BrAIcht: An AI Conversational Agent that creates plays in the style of the famous German playwright Bertolt Brecht

Author: Roland BAZ

Supervisor: Professor Tristan CAZENAVE

Master in Artificial Intelligence, Systems and Data

Paris Dauphine - PSL

This internship was conducted within

Le Laboratoire d'Analyse et de Modélisation de Systèmes pour l'Aide à la Décision

LAMSADE - PSL

November, 2023

ACKNOWLEDGEMENTS

First and foremost, I am deeply grateful to the University of Paris Dauphine, which has been my academic home for the past year.

My internship supervisor, Professor Tristan CAZENAVE, deserves special recognition for his support, guidance, and supervision during my research internship. His profound knowledge, vision, and sincerity were a constant source of inspiration at all stages of my research. I cannot thank him enough for the invaluable time, comments, and suggestions he had provided me that enriched and enhanced my work. It has been my great honour to study and work under his guidance.

I would also like to thank Guillaume GROSJEAN and Olivier ROYER for their invaluable support and guidance, as their advice has been crucial to me throughout this project.

Last but not least, I would like to express my deep gratitude to my parents, siblings, and friends for their unwavering love, support, and prayers throughout my educational journey. I am eternally grateful to my parents for the remarkable legacy of education they have instilled in me. I also thank my friends for their motivation and continuous support.

Finally, I would like to express my sincere gratitude to all those who have played an important role in my academic journey.

ABSTRACT

This project introduces *BrAIcht*, an AI Conversational Agent that can create plays in the distinctive style of the famous German playwright *Bertolt Brecht*. BrAIcht is fine-tuned using the *German LLaMA2-7B*, a large language model with 7 billion parameters and a modified version of the base LLaMA2 suitable for German language tasks. For fine-tuning, 29 plays of Bertolt Brecht and 907 of other German plays that are stylistically similar to Bertolt Brecht are used for a more diverse dataset. Due to the limited memory capacity, QLoRA is implemented to train the large language model. This method converts the model to 4-bit precision and only fine-tunes adapters, a layer used specifically for downstream tasks, while freezing all other weights. The results, based on metrics such as BLEU score and perplexity show very promising performance of BrAIcht in generating plays in the style of Bertolt Brecht.

Keywords: Natural Language Processing, Large Language Models, Conversational Agent, Fine-tuning, Generative Pre-trained Transformers, Quantization, Low-rank Adapters, LLaMA2, MistralAI

CONTENTS

LIST OF TABLES

Tables **Page**

LIST OF FIGURES

List of Abbreviations

AI: **A**rtificial **I**ntelligence

NLP: **N**atural **L**anguage **P**rocessing

CA: **C**onversational **A**gent

GPT: **G**enerative **P**re-trained **T**ransformers

BERT: **B**idirectional **E**ncoder **R**epresentations from **T**ransformers

LLaMA: Large Language Model Meta AI

BLEU: **B**iLingual **E**valuation **U**nderstudy

RNN: **R**ecurrent **N**eural **N**etwork

GPU: **G**raphics **P**rocessing **U**nits

TPU: **T**ensor **P**rocessing **U**nits

CHAPTER 1
INTRODUCTION

1.1 Background of the Study

In the fast-paced world of technology and creativity, *chatbots* and *conversational agents* have significantly changed the way we engage with *art*, *music*, and *theater*. They are equipped with advanced language skills, contextual understanding, the ability to engage in meaningful conversations, and have a remarkable impact on creative innovation. They have evolved from simple tools to true artistic partners. This marks a new era in which human imagination and artificial come together to reshape the way we create art.

In the world of theater, CAs have become important actors. They speak for themselves, invent conversations, make the audience part of the show, and add new elements to live performances. Musicians are now creating new and unusual music by collaborating with chatbot composers. These CAs know a lot about music and use clever tricks to create unique songs. In the world of visual arts, CAs create images, help artists come up with ideas, and even choose which art to show in exhibitions. These CAs are like active helpers in the creative process, making it possible to do new and exciting things in art.

One of the most influential figures in 20th century theater and literature is *Bertolt Brecht (1898-1956)*, a German playwright, poet, and theater director. Brecht's work, characterized by his innovative approach to drama, storytelling, and social commentary, has left a permanent mark on the world of theater and literature. His distinctive style, often referred to as "epic theater", sought to engage audiences intellectually and emotionally while creating a critical awareness of the social and political issues presented on stage. Brecht's unique approach to theater can be reduced to a few key elements. These include the use of alienation, a technique designed to prevent the audience from becoming emotionally immersed in the narrative and instead encourage them to adopt a critical, reflective stance. Brecht's use of simple, direct language, coupled with a focus on social and political issues, challenged traditional theatrical conventions and paved the way for a new form of dramatic expression.

