in dialogue generation tasks, ultimately enhancing the efficiency of the training process and facilitating deployment on resource-constrained devices, as demonstrated in our study by the superior performance showcased by models trained on the IEMOCAP and DailyDialog datasets.
This targeted approach fosters efficient training and facilitates the deployment of these models on resource-constrained devices due to their inherent efficiency.

The experiments conducted across different proportions of the OpenSubtitles Dataset reveal a consistent trend: augmenting the size of the training data does not result in significant performance enhancements when evaluated using various metrics. Surprisingly, even with the data expanded by a factor of ten, there is no substantial improvement in performance observed across the metrics employed in the evaluation process.

The observed challenges with performance improvement despite increased training data size could be attributed to various factors that interact to impede the learning process. Firstly, the quality of the OpenSubtitles dataset itself may be subpar, containing noise, inconsistencies, and irrelevant information that complicates the learning task for models. This necessitates further investigation to precisely understand how dataset quality impacts model performance and to develop strategies to mitigate its adverse effects.

Moreover, the suitability of large datasets for smaller language models warrants consideration. While large datasets offer rich and diverse data, smaller models may struggle to extract meaningful patterns and insights from such vast amounts of information due to their limited processing capacity. Consequently, the learning potential of smaller models may be constrained, leading to suboptimal performance even with increased training data.

Furthermore, the challenges faced by smaller models are exacerbated when dealing with large datasets of inferior quality. The overwhelming volume of data becomes difficult for smaller models to process effectively, especially when coupled with noise and irrelevant information present in the dataset. This creates a double burden for smaller models, as they not only grapple with the sheer quantity of data but also struggle to discern relevant patterns amidst the noise, hindering their learning and generalisation capabilities.

Understanding the intricate relationship between data quality and model size stands as a pivotal endeavour for advancing dialogue modelling:

Delving into Dataset Quality: Thorough investigation into the impact of dataset quality on model performance is imperative. Techniques aimed at data cleaning, preprocessing, and filtering hold the potential to significantly enhance the learning process by ensuring that models are trained on high-quality, relevant data free from noise and inconsistencies. For instance, Kowsari et al. emphasize the importance of standard data collection protocols, noting that variations in training and test sets can introduce inconsistencies that adversely affect model performance (Kowsari et al., 2019). Similarly, Kunilovskaya and Plum highlight how preprocessing impacts the effectiveness of NLP models, suggesting that appropriate text representation through preprocessing can lead to improved outcomes in various applications (Kunilovskaya & Plum, 2021).

While increasing the amount of training data generally aids in improving language model performance, the benefits are heavily dependent on the quality and relevance of that data. The section on the OpenSubtitles dataset experiments highlights an important phenomenon - simply scaling up dataset size does not guarantee commensurate performance gains, especially for smaller language models. This underscores the pivotal role that data quality plays in enabling effective learning from limited training resources (Tian et al., 2019). Research by Li et al. further supports this notion, indicating that models trained on diverse and contextually rich datasets perform better than those trained on larger but less relevant datasets (Li et al., 2022).

Data quality issues can manifest in various ways that impede model learning. Noise in the form of irrelevant content, inconsistencies, errors, or lack of context can obfuscate the true underlying patterns models aim to learn (Muthuraman et al., 2021). Furthermore, data lacking in diversity and representative coverage of the target distribution can lead to models acquiring distorted or incomplete knowledge, as noted by Srivastava et al., who emphasize the importance of data preprocessing in extracting useful patterns from web usage data (Srivastava et al., 2015). For dialogue tasks specifically, datasets with incoherent exchanges, lack of grounding information, or poor alignment between utterances and

conversational flows can severely limit a model's ability to learn effective response generation strategies (Serban et al., 2017).

The challenges posed by data quality are further compounded when models have limited capacity, as is the case with smaller language models aimed at domain-specific dialogue tasks. Their tendency to latch onto spurious patterns or be misled by noisy signals in data is exacerbated (Ferrario et al., 2020). Contending with vast quantities of low-quality data becomes computationally prohibitive for these models. As such, thoroughly understanding and mitigating data quality issues is paramount, especially when working with more modest model footprints tailored to specialised dialogue applications.

Optimising Model-Data Pairing: Achieving an optimal match between model size and the volume and quality of the dataset is crucial. While larger models might have the capacity to handle extensive datasets, smaller models could potentially derive more significant benefits from focused datasets containing relevant and contextually rich information. By aligning model size with the appropriate data volume, researchers can enhance the efficiency and effectiveness of model training processes (Rajesh & Hiwarkar, 2023).

By delving into these complexities and developing strategies to address the challenges posed by data quality and model size, researchers can refine model training methodologies and make informed decisions regarding dataset selection. This concerted effort lays the groundwork for the development of more robust and effective language models tailored to meet the demands of diverse dialogue modelling applications.

6.8 Beyond the Correlation: Exploring the Reencoder Architecture

The Reencoder architecture provides a compelling case study within the framework of the dimensionality-data-task relationship, offering valuable insights into its scalability and performance dynamics. Future investigations could delve into the effects of augmenting the dimensionality of the embedding space within the Reencoder architecture on its overall performance, particularly when trained on larger datasets. This exploration would shed light on how the architecture scales with increasing data volumes and unveil the balance between capturing intricate contextual nuances and maintaining computational efficiency, thereby advancing our understanding of its scalability and performance characteristics.

Moreover, research endeavours could delve into exploring innovative data augmentation strategies to enrich the training data utilised by the Reencoder architecture. By integrating domain-specific knowledge and context into the training process, such strategies could bolster the architecture's versatility and effectiveness across diverse dialogue modelling tasks. This entails incorporating domain-specific information, factual databases, and relevant ontologies into the training data, thereby equipping the Reencoder with a deeper understanding of domain-specific nuances and enhancing its applicability in various dialogue modelling scenarios.

The emerging relationship between model dimensionality, data requirements, and task specificity underscores a significant avenue for future research in dialogue modelling.

Leveraging smaller language models in tandem with meticulously curated datasets holds immense potential for developing dialogue systems tailored to meet the unique demands of specific domains. Addressing the associated challenges related to data curation and harnessing the capabilities of architectures like the Reencoder are pivotal steps towards advancing the dialogue modelling field, paving the way for the development of more efficient and effective conversational AI systems.

6.9 Summary

This chapter presented a comprehensive analysis of experimental findings in transformer architectures for dialogue modelling, revealing several significant insights into the interplay between model architecture, dataset characteristics, and embedding approaches. The Reencoder architecture consistently demonstrated superior performance across multiple evaluation metrics and datasets, attributed to its unique ability to iteratively refine input representations and effectively leverage contextual information from previous conversation turns. However, the absolute performance metrics across all architectures were lower than anticipated, leading to important observations about the relationship between model dimensionality, data requirements, and task specificity.

A crucial finding emerged regarding the impact of dataset quality and size on model performance. While conventional wisdom might suggest that larger datasets invariably lead to better results, our experiments revealed that carefully curated, smaller datasets often yielded superior performance compared to larger, noisier alternatives. This was particularly evident with the DailyDialog and IEMOCAP datasets, which consistently produced better results across different architectures. Furthermore, the study uncovered an interesting relationship between embedding layers and model performance, with BERT embeddings generally showing superior results, except for the notable case of the Reencoder model trained on DailyDialog, which performed optimally with SubwordTokenizer.

These findings contribute to an emerging understanding of the advantages of smaller language models in specific dialogue modelling tasks. With the largest model in our study comprising only 84 million parameters (0.4% of ChatGPT's size), the research demonstrates that efficient, task-specific models can achieve competitive performance when paired with high-quality, domain-specific data. This suggests a promising direction for practical applications where computational resources are constrained, challenging the assumption that ever-larger models are necessary for effective dialogue modelling. Future research directions should focus on optimizing the relationship between model dimensionality and data requirements, exploring innovative data augmentation strategies, and investigating the scalability of successful architectures like the Reencoder.

Chapter 7: Conclusion and Future Work

7.1 Conclusion

Dialogue modelling plays a pivotal role in natural language processing (NLP) research, enabling the development of conversational AI systems capable of engaging in human-like interactions. Central to the success of dialogue modelling endeavours is the careful selection and curation of training datasets, as well as the choice of appropriate model architectures.

Given the significance of understanding the interplay between model architecture, embedding mechanisms, and dataset characteristics in dialogue modelling tasks, the study embarked on an extensive series of experiments. Our objective was to comprehensively evaluate the performance and efficacy of various architectures and embedding techniques across datasets of varying sizes and qualities. This systematic approach allowed us to gain insights into the relative strengths and weaknesses of different models and methodologies, facilitating informed decision-making in model selection for specific dialogue modelling applications. Additionally, our experiments enabled us to explore how different datasets, characterised by their scale and quality, impact model performance and generalisation capability, thereby contributing to a deeper understanding of the factors influencing dialogue model efficacy.

This research addressed a critical limitation in current chatbot models by exploring innovative approaches to incorporate broader conversational context in dialogue modelling. The study focused on enhancing three different Transformer architectures — Encoder-Decoder Transformer, Extractor, and a novel Reencoder — by introducing modifications tailored to address their limitations in modelling dialogue, including previous turns of the conversation. The research also extended these architectures to handle multimodal inputs, incorporating both text and audio data to capture a more comprehensive range of conversational cues.

The primary novelty of this research lies in the development and implementation of the Reencoder architecture, which introduces a groundbreaking approach to dialogue modelling through its innovative reencoding mechanism. Unlike traditional transformer architectures that process conversational context in a single pass, the Reencoder's additional reencoding step enables a more sophisticated analysis of conversational dynamics. This novel approach fundamentally transforms how contextual information is processed and integrated into the dialogue generation process, leading to demonstrably superior performance across multiple standardized metrics. The architecture's ability to maintain consistent performance improvements across diverse datasets represents a significant advancement in the field's understanding of contextual processing in dialogue systems.

Furthermore, this research challenges the prevailing trend toward increasingly large language models by demonstrating the unexpected effectiveness of smaller, specialized architectures in dialogue modelling tasks. The novel finding that compact models, when paired with carefully curated datasets, can achieve comparable or superior performance to

larger models represents a paradigm shift in dialogue system development. This discovery is particularly significant as it suggests a more resource-efficient approach to building effective dialogue systems, contrasting sharply with the conventional wisdom that emphasizes the necessity of large-scale models and extensive training data.

Key findings from the study revealed that the novel Reencoder architecture, particularly when paired with BERT embeddings, consistently outperformed other models across various metrics. This architecture demonstrated a superior ability to capture and integrate contextual information, leading to more coherent and contextually appropriate responses. The research also highlighted the effectiveness of subword tokenizers, such as those used in BERT and TensorFlow, in improving dialogue modelling performance. Additionally, the incorporation of audio embeddings alongside text data proved valuable in enhancing the models' ability to capture nuances in emotion, tone, and context.
The outcomes of this research contribute significantly to the field of conversational AI, offering insights into more efficient and accessible dialogue modelling approaches. By developing architectures that can effectively leverage contextual information and handle multimodal inputs, this study addresses critical challenges in human-chatbot interaction. These findings not only advance the state-of-the-art in dialogue modelling but also hold potential for mitigating the resource-intensive nature of large language models, offering promising alternatives for businesses and developers with limited computational resources. Future research directions may include further optimization of the proposed architectures, exploration of additional multimodal inputs, and investigation of their applicability across diverse dialogue domains.

7.2 Contribution

This research study explored various transformer architectures for dialogue modelling, including a baseline architecture, an Encoder-Decoder Transformer, an Extractor model, and a novel Reencoder architecture. The Reencoder model, which incorporates an additional reencoding step, consistently outperformed other architectures across multiple datasets and evaluation metrics. This innovative design allows for better capture and integration of contextual information from previous conversation turns, leading to more coherent dialogue generation. The Reencoder's superior performance was consistent across diverse datasets such as Meld, Cornell, OpenSubtitles, and DailyDialog, showcasing lower TER scores, higher BLEU scores, and superior accuracy compared to its counterparts.
The study highlights the critical role of embedding layers in language models. These layers convert discrete tokens or words into dense, continuous vector representations, encapsulating semantic and contextual information. This process enables language models to acquire meaningful representations of words and their interrelationships, facilitating efficient computation and robust learning of intricate linguistic patterns in a high-dimensional vector space.
While the performance metrics were comparable among architectures, the absolute values were lower than expected. This was attributed to the use of smaller language models and relatively small datasets due to computational constraints. The research emphasises the multifaceted nature of dialogue modelling and the need for comprehensive investigations into model scalability, dataset size, and data quality to enhance the effectiveness of transformer architectures in capturing the nuances of human conversation.

Interestingly, the study revealed the distinct advantage of employing small language models with compact, meticulously curated datasets for specialised tasks like dialogue modelling. This approach contrasts with the conventional use of large language models requiring vast amounts of training data. Smaller models paired with tailored datasets achieved comparable, if not superior, performance while operating within resource-constrained environments. This finding suggests a more efficient and effective approach to dialogue modelling, particularly for specialised tasks. The research concludes by emphasising the importance of selecting appropriate transformer architectures and datasets for chatbot modelling tasks and suggests areas for further exploration in optimising model architectures, exploring novel training techniques, and investigating the generalizability of findings across different datasets.

This research advances the field of dialogue modelling through several significant theoretical and practical contributions that span architectural innovations, empirical findings, and methodological advancements. The work's primary contributions centre on novel architectural developments, particularly in the realm of transformer-based dialogue systems.

At the forefront of our architectural innovations is the pioneering Reencoder architecture, which represents a significant advancement in transformer-based dialogue modelling. This architecture's distinctive feature—an additional reencoding step—has demonstrated substantial improvements in dialogue generation quality across multiple standardized metrics. The architecture's consistent superior performance across diverse datasets, including Meld, Cornell, OpenSubtitles, and DailyDialog, validates its robustness and generalizability. The Reencoder architecture achieved notably lower Translation Edit Rate (TER) scores, higher BLEU scores, and superior accuracy metrics compared to baseline and contemporary architectures, establishing its effectiveness in dialogue generation tasks.

Building upon the foundational work of Riley et al. (2021), we developed an enhanced Extractor model that significantly advances the state-of-the-art in contextual awareness for dialogue systems. This implementation consistently demonstrated performance improvements over baseline architectures across all evaluation metrics, achieving non-zero scores across BLEU, METEOR, TER, and Perplexity metrics, in stark contrast to the baseline's null performance. The model's enhanced capability to incorporate and utilize contextual information from previous conversation turns represents a substantial step forward in dialogue modelling technology.