1.2 Statement of the Problem

The intersection of AI and art has opened up interesting possibilities. Current AI models such as GPT-4 and LLaMA have shown that they can create texts in many different ways and even imitate famous writers, poets, or historical figures. This leads to the following research question:
Can AI capture Brecht's distinctive style and contribute to the creation of conversational agents that engage users in dialogues that reflect Brecht's style?

1.3 Motivation

This study is driven by two main objectives. First, it aims to explore the potential of AI-driven CAs in the field of theater and arts. Second, to honor the legacy of Bertolt Brecht by developing a CA capable of engaging users in conversations that reflect Brecht's distinctive style.
In addition to its contributions to the fields of AI and NLP, this research also contributes to the broader discussion of the intersection of technology and art. By developing a CA that reflects the character of Bertolt Brecht, this research aims to show that AI can create CAs capable of imitating the style of playwrights. In this way, it reveals new opportunities to connect AI and arts.

1.4 Organization of the Study

The remainder of the report is organized as follows:
Chapter two provides an overview of the current research in AI. This includes the continued development of both open-source and closed-source large-scale language models, the performance benefits of fine-tuning for specific downstream tasks, and the complexities and obstacles associated with stylized writing.
Chapter three addresses the methodology used to achieve the research objectives. First, the capabilities and efficiency of transformers in creating LLMs are highlighted. Then, a concise overview of the LLMs that are used in the development of the CA is provided. Furthermore, in this chapter, section three explains the implementation of the Parameter-Efficient Fine-Tuning (PEFT) technique for fine-tuning the models. Following this, the chapter provides information on the training phase, including details on the loss function used, the optimization methods, and the metrics used to evaluate the resulting model.

Finally, this chapter addresses the inference phase and the specific parameters that are tuned, including top-p, top-k, and temperature.
Chapter four discusses the practical results of this study. It begins with a description of the conditions under which the experiments are conducted and covers aspects such as the dataset, the best model used for our project, and the hyperparameter settings. It then presents both the numerical and descriptive results with their respective analyses and explanations.
Chapter five provides a conclusion for the study and suggests avenues for future research.

CHAPTER 2
LITERATURE REVIEW

2.1 Large Language Models

In the last two decades, neural networks have been successfully applied to language modelling, starting with feed-forward models (Bengio, Ducharme, and Vincent (2000)), recurrent neural networks, and LSTMs (Hochreiter and Schmidhuber (1997); Graves (2013)). More recently, transformer networks based on self-attention have led to important improvements, especially in capturing dependencies over long distances (Vaswani et al. (2017); Radford, Narasimhan, Salimans, Sutskever, et al. (2018); Dai et al. (2019)).

Following the remarkable success of the transformer architecture, the AI research community has been actively involved in the development of open-source and closed-source language models capable of communicating effectively with human users. For that purpose, Zhang et al. (2019) from the Microsoft AI research team has introduced *DialoGPT*, a versatile and comprehensive neural model for generating conversational responses. It is trained on an extensive dataset of 147M Reddit comment chains from 2005 to 2017. DialoGPT is specifically designed to perform exceptionally well in one-on-one conversations. Research has shown that it achieves performance that is nearly equivalent to human-generated responses, as determined by both automated and human evaluations. Moreover, compared to other strong conversational AI models, DialoGPT proves its ability to generate answers that are more contextually relevant, informative, and logically coherent.

In addition, Radford et al. (2019) present OpenAI's *GPT-2*, a 1.5B transformer-based model trained by unsupervised learning on 40GB of internet text, capable of learning and generalizing across multiple NLP tasks without requiring task-specific fine-tuning. Through multitask learning during pre-training, GPT-2 acquires the ability to perform various language-related tasks with remarkable performance. These include completing texts, answering questions, and even machine translation. Thus, it outperforms many existing models without requiring specific training for each task.

In this context, Brown et al. (2020) show that scaling language models significantly improves their ability to perform various tasks with minimal prior information, some-times even equaling or surpassing previous fine-tuning methods. They show that *GPT-3*, a language model with an unprecedented 175B parameters, achieves this improvement. It excels at tasks where it must learn from limited examples (few-shot learning) and achieves impressive results on various NLP tasks such as translation, question answering, and text completion. Nevertheless, the authors note that GPT-3 struggles in the context of few-shot learning and encounters challenges in datasets where its training is limited to extensive web data.

Furthermore, in their paper titled "Towards a Human-like Open-Domain Chatbot", Adiwardana et al. (2020) introduce *MEENA*, a model based on 2.6B parameter transformers trained using data from public discussions on social media. The authors propose the Sensibleness and Specificity Average (SSA) metric to evaluate MEENA's human-like multi-turn conversational abilities. They prove that there is a strong correlation between perplexity and SSA scores, and their results indicate that MEENA achieves a high SSA of 72 % on multi-turn evaluation due to its low perplexity training. The full version of MEENA with filtering and tuned decoding achieves an impressive SSA of 79 %, outperforming variants of DialoGPT and Xiaolce. They conclude that dialog models are well suited for model scaling and that there is a strong correlation between model size and dialog quality.