A significant contribution lies in our development of novel Audio-Transformer architectures that successfully integrate textual and audio modalities. This work addresses fundamental challenges in multimodal dialogue processing through innovative alignment techniques for audio and text modalities, the creation of enriched word representations combining standard embeddings with audio features, and the implementation of efficient fusion strategies for different embedding types. This multimodal integration framework opens new avenues for more comprehensive dialogue understanding and generation.

Our empirical findings challenge conventional wisdom regarding model scaling by demonstrating the effectiveness of smaller, specialized models. This research shows that

comparable or superior performance can be achieved with reduced computational resources, particularly in specialized dialogue modelling tasks. These findings have significant practical implications for resource-constrained applications and suggest a more efficient approach to dialogue system development.

The comprehensive investigation into embedding layers provides crucial insights for language model design, offering detailed analysis of token-to-vector transformation processes and deepening our understanding of semantic and contextual information encoding. This work documents the critical relationship between embedding quality and model performance, contributing valuable knowledge to the field of natural language processing.

Our cross-modal performance analysis provides valuable insights into the relative effectiveness of different modality combinations. Through systematic comparison of audio, text, and multimodal architectures, we have identified optimal modality combinations for specific dialogue tasks, supported by quantitative analysis of performance differences between modality types.

From a methodological perspective, this research delivers an improved TensorFlow baseline architecture that provides a robust foundation for future research comparisons and establishes new benchmarks for architecture evaluation. The development of a thorough evaluation framework, encompassing multiple standardized metrics and enabling meaningful comparison of architectural variations across diverse datasets, represents a significant methodological contribution to the field.

These contributions collectively advance our understanding of dialogue modelling systems and provide practical architectures for improved conversational AI applications. The research not only introduces novel technical solutions but also challenges existing paradigms, particularly regarding model scaling and modality integration. These findings have significant implications for both academic research and practical applications in the field of conversational AI, paving the way for more efficient and effective dialogue systems development.

7.3 Future Work

This study highlights the effectiveness of leveraging small language models (SLMs) in conjunction with meticulously curated datasets tailored for specific tasks such as dialogue modelling. The empirical findings underscore the significance of selecting a high-performing embedding layer and coupling it with an appropriate architecture to achieve optimal performance. By demonstrating the importance of this synergy, the research emphasises the critical role that model architecture and data curation play in enhancing the performance of dialogue modelling systems. Moreover, beyond the selection of existing models and datasets, this study introduces a novel architecture, the Reencoder model, which exhibits promising results in the realm of dialogue modelling. The emergence of this innovative architecture opens up new possibilities for enhancing the efficiency and effectiveness of dialogue modelling systems.

Furthermore, this research paves the way for further exploration into the intricate interplay between model size, architecture, and data curation strategies. By delving deeper into these factors, future studies can elucidate the nuanced dynamics that influence the performance of dialogue modelling systems. This entails investigating how variations in model size and architecture impact the model's ability to capture and comprehend the intricacies of natural language conversations. Additionally, exploring advanced data curation techniques and their synergistic effects with specific model architectures could offer valuable insights into optimising dialogue modelling systems for various applications and domains. Thus, this research not only contributes to the current understanding of dialogue modelling but also lays the groundwork for future investigations aimed at pushing the boundaries of conversational AI technology.

Moreover, the findings of this study underscore the importance of continued research and development efforts in the field of dialogue modelling. As the demand for sophisticated conversational AI systems continues to grow, there is a pressing need for advancements in model architecture, data curation methodologies, and evaluation metrics. By addressing these challenges, researchers can propel the field of dialogue modelling forward, enabling the creation of more robust, contextually aware, and natural-sounding conversational agents. Ultimately, this ongoing research endeavours to bridge the gap between human and machine communication, unlocking new possibilities for human-computer interaction and transforming the way users engage with AI systems in various domains.
Exploring the interplay between model size, architecture, and data curation presents a promising avenue for future research.

7.3.1 Leveraging Small Language Models and Focused Datasets

The empirical findings underscore the efficacy of employing small language models in conjunction with compact, meticulously curated datasets for specialised tasks like dialogue modelling. To further elucidate the potential of this approach, future experiments could delve into exploring the optimal size and architecture of small language models for different dialogue modelling tasks. By systematically varying the size and complexity of the models while keeping the dataset size constant, researchers can elucidate the trade-offs between model complexity, dataset curation, and performance. Recent empirical findings have shed light on the effectiveness of integrating Small Language Models (SLMs) with compact, meticulously chosen datasets for dialogue modelling. This strategic approach yields several notable advantages:

Efficiency: SLMs, characterised by their smaller size and reduced complexity, demand fewer computational resources for both training and operation compared to their larger counterparts. This efficiency renders them suitable for deployment on resource-constrained devices, thereby widening the scope of potential applications for dialogue modelling.

Focus: Through training on carefully curated datasets tailored to specific dialogue domains, SLMs can attain proficiency in understanding and responding to the intricacies within that domain. This focused training enables them to generate responses that are not only more accurate but also contextually relevant within the specified domain.
Moving forward, future research endeavours can delve deeper into two key areas:

1. Model Size and Architecture Exploration: Systematically exploring the complexity of SLMs while keeping dataset size constant allows researchers to pinpoint the optimal balance between model capability and data requirements for distinct dialogue tasks. This exploration serves as a guiding force in developing SLMs that can achieve peak performance tailored to specific dialogue applications.

2. Data Augmentation and Transfer Learning: Further investigation into techniques such as data augmentation, which involves artificially expanding the dataset, and transfer learning, which leverages knowledge from related tasks, holds immense potential in enhancing the efficacy of SLMs. These approaches offer avenues to address the limitations posed by smaller datasets and bolster the model's capacity to generalise to unseen scenarios, thereby advancing the state-of-the-art in dialogue modelling.

To accomplish this research path, researchers could design a comprehensive experimental framework that systematically explores the efficacy of Small Language Models (SLMs) in dialogue modelling. This framework would involve defining a range of SLM sizes and architectures to test, selecting a fixed, curated dataset for a specific dialogue task, and establishing clear evaluation metrics such as perplexity, BLEU score, and response relevance. The core of the investigation would focus on two key areas: model size and architecture exploration, and the application of data augmentation and transfer learning techniques.

In exploring model size and architecture, researchers would systematically vary model parameters, such as the number of layers and hidden units, while training these different configurations on the fixed dataset. This approach would allow for a detailed analysis of the trade-offs between model complexity and performance, helping to identify the optimal balance for specific dialogue tasks. Concurrently, the study would delve into data augmentation techniques, implementing methods like paraphrasing and back-translation to artificially expand the training dataset. By comparing the performance of models trained on augmented data against those trained on non-augmented data, researchers can assess the effectiveness of these techniques in enhancing SLM capabilities.

The research would also explore transfer learning approaches, involving pre-training models on large, general domain corpora before fine-tuning them on the specific dialogue task. This strategy would be compared with models trained from scratch to evaluate the benefits of transfer learning in the context of SLMs for dialogue modelling. Additionally, generalisation studies would be conducted by testing the models on out-of-domain dialogue tasks, providing insights into how well different model sizes and architectures adapt to new scenarios.

However, this research path is not without challenges. The systematic exploration of model sizes and architectures, even with SLMs, requires significant computational resources. Creating high-quality, domain-specific datasets for training is time-consuming and may necessitate expert knowledge. Researchers must navigate the complex task of balancing model size, performance, and generalisation ability, which may vary across different dialogue tasks. The risk of overfitting, particularly with smaller datasets and more complex models, presents another hurdle. Choosing appropriate evaluation metrics that accurately

reflect model performance in dialogue tasks can be challenging, as can be ensuring the reproducibility of results across different experimental runs and hardware setups.

Moreover, while SLMs offer advantages in efficiency and focus, they may struggle to generalise beyond their specific domain, potentially limiting their broader applicability. The effectiveness of transfer learning may also vary depending on the similarity between the pre-training and fine-tuning tasks. Despite these challenges, by systematically addressing these issues and thoroughly exploring the proposed research areas, researchers can gain valuable insights into optimising SLMs for dialogue modelling tasks. This research has the potential to significantly advance our understanding of efficient and effective dialogue modelling techniques, paving the way for more sophisticated and resource-conscious conversational AI systems.

7.3.2 The relationship between model dimensionality and dataset size

LMs of smaller size trained on specialised datasets provide a scalable and accessible solution to dialogue modelling, especially in environments with limited resources. With reduced computational demands and less extensive training data requirements, these models promote inclusivity in accessing advanced dialogue modelling capabilities. This inclusivity facilitates the broad adoption of dialogue systems across diverse sectors and use cases, spanning from customer service and virtual assistants to educational aids and healthcare solutions.
Expanding on our current understanding of the correlation between model dimensionality and data size opens up several promising avenues for future research:

1. Quantifying the Correlation: A crucial aspect of future research involves establishing a more precise mathematical relationship between the dimensionality of language models (LMs) and the size of training data required for optimal performance. By quantifying this correlation, researchers can develop predictive models or guidelines to determine the ideal data volume necessary for training LMs of varying sizes. This quantitative insight will be invaluable for optimising resource allocation in training endeavours, ensuring efficient use of computational resources and reducing unnecessary data collection efforts.

2. Impact on Generalizability: Investigating the impact of the dimensionality-data correlation on the generalizability of dialogue models is paramount. While smaller models have demonstrated proficiency in specific domains, assessing their ability to adapt to diverse conversational contexts is essential for real-world deployment. Future research should explore how variations in model size and training data influence the model's capacity to generalise across different dialogue tasks and domains. Understanding these dynamics will inform strategies for enhancing model adaptability and robustness in varied application scenarios.

3. Curriculum Learning Techniques: Exploring curriculum learning techniques offers a promising avenue for addressing the challenges associated with smaller datasets. Curriculum learning involves exposing the model to progressively complex training data, starting from simpler patterns and gradually introducing more intricate linguistic structures. By guiding the model's learning process in a structured

manner, curriculum learning can effectively mitigate the limitations of smaller datasets, enabling LMs to learn more sophisticated representations of language. Future research should investigate the efficacy of different curriculum strategies and their impact on model performance and generalisation capabilities in dialogue modelling tasks.

To pursue this research path, researchers could design a comprehensive study that explores the relationship between model dimensionality, dataset size, and performance in dialogue modelling tasks. The study would focus on smaller language models (LMs) trained on specialised datasets, aiming to provide scalable and accessible solutions for dialogue modelling, especially in resource-constrained environments.

The research would begin by establishing a framework to quantify the correlation between model dimensionality and the size of training data required for optimal performance. This would involve systematically varying model sizes and dataset sizes, training models on these different configurations, and measuring performance across a range of dialogue tasks. By analysing the resulting data, researchers could develop mathematical models or heuristics that predict the ideal data volume needed for LMs of different sizes. This quantitative approach would help optimise resource allocation in training efforts, potentially reducing computational costs and data collection requirements.

Next, the study would investigate the impact of the dimensionality-data correlation on the generalizability of dialogue models. This would involve training models of varying sizes on domain-specific datasets, and then testing their performance on out-of-domain tasks or in diverse conversational contexts. By examining how well these models adapt to new scenarios, researchers can gain insights into the trade-offs between model size, dataset specificity, and generalisation capabilities. This understanding would be crucial for developing strategies to enhance model adaptability and robustness across different dialogue domains and tasks.

The research would also explore the potential of curriculum learning techniques to address challenges associated with smaller datasets. This would involve designing and implementing various curriculum strategies, such as gradually increasing the complexity of training examples or introducing domain-specific knowledge in a structured manner. Researchers would compare the performance of models trained with different curriculum approaches against those trained using traditional methods, assessing their impact on model performance and generalisation capabilities in dialogue modelling tasks.

However, this research path faces several challenges. Accurately quantifying the relationship between model dimensionality and data size requires extensive experimentation, which can be computationally intensive and time-consuming. Ensuring the quality and relevance of specialised datasets for different domains is another significant challenge, as is developing meaningful evaluation metrics that capture the nuances of dialogue performance across various contexts.

Moreover, balancing the trade-offs between model size, performance, and generalizability is a complex task that may vary across different dialogue domains and applications. The effectiveness of curriculum learning techniques may also depend on the specific characteristics of the dialogue task and domain, requiring careful design and adaptation.

Despite these challenges, this research path holds significant promise for advancing our understanding of efficient and effective dialogue modelling. By quantifying the relationship between model size and data requirements, investigating generalizability, and exploring innovative training techniques like curriculum learning, researchers can develop more accessible and adaptable dialogue systems. This could lead to broader adoption of advanced dialogue modelling capabilities across diverse sectors, from customer service and virtual assistants to educational aids and healthcare solutions, ultimately making conversational AI more inclusive and impactful.

7.3.3 Generalizability and Transferability of SLMs

Future research endeavours in machine learning could concentrate on delving into the generalizability and transferability of small language models (SLMs) trained on meticulously curated datasets across a spectrum of dialogue modelling tasks and domains.

1. Benchmarking Across Diverse Scenarios: A crucial aspect of future work involves conducting extensive experiments across a wide array of dialogue modelling benchmarks and real-world applications. This comprehensive assessment is indispensable for evaluating the robustness and adaptability of SLMs trained on curated datasets. By benchmarking performance in diverse scenarios encompassing various dialogue tasks and domains, researchers can gain valuable insights into the generalizability of the approach. This holistic evaluation aids in identifying potential strengths and weaknesses, thereby guiding the refinement of SLMs for improved performance across different contexts.

2. Fine-tuning for Seamless Integration: Another promising avenue for future exploration is the investigation of techniques for fine-tuning pre-trained SLMs on domain-specific dialogue data. Fine-tuning offers a strategic approach to leverage the general knowledge encoded within pre-trained models while tailoring them to specific domains or applications. By fine-tuning SLMs on domain-specific dialogue datasets, researchers can enhance the models' performance and adaptability to specific contexts. This fine-tuning process facilitates the seamless integration of SLMs into various applications across diverse domains, ensuring that the models can effectively address the nuanced requirements of different dialogue tasks and domains.

7.3.4 The Reencoder Architecture: Scalability and Performance

Future work on expanding the Reencoder Architecture could focus on scalability and performance enhancements, particularly in two key areas

1. Dimensionality of Embedding Space: A fruitful avenue for exploration involves systematically increasing the dimensionality of the embedding space while simultaneously training the model on progressively larger datasets. By systematically varying the dimensionality of the embedding space, researchers can assess how the richness and complexity of the embedding representations impact the model's ability to capture nuanced contextual information in dialogue interactions. Additionally, exploring the effect of dimensionality on model

performance across different dialogue modelling tasks and datasets can provide insights into the optimal embedding dimensionality for maximising the efficacy of the Reencoder architecture.