Additionally, Roller et al. (2020), an AI research team from Facebook, has gone a step further with *Blenderbot*. While previous studies have shown that scaling neural networks with more parameters and larger training data improves results, the authors emphasize that additional factors are critical to a successful chatbot. They discuss the critical skills needed for engaging in seamless conversations, such as providing interesting topics of conversation, active listening, and knowledge representation. To meet these requirements, the authors employ large-scale models and show that they can learn these conversational skills given the right training data and generation strategies. They create variants of models with different parameter sizes (90M, 2.7B, and 9.4B) and their results reveal that the best models outperform existing approaches, such as MEENA, in multi-turn dialogues, achieving higher ratings in terms of engagingness and humanness.

In a similar step, Thoppilan et al. (2022) present *LaMDA* with up to 137B parameters. These models are trained with 1.56T words from public dialog data and web text. The authors note that while scaling the model improves overall quality, it has limited impact on safety and factual grounding. To overcome these challenges, the authors demonstrate the effectiveness of fine-tuning the model with annotated data and the ability to consult external knowledge sources (information retrieval system, translators, calculators). For safety, they use a human values-based metric and show that filtering candidate responses with a fine-tuned LaMDA classifier increases the certainty of the model. For factual grounding, they introduce a groundedness metric and prove that allowing the model to consult external sources leads to answers based on known information.

One of the latest state-of-the-art LLMs for dialog applications presented by Touvron, Lavril, et al. (2023) is *LLaMA*, a collection of basic language models with 7B to 65B parameters trained on publicly available datasets without relying on proprietary data. The results of this work show that LLaMA-13B outperforms GPT3-175B on most benchmarks while being more than 10 times smaller. LLaMA-65B competes with other large models such as Chinchilla-70B and PaLM-540B. The authors' main contribution is to show that state-of-the-art language model performance can be achieved using only publicly available data, which promotes openness and accessibility in AI research. Finally, the authors note promising results from fine-tuning the models on English specific tasks.

In addition, as part of an update to LLaMA, Touvron, Martin, et al. (2023) has developed *LLaMA2*, a collection of pre-trained and fine-tuned LLMs with a scale of 7B to 70B parameters. The fine-tuned LLMs, called LLaMA2-chat, are optimised for use in dialogues. In this updated version of LLaMA2, most of the pre-training and model architecture settings are inherited from LLaMA. Key architectural differences from LLaMA include a longer context length (from 2k to 4k) and Grouped-Query Attention (GQA) for faster inference. Their results show that their models perform better than open-source chat models on most benchmarks tested. Based on human evaluations for helpfulness and safety, these models could be a suitable replacement for closed-source models.

Finally, in a recent update, Jiang et al. (2023) present *Mistral*, a language model with a size of 7B parameters, designed for high performance and efficiency. Mistral-7B not only outperforms the leading open source model with 13B parameters, LLaMA2, in all benchmark evaluations, but also outperforms the best published model with 34B parameters, LLaMA1, in reasoning, math, and code generation tasks. Their model uses Grouped-Query Attention (GQA) for faster inference and incorporates Sliding Window

Attention (SWA) to effectively process sequences of varying lengths with less computational overhead.

2.2 Fine-tuning LLMs

The pre-trained models developed in the previous section can be further fine-tuned for specific tasks. Fine-tuning involves further training the pre-trained LLM on a narrower dataset for a specific application, e.g., text classification, speech generation, text generation, or translation. There are several common techniques of fine-tuning, such as *instruction-based fine-tuning* and *reinforcement learning from human feedback*.

In instruction-based fine-tuning, a pre-trained LLM is fine-tuned with explicit instructions or guidelines during the training process. These instructions can take the form of natural language prompts, example input-output pairs, or constraints. The model is trained to generate text that adheres to the given instructions. Instruction-based fine-tuning is often used to make LLMs more controllable and to guide their output in a particular direction, e.g., by generating text summaries or adhering to certain style or content criteria.

Reinforcement learning from human feedback, however, requires a reward model that provides feedback on the quality of the output produced by the model. Human feedback is often used to create this reward model, with human annotators evaluating the quality of the model's responses and assigning rewards or points. Reinforcement learning then optimizes the model's behavior to maximize these rewards. This technique is used to improve the model's performance over time, especially in tasks where it is difficult to obtain explicit training data.