2. Performance Across Tasks and Datasets: Another promising direction for future research involves evaluating the effect of dimensionality on model performance across diverse dialogue modelling tasks and datasets. This comprehensive analysis can offer valuable insights into the optimal embedding space size for maximising the Reencoder architecture's efficacy across different domains and applications. By systematically assessing performance across various tasks and datasets, researchers can identify the embedding space size that strikes the optimal balance between capturing contextual nuances and computational efficiency, thus informing the design of more effective dialogue modelling systems.

3. Training Process Optimization: Exploring advanced training techniques such as curriculum learning and transfer learning can address challenges associated with training the Reencoder architecture on larger datasets. Curriculum learning involves gradually increasing the difficulty of the training examples presented to the model, enabling more efficient learning and better generalisation to complex dialogue scenarios. Similarly, transfer learning leverages knowledge from pre-trained models to bootstrap the training process, facilitating faster convergence and improved performance, particularly in scenarios with limited labelled data. By integrating these optimization techniques into the training process, researchers can enhance the efficiency, robustness, and scalability of the Reencoder architecture for dialogue modelling tasks

Another pivotal focus area for future research is enhancing the performance of the Reencoder Architecture through iterative refinement and optimization. This involves fine-tuning various architectural parameters, optimising training processes, and leveraging advanced techniques such as transfer learning and curriculum learning. By systematically optimising the Reencoder Architecture in these key areas, researchers can elevate its performance metrics, including accuracy, fluency, and responsiveness. Additionally, enhancing the architecture's adaptability to diverse linguistic styles, dialects, and conversational nuances can further bolster its utility across a broad spectrum of dialogue modelling tasks and applications. In summary, future work on the Reencoder Architecture should prioritise investigating the impact of embedding space dimensionality on scalability and performance across a range of dialogue modelling tasks and datasets. By systematically exploring these factors, researchers can gain a deeper understanding of how to optimise the Reencoder architecture for capturing nuanced contextual information while maintaining computational efficiency, thereby advancing the state-of-the-art in dialogue modelling.

References

AbuShawar, B., & Atwell, E. (2015). ALICE Chatbot: Trials and Outputs. *Computación y Sistemas*, *19*(4), Article 4. https://doi.org/10.13053/cys-19-4-2326

Adiwardana, D., Luong, M.-T., So, D. R., Hall, J., Fiedel, N., Thoppilan, R., Yang, Z., Kulshreshtha, A., Nemade, G., Lu, Y., & Le, Q. V. (2020). Towards a Human-like Open-Domain Chatbot. *arXiv:2001.09977 [Cs, Stat]*. http://arxiv.org/abs/2001.09977

Agarwal, A., & Lavie, A. (2008). Meteor, M-BLEU and M-TER: Evaluation Metrics for High-Correlation with Human Rankings of Machine Translation Output. In C. Callison-Burch, P. Koehn, C. Monz, J. Schroeder, & C. S. Fordyce (Eds.), *Proceedings of the Third Workshop on Statistical Machine Translation* (pp. 115–118). Association for Computational Linguistics. https://aclanthology.org/W08-0312

Alamri, H., Cartillier, V., Das, A., Wang, J., Cherian, A., Essa, I., Batra, D., Marks, T. K., Hori, C., Anderson, P., Lee, S., & Parikh, D. (2019). *Audio Visual Scene-Aware Dialog*. 7558–7567. https://openaccess.thecvf.com/content_CVPR_2019/html/Alamri_Audio_Visual_Scene-Aware_Dialog_CVPR_2019_paper.html

Althnian, A., AlSaeed, D., Al-Baity, H., Samha, A., Dris, A. B., Alzakari, N., Abou Elwafa, A., & Kurdi, H. (2021). Impact of Dataset Size on Classification Performance: An Empirical Evaluation in the Medical Domain. *Applied Sciences*, *11*(2), Article 2. https://doi.org/10.3390/app11020796

Arevalo, J., Solorio, T., Montes-y-Gómez, M., & González, F. A. (2020). Gated multimodal networks. *Neural Computing and Applications*, *32*(14), 10209–10228. https://doi.org/10.1007/s00521-019-04559-1

Ayanouz, S., Abdelhakim, B. A., & Benhmed, M. (2020). A Smart Chatbot Architecture based NLP and Machine Learning for Health Care Assistance. *Proceedings of the 3rd International Conference on Networking, Information Systems & Security*, 1–6.

https://doi.org/10.1145/3386723.3387897

Banerjee, S., & Lavie, A. (2005). *METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments*. 8.

Bathija, R., Agarwal, P., Somanna, R., & Pallavi, G. B. (2020). Guided Interactive Learning through Chatbot using Bi-directional Encoder Representations from Transformers (BERT). *2020 2nd International Conference on Innovative Mechanisms for Industry Applications (ICIMIA)*, 82–87. https://doi.org/10.1109/ICIMIA48430.2020.9074905

Bird, J. J., Ekárt, A., & Faria, D. R. (2021). Chatbot Interaction with Artificial Intelligence: Human data augmentation with T5 and language transformer ensemble for text classification. *Journal of Ambient Intelligence and Humanized Computing*. https://doi.org/10.1007/s12652-021-03439-8

Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2017). Enriching Word Vectors with Subword Information. *Transactions of the Association for Computational Linguistics*, *5*, 135–146. https://doi.org/10.1162/tacl_a_00051

Bostrom, K., & Durrett, G. (2020). Byte Pair Encoding is Suboptimal for Language Model Pretraining. In T. Cohn, Y. He, & Y. Liu (Eds.), *Findings of the Association for Computational Linguistics: EMNLP 2020* (pp. 4617–4624). Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.findings-emnlp.414

Busso, C., Bulut, M., Lee, C.-C., Kazemzadeh, A., Mower, E., Kim, S., Chang, J. N., Lee, S., & Narayanan, S. S. (2008). IEMOCAP: Interactive emotional dyadic motion capture database. *Language Resources and Evaluation*, *42*(4), 335–359. https://doi.org/10.1007/s10579-008-9076-6

Cahn, J. (2017). *CHATBOT: Architecture, Design, & Development*. University of Pennsylvania.

Caldarini, G., Jaf, S., & McGarry, K. (2022). A Literature Survey of Recent Advances in Chatbots. *Information*, *13*(1), Article 1. https://doi.org/10.3390/info13010041

Can ChatGPT understand context? (n.d.). ResearchGate. Retrieved November 10, 2023,

from https://www.researchgate.net/post/Can_ChatGPT_understand_context

Chada, R. (2020). Simultaneous paraphrasing and translation by fine-tuning Transformer models. *Proceedings of the Fourth Workshop on Neural Generation and Translation*, 198–203. https://doi.org/10.18653/v1/2020.ngt-1.23

Chen, F., Meng, F., Chen, X., Li, P., & Zhou, J. (2021). *Multimodal Incremental Transformer with Visual Grounding for Visual Dialogue Generation* (arXiv:2109.08478). arXiv. https://doi.org/10.48550/arXiv.2109.08478

Chen, T., Kornblith, S., Norouzi, M., & Hinton, G. (2020). *A Simple Framework for Contrastive Learning of Visual Representations* (arXiv:2002.05709). arXiv. http://arxiv.org/abs/2002.05709

Christensen, S., Johnsrud, S., Ruocco, M., & Ramampiaro, H. (2018). Context-Aware Sequence-to-Sequence Models for Conversational Systems. *arXiv:1805.08455 [Cs]*. http://arxiv.org/abs/1805.08455

Chung, W., Cahyawijaya, S., Wilie, B., Lovenia, H., & Fung, P. (2023). InstructTODS: Large Language Models for End-to-End Task-Oriented Dialogue Systems. In K. Chen & L.-W. Ku (Eds.), *Proceedings of the Second Workshop on Natural Language Interfaces* (pp. 1–21). Association for Computational Linguistics. https://doi.org/10.18653/v1/2023.nlint-1.1

Colby, K. M., Hilf, F. D., Weber, S., & Kraemer, H. C. (1972). Turing-like indistinguishability tests for the validation of a computer simulation of paranoid processes. *Artificial Intelligence*, *3*, 199–221. https://doi.org/10.1016/0004-3702(72)90049-5

Cornell Movie-Dialogs Corpus. (2023, November 21). https://www.cs.cornell.edu/~cristian/Cornell_Movie-Dialogs_Corpus.html

Costa, P. (2018). Conversing with personal digital assistants: On gender and artificial intelligence. *Journal of Science and Technology of the Arts*, 59-72 Páginas. https://doi.org/10.7559/CITARJ.V10I3.563

Decoder models—Hugging Face NLP Course. (2023, November 9).

https://huggingface.co/learn/nlp-course/chapter1/6

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In J. Burstein, C. Doran, & T. Solorio (Eds.), *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)* (pp. 4171–4186). Association for Computational Linguistics. https://doi.org/10.18653/v1/N19-1423

Dhyani, M., & Kumar, R. (2020). An intelligent Chatbot using deep learning with Bidirectional RNN and attention model. *Materials Today: Proceedings*. https://doi.org/10.1016/j.matpr.2020.05.450

Du, L. (2024). Query-Based Dialogue Summarization Using BART. *Applied and Computational Engineering*, *29*, 160–166. https://doi.org/10.54254/2755-2721/29/20231149

Fernandes, A. (2018, November 9). *NLP, NLU, NLG and how Chatbots work*. Medium. https://chatbotslife.com/nlp-nlu-nlg-and-how-chatbots-work-dd7861dfc9df

Ferrario, A., Demiray, B., Yordanova, K., Luo, M., & Martin, M. (2020). Social Reminiscence in Older Adults' Everyday Conversations: Automated Detection Using Natural Language Processing and Machine Learning. *Journal of Medical Internet Research*, *22*(9), e19133. https://doi.org/10.2196/19133

Gangi, M. A. D., Negri, M., & Turchi, M. (2019). Adapting Transformer to End-to-End Spoken Language Translation. *Interspeech 2019*, 1133–1137. https://doi.org/10.21437/Interspeech.2019-3045

Ghandeharioun, A., Shen, J. H., Jaques, N., Ferguson, C., Jones, N., Lapedriza, A., & Picard, R. (2019). Approximating Interactive Human Evaluation with Self-Play for Open-Domain Dialog Systems. *arXiv:1906.09308 [Cs, Stat]*. http://arxiv.org/abs/1906.09308

Go, E., & Sundar, S. S. (2019). Humanizing chatbots: The effects of visual, identity and conversational cues on humanness perceptions. *Computers in Human Behavior*,

97, 304–316. https://doi.org/10.1016/j.chb.2019.01.020

Gong, Y., Khurana, S., Karlinsky, L., & Glass, J. (2023). Whisper-AT: Noise-Robust Automatic Speech Recognizers are Also Strong General Audio Event Taggers. *INTERSPEECH 2023*, 2798–2802. https://doi.org/10.21437/Interspeech.2023-2193

Google Colab. (n.d.). Retrieved June 28, 2024, from https://colab.research.google.com/github/tensorflow/examples/blob/demo/community/en/transformer_chatbot.ipynb#scrollTo=_B147qKb_0ks

GPT-3.5-turbo how to remember previous messages like Chat-GPT website—API. (2023, April 20). OpenAI Developer Forum. https://community.openai.com/t/gpt-3-5-turbo-how-to-remember-previous-messages-like-chat-gpt-website/170370

Gu, X., Chen, G., Wang, Y., Zhang, L., Luo, T., & Wen, L. (2023). Text with Knowledge Graph Augmented Transformer for Video Captioning. *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 18941–18951. https://doi.org/10.1109/CVPR52729.2023.01816

He, T., Liu, J., Cho, K., Ott, M., Liu, B., Glass, J., & Peng, F. (2021). Analyzing the Forgetting Problem in the Pretrain-Finetuning of Dialogue Response Models. *arXiv:1910.07117 [Cs]*. http://arxiv.org/abs/1910.07117

Hinton, G., Deng, L., Yu, D., Dahl, G., Mohamed, A., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T., & Kingsbury, B. (2012). Deep Neural Networks for Acoustic Modeling in Speech Recognition: The Shared Views of Four Research Groups. *IEEE Signal Processing Magazine*, *29*(6), 82–97. https://doi.org/10.1109/MSP.2012.2205597

Hinton, G., Vinyals, O., & Dean, J. (2015). *Distilling the Knowledge in a Neural Network* (arXiv:1503.02531). arXiv. http://arxiv.org/abs/1503.02531

Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, *9*(8), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

How does ChatGPT handle multi-turn conversations? (n.d.). Quora. Retrieved November

10, 2023, from https://chatgptmakemoney1.quora.com/How-does-ChatGPT-handle-multi-turn-conversations

Hu, T., Xu, A., Liu, Z., You, Q., Guo, Y., Sinha, V., Luo, J., & Akkiraju, R. (2018). *Touch Your Heart: A Tone-aware Chatbot for Customer Care on Social Media* (arXiv:1803.02952). arXiv. http://arxiv.org/abs/1803.02952

Introducing ChatGPT. (n.d.). Retrieved November 10, 2023, from https://openai.com/blog/chatgpt

Jia, J. (2003). *The Study of the Application of a Keywords-based Chatbot System on the Teaching of Foreign Languages*. 11.

Jurafsky, D., & Martin, J. (2008). *Speech And Language Processing: An Introduction to Natural Language Processing , Computational Linguistics, and Speech Recognition* (2nd edition). Pearson.

Jurafsky, D., & Martin, J. (2020). *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition* (Vol. 2).

Kaplan, J., McCandlish, S., Henighan, T., Brown, T. B., Chess, B., Child, R., Gray, S., Radford, A., Wu, J., & Amodei, D. (2020). *Scaling Laws for Neural Language Models* (arXiv:2001.08361). arXiv. http://arxiv.org/abs/2001.08361

Klein, G., Kim, Y., Deng, Y., Senellart, J., & Rush, A. (2017). OpenNMT: Open-Source Toolkit for Neural Machine Translation. *Proceedings of ACL 2017, System Demonstrations*, 67–72. https://doi.org/10.18653/v1/P17-4012

Kowsari, K., Jafari Meimandi, K., Heidarysafa, M., Mendu, S., Barnes, L., & Brown, D. (2019). Text Classification Algorithms: A Survey. *Information*, *10*(4), Article 4. https://doi.org/10.3390/info10040150

Kruger, J.-L., & Steyn, F. (2013). Subtitles and Eye Tracking: Reading and Performance. *Reading Research Quarterly*, *49*. https://doi.org/10.1002/rrq.59

Kumar, R., & Ali, M. M. (2020). A Review on Chatbot Design and Implementation Techniques. *International Journal of Engineering and Technology. 7. 2791*,

07(02), Article 02.

Kunilovskaya, M., & Plum, A. (2021). *Text Preprocessing and its Implications in a Digital Humanities Project*. 85–93. https://doi.org/10.26615/issn.2603-2821.2021_013

Lavie, A., & Agarwal, A. (2007). METEOR: An Automatic Metric for MT Evaluation with High Levels of Correlation with Human Judgments. In C. Callison-Burch, P. Koehn, C. S. Fordyce, & C. Monz (Eds.), *Proceedings of the Second Workshop on Statistical Machine Translation* (pp. 228–231). Association for Computational Linguistics. https://aclanthology.org/W07-0734

Le, H., Chen, N., & Hoi, S. (2022). Multimodal Dialogue State Tracking. In M. Carpuat, M.-C. de Marneffe, & I. V. Meza Ruiz (Eds.), *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (pp. 3394–3415). Association for Computational Linguistics. https://doi.org/10.18653/v1/2022.naacl-main.248

Lee, C.-C., Mower Provost, E., Busso, C., Lee, S., & Narayanan, S. (2011). Emotion recognition using a hierarchical binary decision tree approach. *Speech Communication*, *53*, 1162–1171.

Lee, J., & Lee, W. (2022). *CoMPM: Context Modeling with Speaker's Pre-trained Memory Tracking for Emotion Recognition in Conversation* (arXiv:2108.11626). arXiv. https://doi.org/10.48550/arXiv.2108.11626

Lemon, O., & Gruenstein, A. (2004). Multithreaded context for robust conversational interfaces: Context-sensitive speech recognition and interpretation of corrective fragments. *ACM Transactions on Computer-Human Interaction (TOCHI)*, *11*, 241–267.

Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., & Zettlemoyer, L. (2020). BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 7871–7880. https://doi.org/10.18653/v1/2020.acl-main.703

Li, Y., Su, H., Shen, X., Li, W., Cao, Z., & Niu, S. (2017). DailyDialog: A Manually Labelled Multi-turn Dialogue Dataset. In G. Kondrak & T. Watanabe (Eds.), *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)* (pp. 986–995). Asian Federation of Natural Language Processing. https://aclanthology.org/I17-1099

Li, Y., Sun, B., Feng, S., & Li, K. (2022). *Stop Filtering: Multi-View Attribute-Enhanced Dialogue Learning* (arXiv:2205.11206). arXiv. https://doi.org/10.48550/arXiv.2205.11206

Linzen, T. (2020). How Can We Accelerate Progress Towards Human-like Linguistic Generalization? In D. Jurafsky, J. Chai, N. Schluter, & J. Tetreault (Eds.), *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics* (pp. 5210–5217). Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.acl-main.465

Lison, P., & Tiedemann, J. (2016). OpenSubtitles2016: Extracting Large Parallel Corpora from Movie and TV Subtitles. *Proceedings of the 10th International Conference on Language Resources and Evaluation*, 7.

Liu, G., Wang, S., Yu, J., & Yin, J. (2022). *A Survey on Multimodal Dialogue Systems: Recent Advances and New Frontiers*. 845–853. https://doi.org/10.1109/AEMCSE55572.2022.00170

Liu, H., Dacon, J., Fan, W., Liu, H., Liu, Z., & Tang, J. (2020). Does Gender Matter? Towards Fairness in Dialogue Systems. In D. Scott, N. Bel, & C. Zong (Eds.), *Proceedings of the 28th International Conference on Computational Linguistics* (pp. 4403–4416). International Committee on Computational Linguistics. https://doi.org/10.18653/v1/2020.coling-main.390

Liu, Y., Chen, K., Liu, C., Qin, Z., Luo, Z., & Wang, J. (2019). Structured Knowledge Distillation for Semantic Segmentation. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2599–2608. https://doi.org/10.1109/CVPR.2019.00271

Lu, Z., & Li, H. (2013). *A Deep Architecture for Matching Short Texts*. 9.

Lukovnikov, D., Fischer, A., & Lehmann, J. (2019). Pretrained Transformers for Simple Question Answering over Knowledge Graphs. In C. Ghidini, O. Hartig, M. Maleshkova, V. Svátek, I. Cruz, A. Hogan, J. Song, M. Lefrançois, & F. Gandon (Eds.), *The Semantic Web – ISWC 2019* (pp. 470–486). Springer International Publishing. https://doi.org/10.1007/978-3-030-30793-6_27

Luo, X., Tong, S., Fang, Z., & Qu, Z. (2019). Frontiers: Machines vs. Humans: The Impact of Artificial Intelligence Chatbot Disclosure on Customer Purchases. *Marketing Science*, mksc.2019.1192. https://doi.org/10.1287/mksc.2019.1192

Mesnil, G., Dauphin, Y., Yao, K., Bengio, Y., Deng, L., Hakkani-Tur, D., He, X., Heck, L., Tur, G., Yu, D., & Zweig, G. (2015). Using Recurrent Neural Networks for Slot Filling in Spoken Language Understanding. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, *23*(3), Article 3. https://doi.org/10.1109/TASLP.2014.2383614

Metallinou, A., Wöllmer, M., Katsamanis, A., Eyben, F., Schuller, B., & Narayanan, S. (2012). Context-Sensitive Learning for Enhanced Audiovisual Emotion Classification. *IEEE Transactions on Affective Computing*. https://doi.org/10.1109/T-AFFC.2011.40

Meyer, J. G., Urbanowicz, R. J., Martin, P. C. N., O'Connor, K., Li, R., Peng, P.-C., Bright, T. J., Tatonetti, N., Won, K. J., Gonzalez-Hernandez, G., & Moore, J. H. (2023). ChatGPT and large language models in academia: Opportunities and challenges. *BioData Mining*, *16*(1), 20. https://doi.org/10.1186/s13040-023-00339-9

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed Representations of Words and Phrases and their Compositionality. *Advances in Neural Information Processing Systems*, *26*. https://papers.nips.cc/paper_files/paper/2013/hash/9aa42b31882ec039965f3c4923ce901b-Abstract.html

Minixhofer, B., Pfeiffer, J., & Vulić, I. (2023). CompoundPiece: Evaluating and Improving

Decompounding Performance of Language Models. In H. Bouamor, J. Pino, & K. Bali (Eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing* (pp. 343–359). Association for Computational Linguistics. https://doi.org/10.18653/v1/2023.emnlp-main.24

Muthuraman, K., Reiss, F., Xu, H., Cutler, B., & Eichenberger, Z. (2021). Data Cleaning Tools for Token Classification Tasks. In E. Dragut, Y. Li, L. Popa, & S. Vucetic (Eds.), *Proceedings of the Second Workshop on Data Science with Human in the Loop: Language Advances* (pp. 59–61). Association for Computational Linguistics. https://doi.org/10.18653/v1/2021.dash-1.10

Nakano, M., Hasegawa, Y., Funakoshi, K., Takeuchi, J., Torii, T., Nakadai, K., Kanda, N., Komatani, K., Okuno, H. G., & Tsujino, H. (2011). A multi-expert model for dialogue and behavior control of conversational robots and agents. *Knowledge-Based Systems*, *24*(2), 248–256. https://doi.org/10.1016/j.knosys.2010.08.004

Naseem, U., Razzak, I., Musial, K., & Imran, M. (2020). Transformer based Deep Intelligent Contextual Embedding for Twitter sentiment analysis. *Future Generation Computer Systems*, *113*, 58–69. https://doi.org/10.1016/j.future.2020.06.050

Neumann, M., & Vu, N. T. (2019). Improving Speech Emotion Recognition with Unsupervised Representation Learning on Unlabeled Speech. *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 7390–7394. https://doi.org/10.1109/ICASSP.2019.8682541

Okuda, T., & Shoda, S. (2018). AI-based Chatbot Service for Financial Industry. *FUJITSU Sci. Tech. J.*, *54*(2), Article 2.

OpenSubtitles. (2021, February 3). http://opus.nlpl.eu/OpenSubtitles-v2018.php

Pan, L., Saxon, M., Xu, W., Nathani, D., Wang, X., & Wang, W. Y. (2023). *Automatically Correcting Large Language Models: Surveying the landscape of diverse self-correction strategies* (arXiv:2308.03188). arXiv. http://arxiv.org/abs/2308.03188

Pandeya, Y. R., Bhattarai, B., & Lee, J. (2021). Music video emotion classification using

slow–fast audio–video network and unsupervised feature representation. *Scientific Reports*, *11*(1), 19834. https://doi.org/10.1038/s41598-021-98856-2

Pennington, J., Socher, R., & Manning, C. (2014). GloVe: Global Vectors for Word Representation. In A. Moschitti, B. Pang, & W. Daelemans (Eds.), *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (pp. 1532–1543). Association for Computational Linguistics. https://doi.org/10.3115/v1/D14-1162

Peters, B., & Martins, A. F. T. (2022). Beyond Characters: Subword-level Morpheme Segmentation. In G. Nicolai & E. Chodroff (Eds.), *Proceedings of the 19th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology* (pp. 131–138). Association for Computational Linguistics. https://doi.org/10.18653/v1/2022.sigmorphon-1.14

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., & Zettlemoyer, L. (2018). Deep Contextualized Word Representations. In M. Walker, H. Ji, & A. Stent (Eds.), *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)* (pp. 2227–2237). Association for Computational Linguistics. https://doi.org/10.18653/v1/N18-1202

Pölz, A., Blaschke, A. P., Komma, J., Farnleitner, A. H., & Derx, J. (2024). Transformer Versus LSTM: A Comparison of Deep Learning Models for Karst Spring Discharge Forecasting. *Water Resources Research*, *60*(4), e2022WR032602. https://doi.org/10.1029/2022WR032602

Poria, S., Hazarika, D., Majumder, N., Naik, G., Cambria, E., & Mihalcea, R. (2019). MELD: A Multimodal Multi-Party Dataset for Emotion Recognition in Conversations. In A. Korhonen, D. Traum, & L. Màrquez (Eds.), *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics* (pp. 527–536). Association for Computational Linguistics. https://doi.org/10.18653/v1/P19-1050

Qi, J., Lyu, B., Wu, X., & Marfurt, K. (2020). Comparing convolutional neural networking and image processing seismic fault detection methods. *SEG Technical Program Expanded Abstracts 2020*, 1111–1115. https://doi.org/10.1190/segam2020-3428171.1

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). *Language Models are Unsupervised Multitask Learners*. 24.

Radziwill, N., & Benton, M. (2019). *Evaluating Quality of Chatbots and Intelligent Conversational Agents*. 21.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., & Liu, P. J. (2020). Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *Journal of Machine Learning Research*, *21*(140), 1–67.

Rajesh, M. A., & Hiwarkar, D. T. (2023). Exploring Preprocessing Techniques for Natural LanguageText: A Comprehensive Study Using Python Code. *International Journal of Engineering Technology and Management Sciences*, *7*(5), 390–399. https://doi.org/10.46647/ijetms.2023.v07i05.047

Rastogi, A., Zang, X., Sunkara, S., Gupta, R., & Khaitan, P. (2020). Towards Scalable Multi-Domain Conversational Agents: The Schema-Guided Dialogue Dataset. *Proceedings of the AAAI Conference on Artificial Intelligence*, *34*(05), Article 05. https://doi.org/10.1609/aaai.v34i05.6394

Riley, P., Constant, N., Guo, M., Kumar, G., Uthus, D., & Parekh, Z. (2021). TextSETTR: Few-Shot Text Style Extraction and Tunable Targeted Restyling. In C. Zong, F. Xia, W. Li, & R. Navigli (Eds.), *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)* (pp. 3786–3800). Association for Computational Linguistics. https://doi.org/10.18653/v1/2021.acl-long.293

Roller, S., Dinan, E., Goyal, N., Ju, D., Williamson, M., Liu, Y., Xu, J., Ott, M., Smith, E. M., Boureau, Y.-L., & Weston, J. (2021). Recipes for Building an Open-Domain

Chatbot. *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, 300–325. https://doi.org/10.18653/v1/2021.eacl-main.24

Rose, L. T., & Fischer, K. W. (2011). Garbage In, Garbage Out: Having Useful Data Is Everything. *Measurement: Interdisciplinary Research and Perspectives*, *9*(4), 222–226. https://doi.org/10.1080/15366367.2011.632338

Saikh, T., Naskar, S. K., Ekbal, A., & Bandyopadhyay, S. (2018). Textual Entailment Using Machine Translation Evaluation Metrics. In A. Gelbukh (Ed.), *Computational Linguistics and Intelligent Text Processing* (Vol. 10761, pp. 317–328). Springer International Publishing. https://doi.org/10.1007/978-3-319-77113-7_25

Sarch, G., Wu, Y., Tarr, M., & Fragkiadaki, K. (2023). Open-Ended Instructable Embodied Agents with Memory-Augmented Large Language Models. In H. Bouamor, J. Pino, & K. Bali (Eds.), *Findings of the Association for Computational Linguistics: EMNLP 2023* (pp. 3468–3500). Association for Computational Linguistics. https://doi.org/10.18653/v1/2023.findings-emnlp.226

Schmidhuber, J. (1992). Learning to Control Fast-Weight Memories: An Alternative to Dynamic Recurrent Networks. *Neural Computation*, *4*(1), Article 1. https://doi.org/10.1162/neco.1992.4.1.131

Sennrich, R., Haddow, B., & Birch, A. (2016). Neural Machine Translation of Rare Words with Subword Units. In K. Erk & N. A. Smith (Eds.), *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (pp. 1715–1725). Association for Computational Linguistics. https://doi.org/10.18653/v1/P16-1162

Serban, I., Klinger, T., Tesauro, G., Talamadupula, K., Zhou, B., Bengio, Y., & Courville, A. (2017). Multiresolution Recurrent Neural Networks: An Application to Dialogue Response Generation. *Proceedings of the AAAI Conference on Artificial Intelligence*, *31*(1), Article 1. https://doi.org/10.1609/aaai.v31i1.10984

Shagass, C., Roemer, R. A., & Amadeo, M. (1976). Eye-Tracking Performance and

Engagement of Attention. *Archives of General Psychiatry*, *33*(1), Article 1. https://doi.org/10.1001/archpsyc.1976.01770010077015