Much research has shown the effectiveness of fine-tuning LLMs on the performance of the specific downstream task. For example, Nguyen, Wilson, and Dalins (2023) has fine-tuned several LLMs such as GPT-2 for generating news summaries. They use these language models to extract several well-structured event patterns from the content of news segments. These event patterns are then evolved using a genetic algorithm and the most appropriate event pattern is selected for input to the language model to generate news summaries. They have developed a News Summary Generator (NSG) that is responsible for selecting and developing event patterns and then generating news summaries. The results of their experiments show that this news summary generator produces accurate and reliable news summaries with some generalization capability.

Additionally, in Yu et al. (2023)'s study, the authors use LLMs to address several challenges in applying machine learning techniques to financial time series data. The study focuses on NASDAQ-100 stocks and uses publicly available historical stock price information, company profiles, and historical economic and financial news. Their experimental approach includes zero-shot and few-shot inference techniques using GPT-4 and instruction-driven fine-tuning techniques using LLaMA. The results of their experiments show that their approach outperforms several benchmark models, including the commonly used ARMA-GARCH model and the gradient boosting tree model.

In the same context, Chang, Peng, and Chen (2023) use the pre-trained GPT-2 model and employ a two-step fine-tuning approach to improve the accuracy of time series prediction. First, they perform supervised fine-tuning to adapt the LLM for processing time series data, followed by additional fine-tuning for the respective prediction tasks. To maximize the benefits of pre-trained LLMs without requiring extensive parameter adjustments, they also use various Parameter-Efficient Fine-Tuning (PEFT) techniques. These methods have led to peak performance in long-term prediction.

Furthermore, Li et al. (2023) introduce *ChatDoctor*, a medical chat model that has been fine-tuned based on the LLaMA model using medical expertise. The main goal of this research is to overcome the limitations of existing LLMs such as ChatGPT, especially with respect to medical knowledge. To achieve this, they employ a dataset of 100,000 dialogues between patients and physicians obtained from a widely used online medical consultation platform. To protect privacy, these dialogues are pre-processed to remove any personally identifiable information. Additionally, the model is enhanced by incorporating a self-directed information retrieval mechanism that allows it to access and utilise real-time information from sources such as Wikipedia and curated offline medical databases. By fine-tuning the model based on actual patient-physician interactions, they greatly improve its ability to understand patient queries and provide informed advice. By equipping the model with the ability to retrieve information from credible online and offline sources, they observe a significant improvement in the accuracy of its responses.

Besides, Shoham and Rappoport (2023) present *CPLLM*, an LLM that is fine-tuned to predict clinical diseases. This fine-tuning process involves quantization and the use of prompts with the goal of predicting whether patients will receive a particular disease at their next visit or at a subsequent diagnosis, based on their medical history. To evaluate the effectiveness of CPLLM, they perform a comparison with several baseline models, such as logistic regression, RETAIN, and Med-BERT, the current leading model for disease prediction using structured electronic health records (EHR). Their experimental results show that CPLLM outperforms all tested models in terms of both PR-AUC and ROC-AUC metrics, showing significant improvements over the baseline models.

Moreover, Yang, Tang, and Tam (2023) introduce *InvestLM*, which is a model developed for finance. They refine it based on the LLaMA 65B model using an instruction dataset that focuses on financial investments. This dataset covers a wide range of finance topics, including Chartered Financial Analyst (CFA) exam questions, SEC filings, and Stack Exchange discussions on quantitative finance. InvestLM demonstrates strong financial text comprehension skills and provides valuable answers to investment-related questions. In particular, financial experts such as hedge fund managers and research analysts consider InvestLM's answers to be on par with those of advanced commercial models such as GPT-3.5, GPT-4, and Claude-2.

In addition, Zheng, Abdel-Aty, Wang, Wang, and Ding (2023) present the *TrafficSafetyGPT*, a LLaMA-based model developed to overcome the limitations of LLMs in traffic safety. They use the TrafficSafety-2k dataset, which includes human labels from government-generated guidance documents and ChatGPT-generated instruction-output pairs. They report that TrafficSafetyGPT shows superior performance in generating responses with reliable road safety knowledge, allocating efficient computational resources and saving training time.

In the field of theater and arts, however, there are few projects that have taken advantage of the fine-tuning of LLMs to produce poetry and plays and to imitate the particular writing styles of playwrights. A few examples in this regard are the works of Rosa et al. (2020) and Lo, Ariss, and Kurz (2022). Bangura, Barabashova, Karnysheva, Semczuk, and Wang (2023) present a GPT-2 model for the automatic generation of German drama texts. They present a two-step approach to building their model. First, a GPT-2 model is fine-tuned to generate scene outlines based on keywords. Then, they fine-tune another model to convert these outlines into complete scenes. They use two datasets: the German drama corpus and the German text archive. To evaluate the effectiveness of their method, they compare their models to the GPT-2 base models. They report that the results are promising when evaluated quantitatively. However, when evaluated qualitatively, they find that the quality of the generated texts seems to be poor. They conclude that this may be due to problems related to the quality of the training data used.

In summary, this chapter has provided a comprehensive overview of the extensive research on the development of novel open-source and closed-source language models, with the goal of reshaping the field of AI and NLP. Many studies have shown that scaling these models and training them with larger datasets leads to better performance. In addition, researchers are actively working to develop new evaluation metrics such as Sensitivity and Specificity Average (SSA) to better assess the effectiveness of their models, taking into account safety and the removal of possible biases in the generated responses. In addition, evidence collected in previous studies shows that applying fine-tuning to these models consistently results in significant performance improvements on the specific downstream task. However, in the context of art and theater, a domain where such applications have rarely been used, our research represents a groundbreaking contribution. By employing the power of a large-scale language model to mimic the styles of playwrights, we not only expand the boundaries of the field, but also open new avenues for exploration and analysis in a previously unexplored area. In this sense, our work represents a novel and valuable addition to the existing literature.

CHAPTER 3
METHODOLOGY

3.1 Transformer-based Models

Transformers are very popular architectures used in many NLP applications. They use an encoder-decoder structure, as shown in Figure 3.1. The encoder is called BERT and takes an input sequence of vectors $\mathbf{x} = (x_1, \ldots, x_n)$ and converts it into a sequence of continuous vectors called $\mathbf{z} = (z_1, \ldots, z_n)$. Then, using the information in **z**, the decoder, called GPT, generates an output sequence $\mathbf{y} = (y_1, \ldots, y_m)$. At each step, the model generates the output in an *auto-regressive* way, one element at a time, using the previously generated symbols as additional input to generate the next symbol.[1]

Moreover, the effectiveness of these transformers, as presented in the work of Vaswani et al. (2017), in various NLP tasks can be attributed to the innovative approach of the *attention mechanism*. This mechanism was originally proposed by Bahdanau, Cho, and Bengio (2016) in their work titled "Neural Machine Translation by Jointly Learning to Align and Translate". They introduce the idea of using an attention mechanism in neural machine translation. This mechanism allows the model to assign different weights to different words in the source sentence as it generates each word in the target sentence. This allows the model to focus on relevant information during the translation process, which greatly improves the quality of translations.

They define the conditional probability as:

$$p\left(y_i \mid y_1, \ldots, y_{i-1}, \mathbf{x}\right) = g\left(y_{i-1}, s_i, c_i\right) \tag{3.1}$$

where s_i is the recurrent neural network (RNN) hidden state for time i, computed by:

$$s_i = f\left(s_{i-1}, y_{i-1}, c_i\right) \tag{3.2}$$

The context vector c_i is determined by a set of annotations $(h_1, \ldots, h_{T_x})$ generated by an encoder from an input sentence. Each annotation h_i contains information about the entire input sentence, focusing on the sections near the i-th word of the input sequence.

[1]For more information about Transformers, please visit: `https://towardsdatascience.com/transformers-89034557de14`

The context vector c_i is then computed as the weighted sum of these annotations h_i:

$$c_i = \sum_{j=1}^{T_x} \alpha_{ij} h_j \tag{3.3}$$

The weight α_{ij} of each annotation h_j is computed by:

$$\alpha_{ij} = \frac{\exp\left(e_{ij}\right)}{\sum_{k=1}^{T_x} \exp\left(e_{ik}\right)} \tag{3.4}$$

where

$$e_{ij} = a\left(s_{i-1}, h_j\right) \tag{3.5}$$

is an alignment model that assigns a score to measure the degree of compatibility between elements in the input around position j and the output at position i. This score is determined considering the hidden RNN state s_{i-1} that occurs immediately before y_i (Eq.(3.1)) is omitted, and the j-th annotation h_j from the input sentence.

Thus, following the success of the attention mechanism introduced by Bahdanau et al. (2016) in machine translation, Vaswani et al. (2017) present the *Transformer*, an architecture that relies heavily on *self-attention* mechanisms, allowing it to effectively capture dependencies between words in a sentence without the need for recurrent or convolutional structures. The attention mechanism of the transformer enables parallel processing, significantly speeding up training and inference compared to traditional sequence-to-sequence models.