Shang, L., Lu, Z., & Li, H. (2015). Neural Responding Machine for Short-Text Conversation. In C. Zong & M. Strube (Eds.), *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)* (pp. 1577–1586). Association for Computational Linguistics. https://doi.org/10.3115/v1/P15-1152

Shangipour ataei, T., Javdan, S., & Minaei-Bidgoli, B. (2020). Applying Transformers and Aspect-based Sentiment Analysis approaches on Sarcasm Detection. *Proceedings of the Second Workshop on Figurative Language Processing*, 67–71. https://doi.org/10.18653/v1/2020.figlang-1.9

Shao, T., Guo, Y., Chen, H., & Hao, Z. (2019). Transformer-Based Neural Network for Answer Selection in Question Answering. *IEEE Access*, *7*, 26146–26156. IEEE Access. https://doi.org/10.1109/ACCESS.2019.2900753

Shum, H., He, X., & Li, D. (2018). From Eliza to XiaoIce: Challenges and opportunities with social chatbots. *Frontiers of Information Technology & Electronic Engineering*, *19*(1), Article 1. https://doi.org/10.1631/FITEE.1700826

Snover, M., Dorr, B., Schwartz, R., Micciulla, L., & Makhoul, J. (2006). A Study of Translation Edit Rate with Targeted Human Annotation. *Proceedings of Association for Machine Translation in the Americas*, 9. https://www.cs.umd.edu/~snover/pub/amta06/ter_amta.pdf

Sojasingarayar, A. (2020). *Seq2Seq AI Chatbot with Attention Mechanism*. https://www.academia.edu/43262982/Seq2Seq_AI_Chatbot_with_Attention_Mechanism

Sordoni, A., Galley, M., Auli, M., Brockett, C., Ji, Y., Mitchell, M., Nie, J.-Y., Gao, J., & Dolan, B. (2015a). A Neural Network Approach to Context-Sensitive Generation of Conversational Responses. In R. Mihalcea, J. Chai, & A. Sarkar (Eds.),

Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (pp. 196–205). Association for Computational Linguistics. https://doi.org/10.3115/v1/N15-1020

Sordoni, A., Galley, M., Auli, M., Brockett, C., Ji, Y., Mitchell, M., Nie, J.-Y., Gao, J., & Dolan, B. (2015b). A Neural Network Approach to Context-Sensitive Generation of Conversational Responses. *arXiv:1506.06714 [Cs]*. http://arxiv.org/abs/1506.06714

Srivastava, M., Garg, R., & Mishra, P. K. (2015). Analysis of Data Extraction and Data Cleaning in Web Usage Mining. *Proceedings of the 2015 International Conference on Advanced Research in Computer Science Engineering & Technology (ICARCSET 2015)*, 1–6. https://doi.org/10.1145/2743065.2743078

Strigér, A. (2017). *End-to-End Trainable Chatbot for Restaurant Recommendations*. https://www.diva-portal.org/smash/get/diva2:1139496/FULLTEXT01.pdf

Sun, C., Baradel, F., Murphy, K., & Schmid, C. (2019). *Learning Video Representations using Contrastive Bidirectional Transformer* (arXiv:1906.05743; Issue arXiv:1906.05743). arXiv. http://arxiv.org/abs/1906.05743

Sun, Q., Wang, Y., Xu, C., Zheng, K., Yang, Y., Hu, H., Xu, F., Zhang, J., Geng, X., & Jiang, D. (2022). Multimodal Dialogue Response Generation. In S. Muresan, P. Nakov, & A. Villavicencio (Eds.), *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (pp. 2854–2866). Association for Computational Linguistics. https://doi.org/10.18653/v1/2022.acl-long.204

Sundar, A., & Heck, L. (2022). Multimodal Conversational AI: A Survey of Datasets and Approaches. In B. Liu, A. Papangelis, S. Ultes, A. Rastogi, Y.-N. Chen, G. Spithourakis, E. Nouri, & W. Shi (Eds.), *Proceedings of the 4th Workshop on NLP for Conversational AI* (pp. 131–147). Association for Computational Linguistics. https://doi.org/10.18653/v1/2022.nlp4convai-1.12

Sutskever, I., Vinyals, O., & Le, Q. V. (2014). *Sequence to Sequence Learning with Neural Networks*. 9.

Tenney, I., Das, D., & Pavlick, E. (2019). BERT Rediscovers the Classical NLP Pipeline. In A. Korhonen, D. Traum, & L. Màrquez (Eds.), *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics* (pp. 4593–4601). Association for Computational Linguistics. https://doi.org/10.18653/v1/P19-1452

Tfds.deprecated.text.Tokenizer | TensorFlow Datasets. (n.d.). TensorFlow. Retrieved March 9, 2024, from https://www.tensorflow.org/datasets/api_docs/python/tfds/deprecated/text/Tokenizer

Thißen, M., & Hergenröther, E. (2023). *Why Existing Multimodal Crowd Counting Datasets Can Lead to Unfulfilled Expectations in Real-World Applications*. 28–35. https://doi.org/10.24132/CSRN.3301.5

Tian, Y., Jia, Y., Li, L., Huang, Z., & Wang, W. (2019). Research on Modeling and Analysis of Generative Conversational System Based on Optimal Joint Structural and Linguistic Model. *Sensors*, *19*(7), Article 7. https://doi.org/10.3390/s19071675

Tjahyana, L. J. (2024). Exploring AI Chatbot Development for Gen-Z: A Study on First-Time and Experienced Voters in Pemilu. *Jurnal Ilmu Komunikasi Dan Bisnis*, *9*. https://doi.org/10.36914/jikb.v9i2.1091

Tripathi, S., Tripathi, S., & Beigi, H. (2019). *Multi-Modal Emotion recognition on IEMOCAP Dataset using Deep Learning* (arXiv:1804.05788). arXiv. http://arxiv.org/abs/1804.05788

Tsai, Y.-H. H., Bai, S., Liang, P. P., Kolter, J. Z., Morency, L.-P., & Salakhutdinov, R. (2019). Multimodal Transformer for Unaligned Multimodal Language Sequences. In A. Korhonen, D. Traum, & L. Màrquez (Eds.), *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics* (pp. 6558–6569). Association for Computational Linguistics. https://doi.org/10.18653/v1/P19-1656

Turian, J., Ratinov, L.-A., & Bengio, Y. (2010). Word Representations: A Simple and

General Method for Semi-Supervised Learning. In J. Hajič, S. Carberry, S. Clark, & J. Nivre (Eds.), *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics* (pp. 384–394). Association for Computational Linguistics. https://aclanthology.org/P10-1040

Turing, A. M. (1950). I.—COMPUTING MACHINERY AND INTELLIGENCE. *Mind*, *LIX*(236), Article 236. https://doi.org/10.1093/mind/LIX.236.433

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention Is All You Need. *arXiv:1706.03762 [Cs]*. http://arxiv.org/abs/1706.03762

Vinyals, O., & Le, Q. (2015). A Neural Conversational Model. *arXiv:1506.05869 [Cs]*. http://arxiv.org/abs/1506.05869

Vogel, C., Koutsombogera, M., & Reverdy, J. (2023). Aspects of Dynamics in Dialogue Collaboration. *Electronics*, *12*(10), Article 10. https://doi.org/10.3390/electronics12102210

Waligora, P., Aslam, H., Zeeshan, O., Belharbi, S., Koerich, A. L., Pedersoli, M., Bacon, S., & Granger, E. (2024). *Joint Multimodal Transformer for Emotion Recognition in the Wild* (arXiv:2403.10488; Version 1). arXiv. https://doi.org/10.48550/arXiv.2403.10488

Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., & Bowman, S. (2018). GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding. In T. Linzen, G. Chrupała, & A. Alishahi (Eds.), *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP* (pp. 353–355). Association for Computational Linguistics. https://doi.org/10.18653/v1/W18-5446

Wang, T., Roberts, A., Hesslow, D., Scao, T. L., Chung, H. W., Beltagy, I., Launay, J., & Raffel, C. (2022). *What Language Model Architecture and Pretraining Objective Work Best for Zero-Shot Generalization?* (arXiv:2204.05832; Issue arXiv:2204.05832). arXiv. http://arxiv.org/abs/2204.05832

Wassan, J. T., Ghuriani, V., Wassan, J. T., & Ghuriani, V. (2023). Perspective Chapter: Recent Trends in Deep Learning for Conversational AI. In *Deep Learning—Recent Findings and Research*. IntechOpen. https://doi.org/10.5772/intechopen.113250

Weidinger, L., Uesato, J., Rauh, M., Griffin, C., Huang, P.-S., Mellor, J., Glaese, A., Cheng, M., Balle, B., Kasirzadeh, A., Biles, C., Brown, S., Kenton, Z., Hawkins, W., Stepleton, T., Birhane, A., Hendricks, L. A., Rimell, L., Isaac, W., … Gabriel, I. (2022). Taxonomy of Risks posed by Language Models. *2022 ACM Conference on Fairness, Accountability, and Transparency*, 214–229. https://doi.org/10.1145/3531146.3533088

Weizenbaum, J. (1966). *ELIZA--A Computer Program For the Study of Natural Language Communication Between Man and Machine*. 6.

Wilcox, B. (2014). *Winning the Loebner's*. https://www.gamasutra.com/blogs/BruceWilcox/20141020/228091/Winning_the_Loebners.php

Wolf, L., Kotar, K., Tuckute, G., Hosseini, E., I. Regev, T., Gotlieb Wilcox, E., & Warstadt, A. S. (2023). WhisBERT: Multimodal Text-Audio Language Modeling on 100M Words. In A. Warstadt, A. Mueller, L. Choshen, E. Wilcox, C. Zhuang, J. Ciro, R. Mosquera, B. Paranjabe, A. Williams, T. Linzen, & R. Cotterell (Eds.), *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning* (pp. 253–258). Association for Computational Linguistics. https://doi.org/10.18653/v1/2023.conll-babylm.21

Xu, A., Liu, Z., Guo, Y., Sinha, V., & Akkiraju, R. (2017). A New Chatbot for Customer Service on Social Media. *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, 3506–3510. https://doi.org/10.1145/3025453.3025496

Xu, K., Lei, J., Kiros, R., Cho, K., Courville, A., Salakhutdinov, R., Zemel, R. S., & Bengio, Y. (2015). Show, Attend and Tell: Neural Image CaptionGeneration with Visual

Attention. *Proceedings of the 32 Nd International Conference on Machine Learning*, *37*, 10.

Xue, J., Wang, Y.-C., Wei, C., Liu, X., Woo, J., & Kuo, C.-C. J. (2023). *Bias and Fairness in Chatbots: An Overview* (arXiv:2309.08836; Version 2). arXiv. https://doi.org/10.48550/arXiv.2309.08836

Xue, L., Barua, A., Constant, N., Al-Rfou, R., Narang, S., Kale, M., Roberts, A., & Raffel, C. (2022). ByT5: Towards a Token-Free Future with Pre-trained Byte-to-Byte Models. *Transactions of the Association for Computational Linguistics*, *10*, 291–306. https://doi.org/10.1162/tacl_a_00461

Yan, R., Song, Y., & Wu, H. (2016). Learning to Respond with Deep Neural Networks for Retrieval-Based Human-Computer Conversation System. *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval - SIGIR '16*, 55–64. https://doi.org/10.1145/2911451.2911542

Young, S., Gašić, M., Thomson, B., & Williams, J. D. (2013). POMDP-Based Statistical Spoken Dialog Systems: A Review. *Proceedings of the IEEE*, *101*(5), Article 5. Proceedings of the IEEE. https://doi.org/10.1109/JPROC.2012.2225812

Young, T., Pandelea, V., Poria, S., & Cambria, E. (2020). Dialogue systems with audio context. *Neurocomputing*, *388*, 102–109. https://doi.org/10.1016/j.neucom.2019.12.126

Yu, S., Chen, Y., & Zaidi, H. (2020). *A Financial Service Chatbot based on Deep Bidirectional Transformers* (arXiv:2003.04987; Issue arXiv:2003.04987). arXiv. http://arxiv.org/abs/2003.04987

Zadeh, A., Liang, P. P., Poria, S., Vij, P., Cambria, E., & Morency, L.-P. (2018). Multi-attention recurrent network for human communication comprehension. *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence*, 5642–5649.

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

Zhang, B., Yang, X., Wang, G., Wang, Y., & Sun, R. (2023). M2ER: Multimodal Emotion Recognition Based on Multi-Party Dialogue Scenarios. *Applied Sciences*, *13*(20), Article 20. https://doi.org/10.3390/app132011340

Zhang, C., Yang, Z., He, X., & Deng, L. (2020). Multimodal Intelligence: Representation Learning, Information Fusion, and Applications. *IEEE Journal of Selected Topics in Signal Processing*, *14*(3), 478–493. https://doi.org/10.1109/JSTSP.2020.2987728

Zhang, M., You, D., & Wang, S. (2024). Novel framework for dialogue summarization based on factual-statement fusion and dialogue segmentation. *PLOS ONE*, *19*(4), e0302104. https://doi.org/10.1371/journal.pone.0302104

Zhang, X., Sun, W., Chen, K., & Jiang, R. (2024). A multimodal expert system for the intelligent monitoring and maintenance of transformers enhanced by multimodal language large model fine-tuning and digital twins. *IET Collaborative Intelligent Manufacturing*, *6*(4), e70007. https://doi.org/10.1049/cim2.70007

Zhang, Z., Li, J., Zhu, P., Zhao, H., & Liu, G. (2018). Modeling Multi-turn Conversation with Deep Utterance Aggregation. In E. M. Bender, L. Derczynski, & P. Isabelle (Eds.), *Proceedings of the 27th International Conference on Computational Linguistics* (pp. 3740–3752). Association for Computational Linguistics. https://aclanthology.org/C18-1317

Zhao, W., Peyrard, M., Liu, F., Gao, Y., Meyer, C. M., & Eger, S. (2019). MoverScore: Text Generation Evaluating with Contextualized Embeddings and Earth Mover Distance. In K. Inui, J. Jiang, V. Ng, & X. Wan (Eds.), *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)* (pp. 563–578). Association for Computational Linguistics. https://doi.org/10.18653/v1/D19-1053

Zhao, Y., Cheng, B., Huang, Y., & Wan, Z. (2023). Beyond Words: An Intelligent Human-Machine Dialogue System with Multimodal Generation and Emotional Comprehension. *International Journal of Intelligent Systems*, *2023*(1), 9267487. https://doi.org/10.1155/2023/9267487

Zhong, P., Wang, D., & Miao, C. (2019). An Affect-Rich Neural Conversational Model with Biased Attention and Weighted Cross-Entropy Loss. *Proceedings of the AAAI Conference on Artificial Intelligence*, *33*(01), Article 01. https://doi.org/10.1609/aaai.v33i01.33017492