Figure 3.1 shows the transformer-model architecture presented by Vaswani et al. (2017). It consists of *Multi-head Attention* layers. They define multi-head attention as:

$$\operatorname{MultiHead}(Q, K, V) = \operatorname{Concat}\left(\text{ head }_1, \ldots, \text{ head }_{\mathrm{h}}\right) W^O \tag{3.6}$$

$$\text{head }_{\mathrm{i}} = \operatorname{Attention}\left(Q W_i^Q, K W_i^K, V W_i^V\right) \tag{3.7}$$

where

$$\operatorname{Attention}(Q, K, V) = \operatorname{softmax}\left(\frac{Q K^T}{\sqrt{d_k}}\right) V \tag{3.8}$$

They refer to the attention mechanism as "Scaled Dot-Product Attention" (as shown in Fig. 3.2 and Eq.(3.8)). This mechanism takes input in the form of queries (Q) and keys (K), both having dimension d_k, and values (V) with dimension of d_v. It calculates the dot products between the query and all the keys, then normalizes each result by dividing

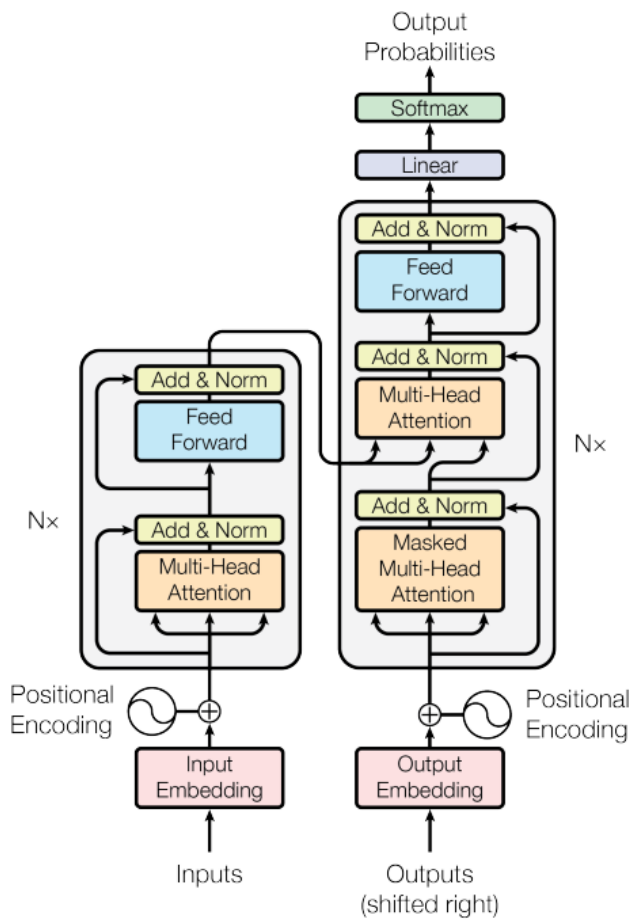

Figure 3.1

Transformer-model architecture. (Vaswani et al. (2017))

by the square root of d_k, and finally uses a softmax function to derive the weights that determine the importance of each value.

In summary, the introduction of transformers characterized by the inclusion of the attention mechanism represents an advancement in the field of AI and NLP. The efficiency of transformers have revolutionized the world of AI applications and made them a foundation of modern machine learning methods. Above all, their integration into LLMs has contributed significantly to the performance of such models. As the broader AI community continues to utilise the impressive capabilities of transformers, their impact will continue and serve as a foundation upon which further innovation and advancements in artificial intelligence will be built.

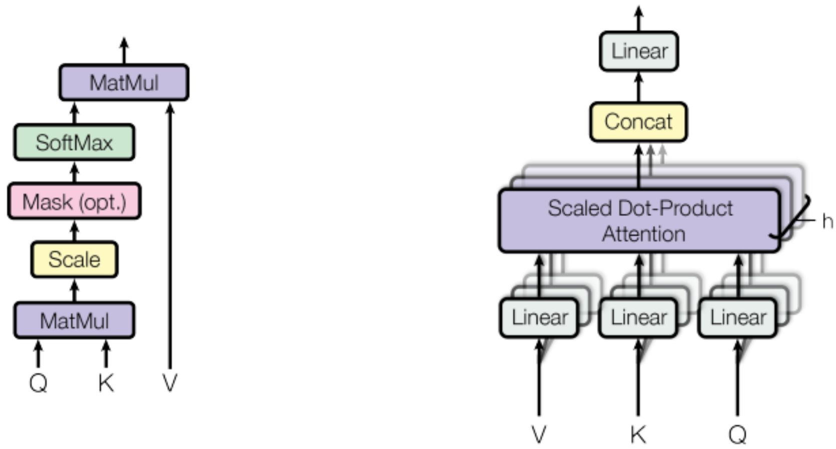

Figure 3.2

(left) Scaled Dot-Product Attention (right) Multi-Head Attention consists of several attention layers running in parallel. (Vaswani et al. (2017))

3.2 Large Language Models

LLMs that generate text are called *Causal LMs*. This refers to a type of language models, such as the GPT series, that are designed to produce text in a causal or auto-regressive manner. In this context, "causal" means that the model generates text token by token in a left-to-right sequence, where each token depends on the tokens that preceded it in the text. In this project, we test two open-source causal LLMs called *LLaMA2* and *Mistral*, developed by MetaAI and MistralAI, respectively.

Touvron, Martin, et al. (2023) present LLaMA2 as an extension of their earlier work on LLaMA (Touvron, Lavril, et al. (2023)), a pre-trained language model designed specifically for CA tasks. LLaMA2 builds on this foundation and introduces several enhancements to improve its performance on various NLP tasks. Notable improvements in LLaMA2 include updates to the data mixtures, a training dataset with 40% more tokens than LLaMA1, a doubling of context length, and the introduction of Grouped-Query Attention (GQA) to improve the scalability of inference for larger models. Table 3.1 below provides a detailed comparison of the features of LLaMA2 models versus LLaMA1 models.

Model	Params	Context Length	GQA	Tokens	LR
LLaMA1	7B	2k	No	1.0T	3.0×10^{-4}
	13B	2k	No	1.0T	3.0×10^{-4}
	33B	2k	No	1.4T	1.5×10^{-4}
	65B	2k	No	1.4T	1.5×10^{-4}
LLaMA2	7B	4k	No	2.0T	3.0×10^{-4}
	13B	4k	No	2.0T	3.0×10^{-4}
	34B	4k	Yes	2.0T	1.5×10^{-4}
	70B	4k	Yes	2.0T	1.5×10^{-4}

Table 3.1

Comparison between the attributes of LLaMA1 and LLaMA2. (Touvron, Lavril, et al. (2023),Touvron, Martin, et al. (2023))

The authors evaluate the performance of LLaMA2 by testing it on a set of standard benchmark datasets and then comparing its performance to that of other leading open source-models. Their results show that LLaMA2 outperforms these models on a variety of tasks, including question answering, text classification, and dialogue generation. Specific details of the results can be found in table 3.2.

Model	Size	Code	Math	MMLU	BBH	AGI Eval
LLaMA2	7B	16.8	**14.6**	**45.3**	**32.6**	**29.3**
	13B	24.5	**28.7**	54.8	**39.4**	**39.1**
Falcon	7B	5.6	4.6	26.2	28.0	21.2
	13B	15.2	12.6	**55.4**	37.2	37.0
MPT	7B	**20.5**	4.9	26.8	31.0	23.5
	13B	**28.9**	9.1	46.9	38.0	33.8

Table 3.2

Performance comparison of open-source models. (Touvron, Lavril, et al. (2023))

In addition, the authors perform a thorough comparison between LLaMA2-70B and various closed-source models. The results show that LLaMA2-70B achieves comparable performance to most of the closed-source models, although it has a smaller number of parameters. The detailed results of this comparison are shown in table 3.3.

Benchmark (shots)	**GPT-3.5**	**GPT-4**	**PaLM**	**PaLM-2-L**	**LLaMA2**
MMLU (5-Shot)	70.0	**86.4**	69.3	78.3	68.9
TriviaQA (1-shot)	-	-	81.4	**86.1**	85.0
Natural Questions (1-shot)	-	-	**29.3**	**37.5**	33.0
GSM8K (8-shot)	57.1	**92.0**	56.5	80.7	56.8
HumanEval (0-shot)	48.1	**67.0**	26.2	-	29.9
BIG-Bench Hard (3-shot)	-	-	52.3	**65.7**	51.2

Table 3.3

Comparison to closed-source models. (Touvron, Lavril, et al. (2023))

On the other hand, Jiang et al. (2023) introduce Mistral with 7B parameters. They use Grouped-Query Attention (GQA) for faster inference, coupled with Sliding Window Attention (SWA) to effectively handle sequences of arbitrary length with a reduced inference cost. Mistral's architecture is shown in table 3.4.

Parameter	dim	n_layers	head_dim	n_heads	context_len	vocab_size
Value	4096	32	128	32	8192	32000

Table 3.4

Mistral architecture. (Jiang et al. (2023))

Their results show that Mistral-7B outperforms LLaMA models on several benchmarks. Some of the results are shown in table 3.5.

Model	**Modality**	**MMLU**	**HellaSwag**	**WinoG**	**PIQA**
LLaMA2 7B	Pre-trained	44.4%	77.1%	69.5%	77.9%
LLaMA2 13B	Pre-trained	55.6%	**80.7%**	72.9%	80.8%
Code-LLaMA 7B	Fine-tuned	36.9%	62.9%	62.3%	72.8%
Mistral 7B	Pre-trained	**60.1%**	**81.3%**	**75.3%**	**83.0%**

Table 3.5

Mistral vs. LLaMA. (Jiang et al. (2023))

In summary, the results clearly demonstrate that LLaMA2 and Mistral are the best performing LLMs to date and consistently excel in a variety of benchmarks when compared to various open-source and closed-source models. Most importantly, they achieve performance on par with several established closed-source models. These results emphasize LLaMA2's and Mistral's impressive potential and competitiveness in the field of NLP and highlight their importance as a powerful language model and may in the future be a replacement for closed-source models.