Zhou, J., Deng, J., Mi, F., Li, Y., Wang, Y., Huang, M., Jiang, X., Liu, Q., & Meng, H. (2022). *Towards Identifying Social Bias in Dialog Systems: Frame, Datasets, and Benchmarks* (arXiv:2202.08011). arXiv. https://doi.org/10.48550/arXiv.2202.08011

Appendix A

***Table a.1**. System evaluation on DailyDialog dataset using different embedding algorithms and performance measures.*

Architectures/ Embeddings	**Extractor**			**Reencoder**			**Encoder-decoder**			**baseline**		
	BERT	Matrix3	GloVe	BERT	Matrix3	GloVe	BERT	Matrix3	GloVe	BERT	Matrix3	GloVe
BLEU	0.078	0.004	0.009	0.085	**0.102**	0.064	0.090	0.000	0.009	0.000	0.081	0.000
METEOR	0.157	0.055	0.087	0.159	**0.169**	0.112	0.163	0.008	0.086	0.000	0.156	0.00
TER	112.981	215.704	114.826	110.165	110.717	114.935	110.400	**98.823**	115.351	0.000	112.726	0.000
Accuracy	0.376	0.351	0.141	**0.379**	0.336	0.258	0.377	0.285	0.146	0.000	0.322	0.000

***Table a.2**. System evaluation on Cornell dataset using different embedding algorithms and performance measures.*

Architectures/ Embeddings	**Extractor**			**Reencoder**			**Encoder-decoder**			**baseline**		
	BERT	Matrix3	GloVe	BERT	Matrix3	GloVe	BERT	Matrix3	GloVe	BERT	Matrix3	GloVe
BLEU	0.006	0.006	0.004	**0.007**	0.003	0.003	0.005	0.002	0.003	0.000	**0.007**	0.000
METEOR	0.079	0.081	0.037	0.070	0.055	0.034	0.079	0.037	0.038	0.000	**0.087**	0.00
TER	117.654	127.885	113.259	117.483	249.194	**111.533**	124.280	246.923	126.591	0.000	119.127	0.000
Accuracy	**0.221**	0.120	0.067	0.224	0.166	0.132	0.233	0.123	0.089	0.000	0.151	0.000

***Table a.3**. System evaluation on OpenSubtitles dataset using different embedding algorithms and performance measures.*

Architectures/ Embeddings	**Extractor**			**Reencoder**			**Encoder-decoder**			**baseline**		
	BERT	Matrix3	GloVe	BERT	Matrix3	GloVe	BERT	Matrix3	GloVe	BERT	Matrix3	GloVe
BLEU	0.000	0.000	0.003	0.003	0.000	0.000	0.002	0.000	0.002	0.000	**0.012**	0.000
METEOR	0.055	0.064	0.029	0.061	0.041	0.025	0.056	0.062	0.025	0.000	**0.119**	0.000
TER	130.076	133.399	123.923	132.709	126.866	119.783	128.429	259.307	125.359	0.000	**118.553**	0.000
Accuracy	**0.185**	0.116	0.058	0.184	0.148	0.057	0.182	0.112	0.071	0.000	0.116	0.000

***Table a.4**. System evaluation on MELD dataset using different embedding algorithms and performance measures on text embeddings only.*

Architectures/ Embeddings	**Extractor**			**Reencoder**			**Encoder-decoder**			**baseline**		
	BERT	Matrix3	GloVe	BERT	Matrix3	GloVe	BERT	Matrix3	GloVe	BERT	Matrix3	GloVe
BLEU	0.000	0.000	0.000	0.000	0.000	0.000	**0.009**	0.000	0.000	0.000	0.000	0.000
METEOR	0.080	0.086	0.032	0.071	0.087	0.038	**0.092**	0.079	0.024	0.000	0.000	0.000
TER	118.499	114.286	121.994	**110.380**	122.097	131.963	117.163	112.025	114.491	0.000	0.000	0.000
Accuracy	0.318	0.275	0.218	0.320	0.294	0.223	**0.322**	0.294	0.218	0.000	0.000	0.000

***Table a.5**. System evaluation on OpenSubtitles dataset with Training data corresponding to 1% of the entire dataset using different embedding algorithms and performance measures on text embeddings only.*

Architectures/ Embeddings	**Extractor**			**Reencoder**			**Encoder-decoder**			**baseline**		
	BERT	Matrix3	GloVe	BERT	Matrix3	GloVe	BERT	Matrix3	GloVe	BERT	Matrix3	GloVe
BLEU	0.005	**0.008**	0.004	0.004	0.000	0.005	0.004	0.005	0.004	0.000	0.004	0.000
METEOR	0.088	0.105	0.044	0.098	**0.108**	0.057	0.092	0.107	0.043	0.000	0.052	0.000
TER	123.537	122.231	123.102	121.130	**116.494**	119.234	119.567	121.373	120.438	0.000	132.758	0.000
Accuracy	0.125	0.097	0.055	**0.134**	0.115	0.055	0.127	0.099	0.055	0.000	0.103	0.000

***Table a.6**. System evaluation on MELD dataset using audio embedding algorithms and different performance measures on audio embeddings only.*

Architectures/ Embeddings	**Extractor**	**Reencoder**	**Encoder-decoder**	**baseline**
	Audio Embeddings	Audio Embeddings	Audio Embeddings	Audio Embeddings
BLEU	0.000	0.000	0.000	0.000
METEOR	**0.031**	0.030	0.030	0.000
TER	149.445	**143.323**	159.462	0.000
Accuracy	0.035	**0.036**	0.035	0.000

***Table a.7**. System evaluation on MELD dataset using different embedding algorithms and performance measures on text and audio embeddings.*

Architectures/ Embeddings	**Extractor**			**Reencoder**			**Encoder-decoder**			**baseline**		
	BERT	Matrix3	GloVe	BERT	Matrix3	GloVe	BERT	Matrix3	GloVe	BERT	Matrix3	GloVe
BLEU	0.000	0.002	0.000	**0.004**	0.002	0.000	0.000	0.000	0.000	0.000	0.000	0.000
METEOR	0.064	0.069	0.034	**0.077**	0.06	0.023	0.053	0.065	0.002	0.000	0.000	0.000
TER	125.578	236.913	105.103	**106.212**	311.49	105.172	106.688	314.73	106.654	0.000	0.000	0.000
Perplexity	3.263	3.364	**2.353**	3.292	2.899	2.358	2.867	3.421	2.413	0.000	0.000	0.000
Accuracy	**0.091**	0.073	0.017	0.055	0.048	0.017	0.033	0.068	0.017	0.000	0.000	0.000

***Table a.8**. System evaluation on IEMOCAP dataset using different embedding algorithms and performance measures on text and audio embeddings.*

Architectures/ Embeddings	**Extractor**			**Reencoder**			**Encoder-decoder**			**baseline**		
	BERT	Matrix3	GloVe	BERT	Matrix3	GloVe	BERT	Matrix3	GloVe	BERT	Matrix3	GloVe
BLEU	0	0	0	0.004	**0.006**	0	0.004	0.006	0	0	0	0
METEOR	0.042	0.042	0.042	0.080	0.095	0.049	0.078	**0.097**	0.059	0	0	0
TER	**121.009**	253.850	253.85	166.326	259.786	174.557	135.952	245.007	271.030	0	0	0
Perplexity	4.137	4.478	3.006	3.674	3.944	**2.548**	3.766	3.923	2.614	0	0	0
Accuracy	0.025	0.045	0.017	0.059	**0.073**	0.021	0.057	0.073	0.021	0	0	0

***Table a.9**. System evaluation on IEMOCAP dataset using audio embedding algorithms and different performance measures on audio embeddings only.*

Architectures/ Embeddings	**Extractor**	**Reencoder**	**Encoder-decoder**	**baseline**
	Audio Embeddings	Audio Embeddings	Audio Embeddings	Audio Embeddings
BLEU	0.000	0.000	0.000	0.000
METEOR	0.028	**0.035**	0.032	0.000
TER	144.272	**120.764**	127.656	0.000
Accuracy	**0.027**	0.022	0.024	0.000

***Table a.10**. System evaluation on IEMOCAP dataset using different embedding algorithms and performance measures on text embeddings.*

Architectures/ Embeddings	**Extractor**			**Reencoder**			**Encoder-decoder**			**baseline**		
	BERT	Matrix 3	GloVe	BERT	Matrix 3	GloVe	BERT	Matrix 3	GloVe	BERT	Matrix 3	GloVe
BLEU	**0.207**	0.037	0.111	0.200	0.060	0.124	0.178	0.070	0.122	0	0	0
METEOR	**0.395**	0.187	0.316	0.387	0.190	0.325	0.366	0.245	0.327	0	0	0
TER	**94.619**	157.459	125.631	95.811	124.275	109.161	99.2	130.84	119.001	0	0	0
Perplexity	3.216	4.716	2.773	3.58	3.35	3.053	**2.728**	4.687	2.813	0	0	0
Accuracy	0.24	0.228	0.144	**0.240**	0.228	0.155	0.224	0.220	0.157	0	0	0

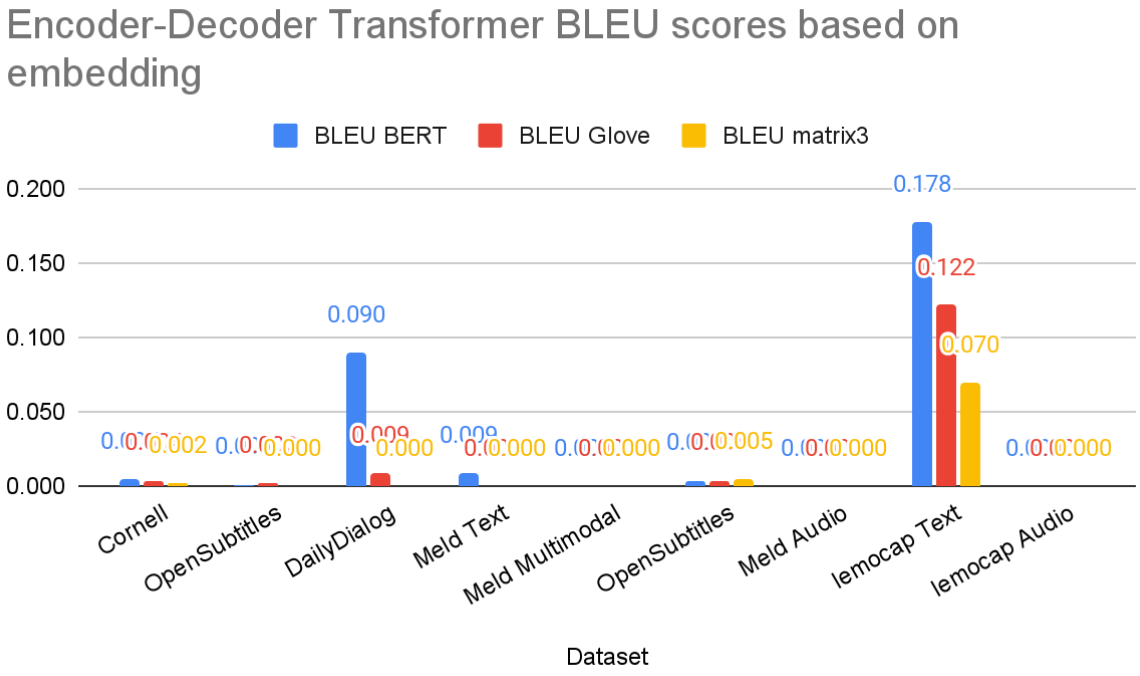

Figure a.1 BLEU scores for the Encoder-Decoder Transformer architecture trained on the different datasets, based on the different embedding methods.

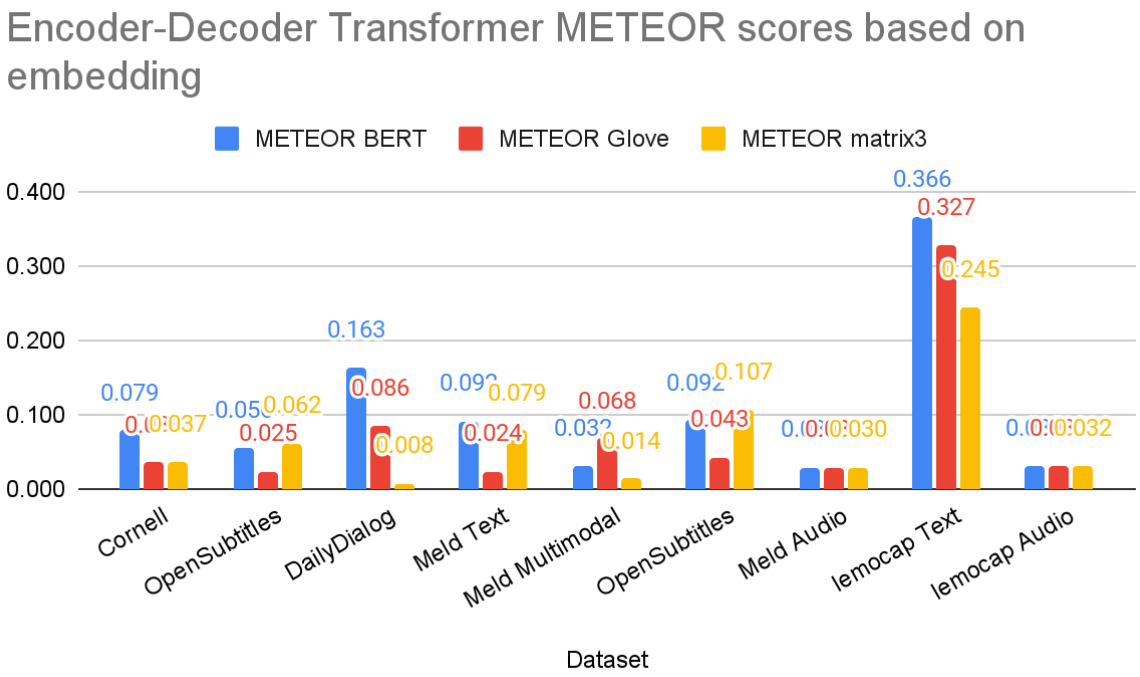

Figure a.2 METEOR scores for the Encoder-Decoder Transformer architecture trained on the different datasets, based on the different embedding methods.

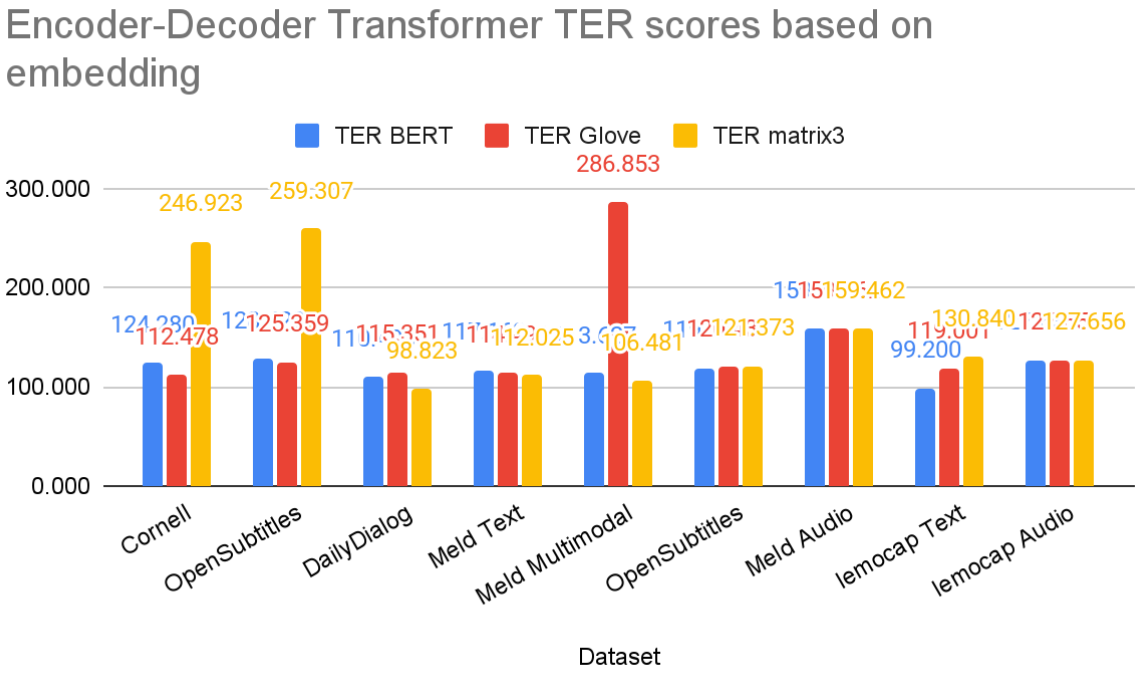

Figure a.3 TER scores for the Encoder-Decoder Transformer architecture trained on the different datasets, based on the different embedding methods.

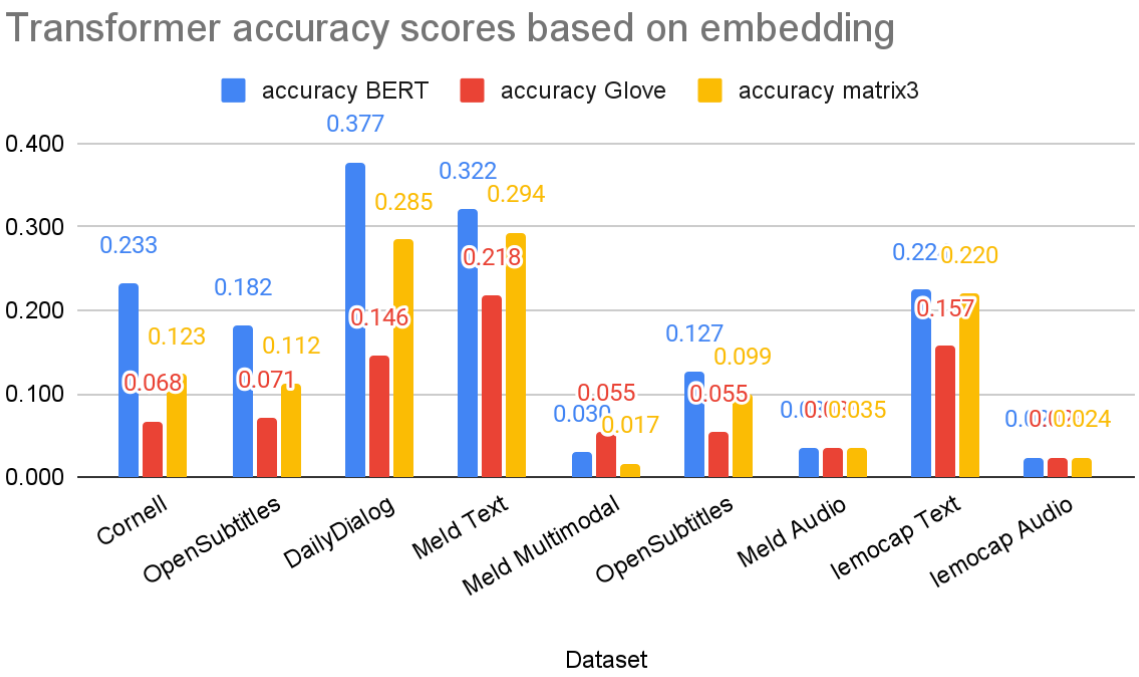

Figure a.4 Accuracy scores for the Encoder-Decoder Transformer architecture trained on the different datasets, based on the different embedding methods.

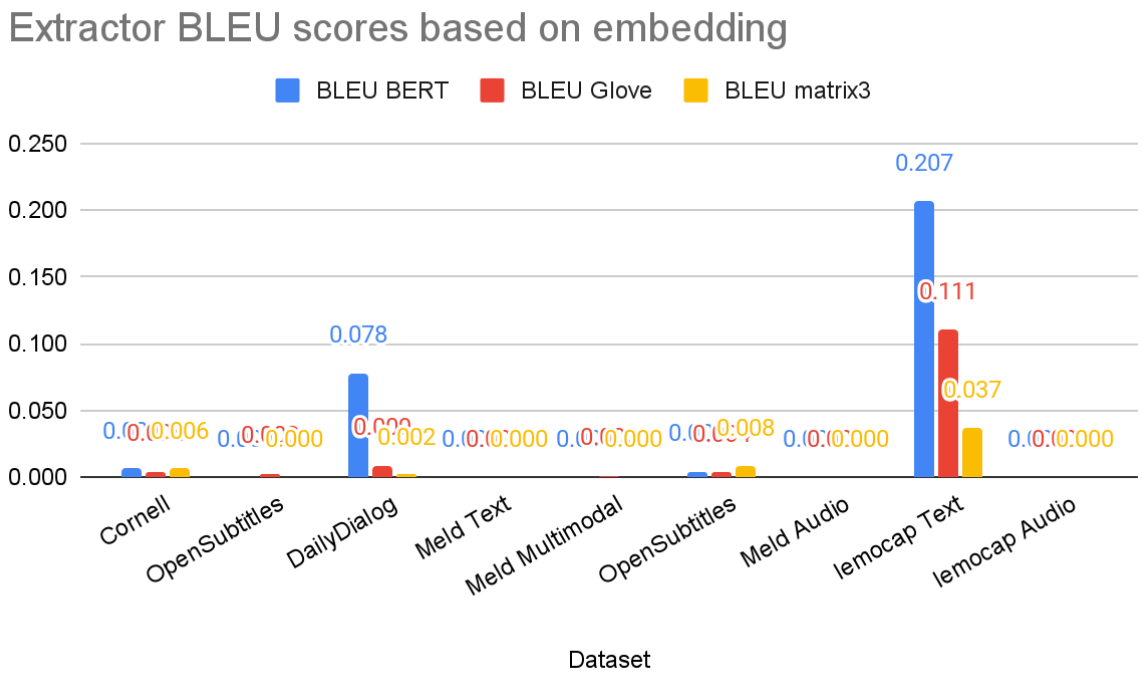

Figure a.5 BLEU scores for the Extractor architecture trained on the different datasets, based on the different embedding methods.

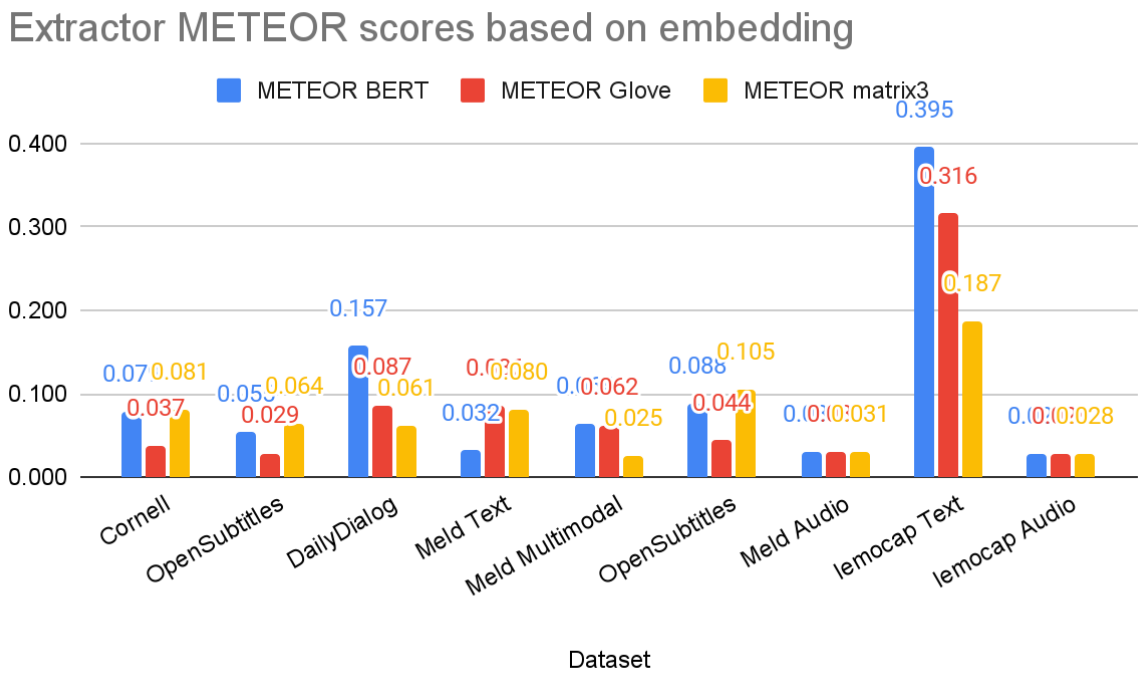

Figure a.6 METEOR scores for the Extractor architecture trained on the different datasets, based on the different embedding methods.

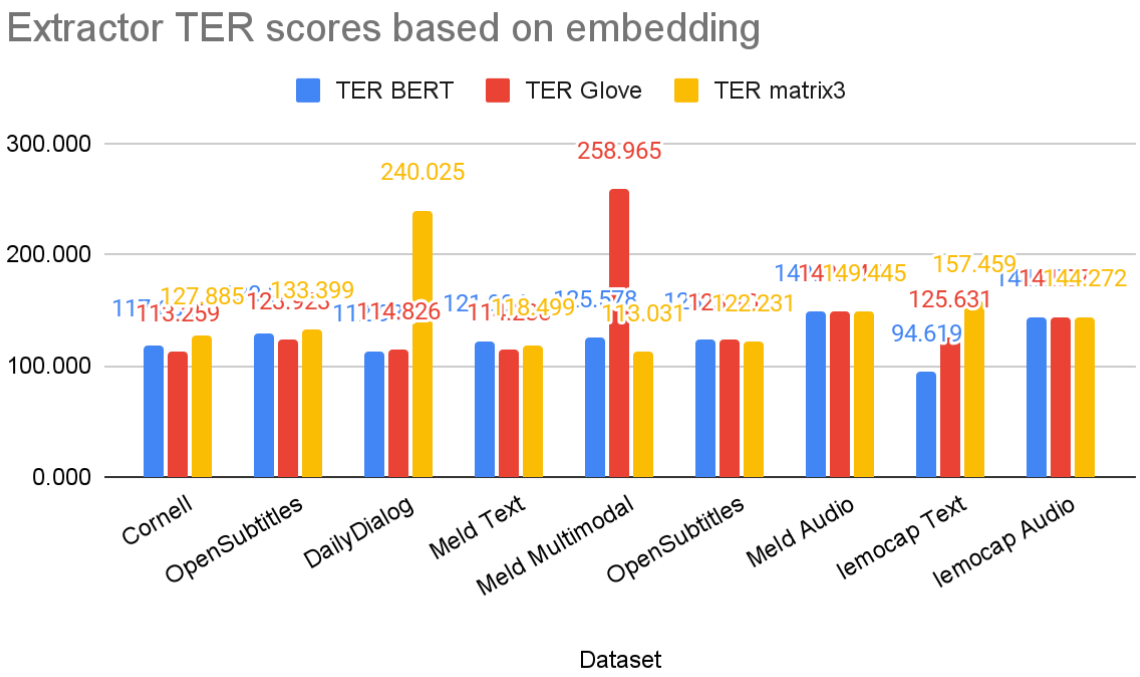

Figure a.7 TER scores for the Extractor architecture trained on the different datasets, based on the different embedding methods.

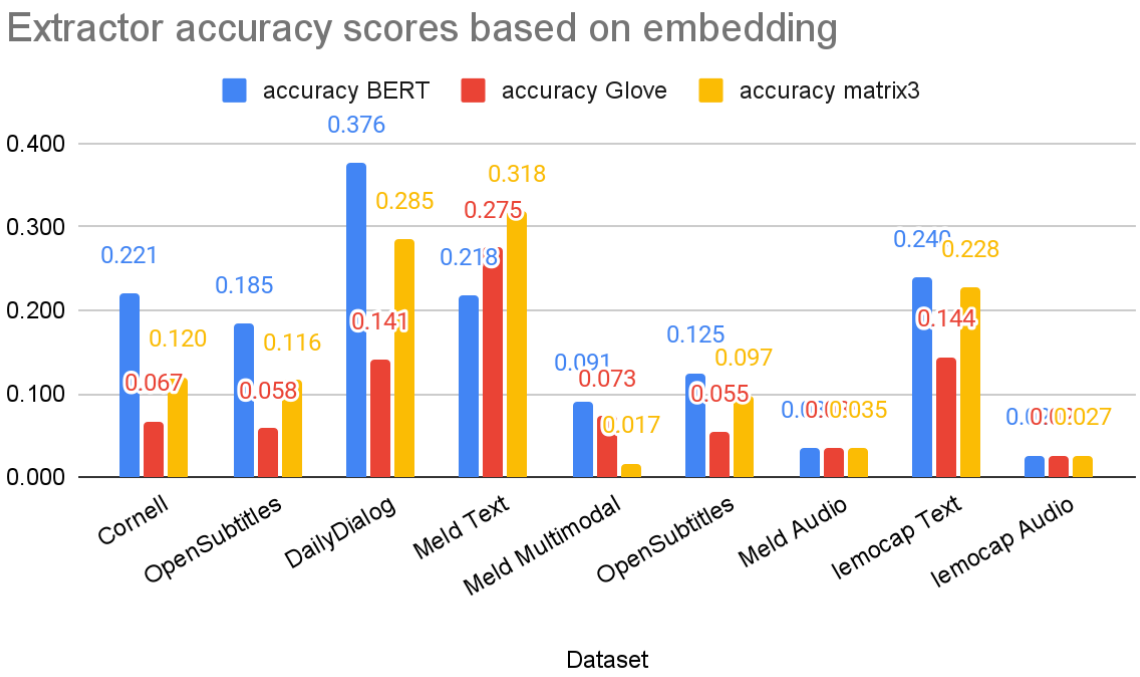

Figure a.8 Accuracy scores for the Extractor architecture trained on the different datasets, based on the different embedding methods.

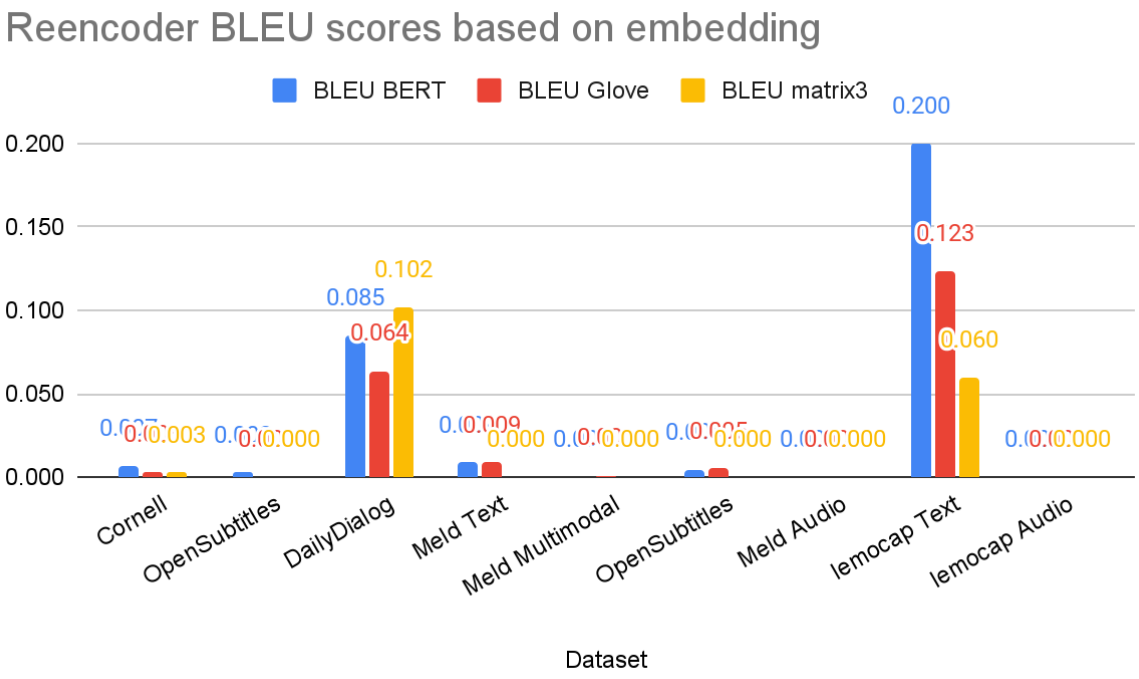

Figure a.9 BLEU scores for the Reencoder architecture trained on the different datasets, based on the different embedding methods.

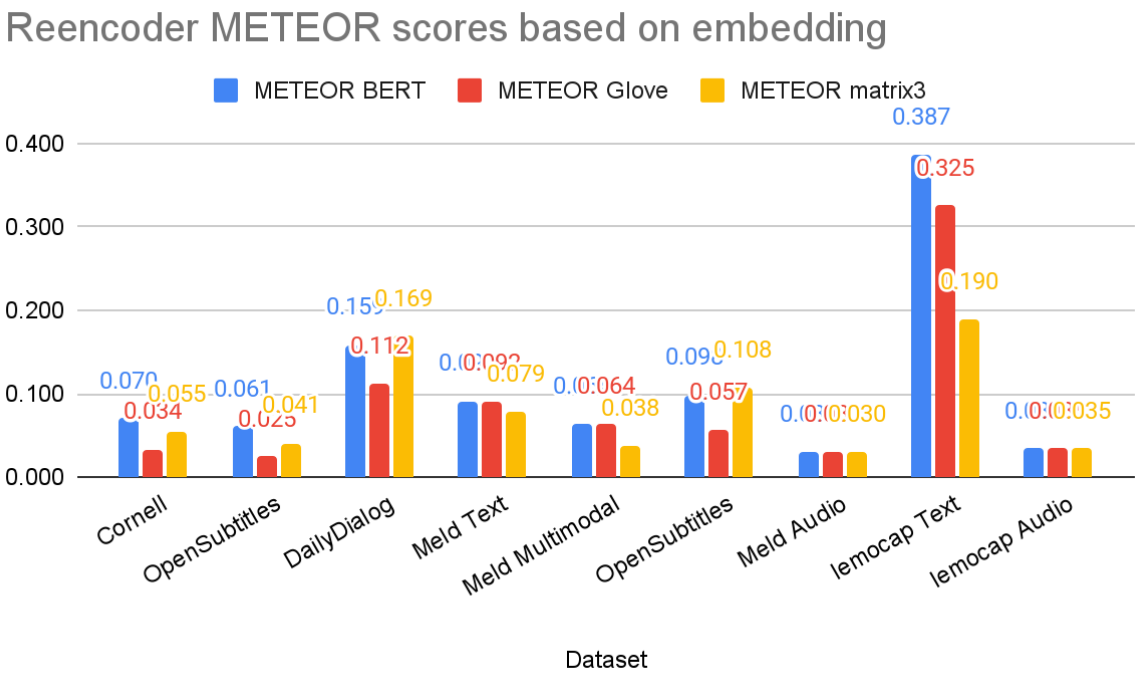

Figure a.10 METEOR scores for the Reencoder architecture trained on the different datasets, based on the different embedding methods.

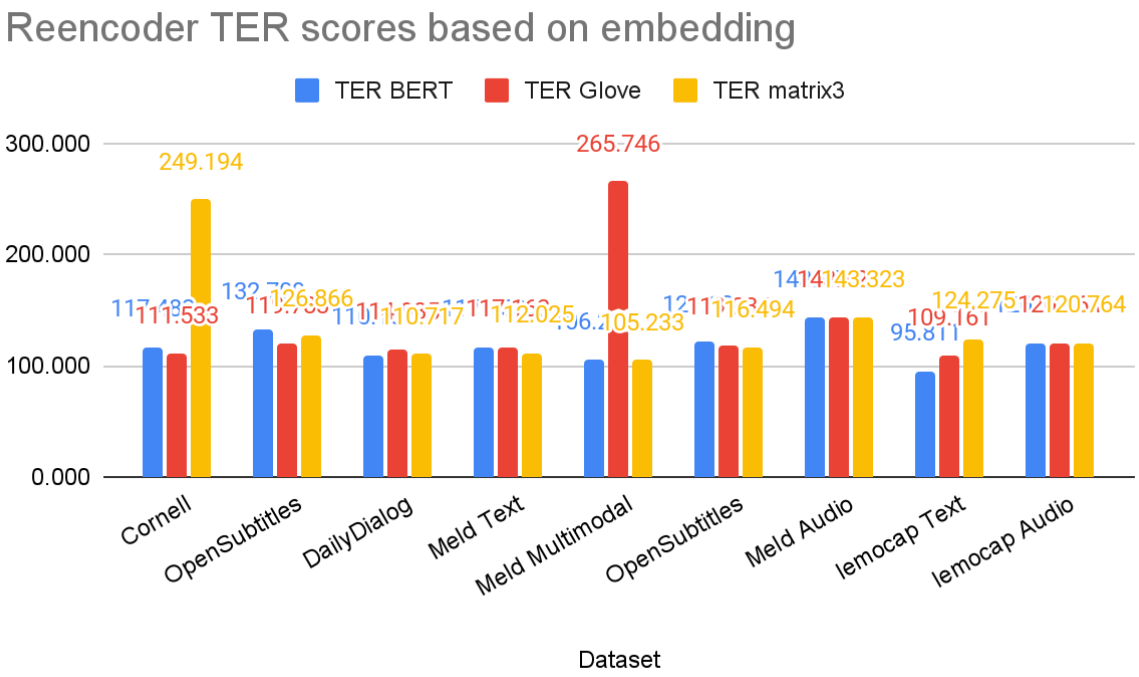

Figure a.11 TER scores for the Reencoder architecture trained on the different datasets, based on the different embedding methods.

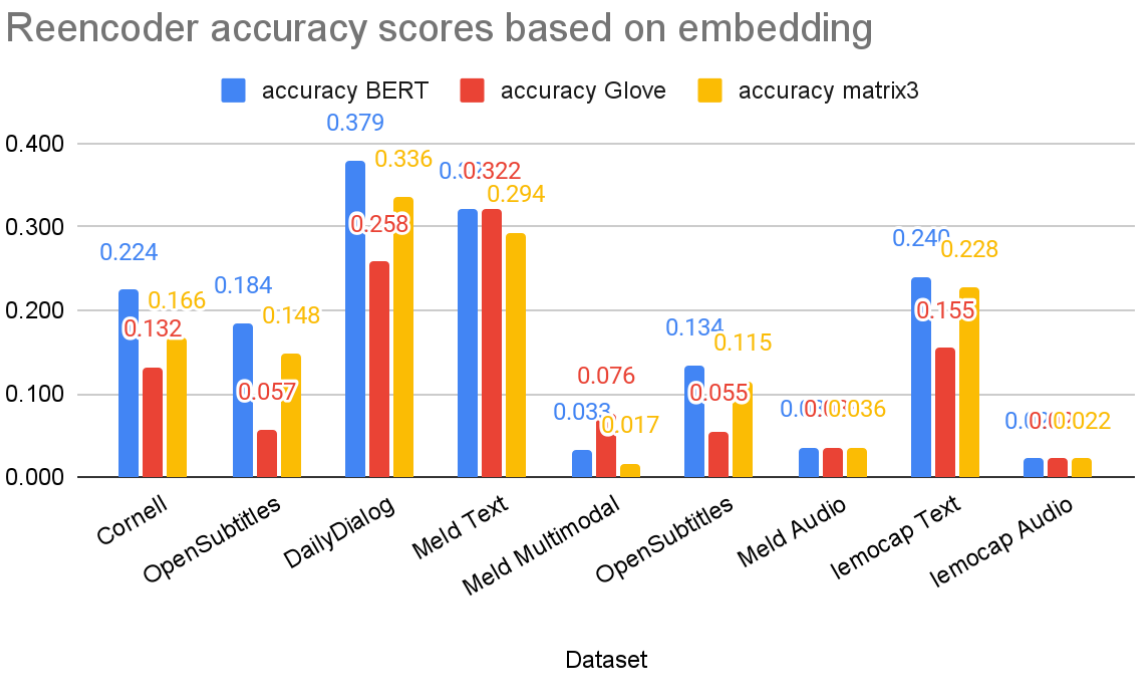

Figure a.12 Accuracy scores for the Reencoder architecture trained on the different datasets, based on the different embedding methods.

Appendix B

Computational Resources

For the purpose of this research, given the scarcity of computational resources readily available, Google Colab runtimes were used. These provided access to T4 GPUs. Running experiments on a T4 GPU in Google Colab involves navigating certain challenges and considerations. The availability of the GPU in Colab is generally favourable; however, the duration of access might be limited, and the resource allocation can fluctuate based on demand. Memory constraints on the T4 GPU have to be carefully managed, especially when working with larger models or datasets. Given the relatively moderate memory capacity of the T4, it's important to optimise batch sizes and model architectures to avoid memory exhaustion during training. Training time can be a crucial factor, as Colab sessions have time limits. It's essential to structure experiments efficiently, monitoring the training progress regularly to ensure that models are saved before the session times out. Additionally, strategic checkpointing and logging can aid in resuming experiments seamlessly in case of interruptions or time constraints. Despite these considerations, Google Colab provides a convenient and cost-effective platform for conducting machine learning experiments, leveraging the power of T4 GPUs for various tasks.
A standard Colab runtime provides for 13 gigabytes of CPU RAM, and about 193 gigabytes of Disk space.

Since the maximum runtime allowed on Google Colab is 24 hours, to avoid having to set up checkpointing, for longer running experiments a virtual machine has been set up on the Google Cloud Platform. The virtual machine could be seamlessly connected to the Colab environment, and run as long as necessary, providing the same computational power and infrastructure (a T4 GPU with 16 gigabytes of RAM), but more flexibility on the amount of system RAM and Disk Space when necessary.

The following were the specifics of the T4 GPU used for the experiments:

```
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 525.105.17   Driver Version: 525.105.17   CUDA Version: 12.0     |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|                               |                      |               MIG M. |
|===============================+======================+======================|
|   0  Tesla T4            Off  | 00000000:00:04.0 Off |                    0 |
| N/A   36C    P8     9W /  70W |      0MiB / 15360MiB |      0%      Default |
|                               |                      |                  N/A |
+-------------------------------+----------------------+----------------------+
```

Figure A. Figure A shows technical specifications for the GPU used in the experiments conducted.

The T4 GPU utilised for the experiments is characterised by its robust specifications, including 2560 CUDA Cores and 320 Tensor Cores, enabling high-speed parallel processing and efficient tensor operations. With FP16 performance reaching 65 TFLOPS and FP32 performance at 8 TFLOPS, the GPU delivers exceptional computational prowess suitable for demanding machine learning tasks. Its 16 GB GDDR6 memory, operating at a

bandwidth of 300 GB/s, facilitates swift data access and manipulation, enhancing overall performance and efficiency.

Furthermore, the T4 GPU supports various software technologies, including CUDA, NVIDIA TensorRT, and ONNX, ensuring compatibility with a wide range of machine learning frameworks and tools. With 16 PCIe lanes and a power consumption of 70W, it strikes a balance between performance and energy efficiency, making it an ideal choice for running intensive experiments. The experiment training time has been meticulously logged alongside the results, providing comprehensive insights into the performance of the conducted experiments.