3.3 Data Pre-processing and Encoding

Before training or fine-tuning an LLM, it is essential to start pre-processing and encoding the data. This is because the quality of the data plays a critical role in determining the performance of the LLM. The success of any NLP project depends on the quality and careful preparation of the data used.

Our dataset includes 29 plays by the famous German playwright Bertolt Brecht and 907 plays by other German playwrights sharing similar style to Bertolt Brecht. From this dataset, we extract the *cues* which in the field of theater refer to specific signals or instructions that prompt actors, technicians, or other members of the production team to perform specific actions or tasks at precise moments during a scene. Using a method known as *prompt engineering*, we create the prompts provided to the model, as shown in Table 3.6. After that, we introduce certain special tokens to better convey the structure of the data. First, we insert a *beginning of sentence* token denoted as <*s*> to signal the beginning of the sentence. Similarly, an *end of sentence* token, denoted as </*s*>, is also inserted to signal the end of the sentence. We also insert a *pad* token, marked as <*pad*>.

At the end, we add some *special tokens* such as *human* and *assistant*. Finally, we tokenize the keywords to make them conform to the LLaMA2 format. This careful process of data preparation ensures that the model receives well-structured and informative inputs.

User:	cue 1
BrAIcht:	cue 2
...	...
User:	cue n-1
BrAIcht:	cue n

Table 3.6

Prompt format for training the LLM.

3.4 Parameter-Efficient Fine-Tuning (PEFT)

Fine-tuning an LLM is an extraordinarily resource-intensive process that requires significant computational power, primarily in the form of GPUs and even TPUs. With a single A6000 GPU with 48GB of RAM on the LAMSADE server, we choose to fine-tune the smallest LLaMA2 model with 7B parameters, a model size that best fits our available GPU resources. The larger models (13B and 70B parameters), while promising, require significantly more memory and compute capacity that our single GPU cannot provide. Moreover, to make the best use of our available resources, we take an alternative approach to fine-tuning. Instead of performing a full fine-tuning process, we use a special technique known as Quantized Low-rank Adapters (QLoRA). This approach allows us to optimize the performance of the model and adapt it to specific tasks while reducing the high memory requirements and training time associated with fine-tuning.

The main idea behind the QLoRA technique developed by Dettmers, Pagnoni, Holtzman, and Zettlemoyer (2023) is that it uses a novel high precision technique to quantize a pre-trained model to 4-bit (Frantar, Ashkboos, Hoefler, and Alistarh (2023), Frantar and Alistarh (2022)), and then adds a small set of learnable low-rank adaptor weights tuned by backpropagating gradients through the quantized weights. They introduce several contributions aimed at reducing memory usage without sacrificing performance. First, they use *4-bit NormalFloat*, an optimally quantized data type for normally distributed data that gives better results compared to 4-bit integers and 4-bit floats. Second, they use *double quantization* to quantize the quantization constants, which saves about 0.37

bits per parameter, equivalent to 3GB for a model with 65B parameters. Third, they employ *paged optimizers*, which use NVIDIA unified memory to prevent memory spikes caused by gradient checkpointing during the processing of mini-batches with extended sequence lengths.

They introduce blockwise k-bit *quantization*, which converts an input from a format with a larger amount of information to a format with less information. This includes converting a data type with a larger bit size to a data type with a smaller bit size, such as converting 32-bit floating-point numbers to 8-bit integers (Dettmers, Lewis, Belkada, and Zettlemoyer (2022)). To ensure that the full range of the lower bit size data type is used effectively, the input data type is usually adjusted to fit within the range of the desired data type. This fitting is achieved by normalizing the input elements by the absolute maximum of the input elements, as shown in Eq.(3.9).

$$\mathbf{X}^{\text{Int8}} = \text{round}\left(\frac{127}{\text{absmax}(\mathbf{X}^{\text{FP32}})} \cdot \mathbf{X}^{\text{FP32}}\right) = \text{round}(c^{\text{FP32}} \cdot \mathbf{X}^{\text{FP32}}) \tag{3.9}$$

where c is the quantization constant or quantization scale. Dequantization is the inverse and is given by Eq.(3.10).

$$\text{dequant}(c^{\text{FP32}}, \mathbf{X}^{\text{Int8}}) = \frac{\mathbf{X}^{\text{Int8}}}{c^{\text{FP32}}} = \mathbf{X}^{\text{FP32}} \tag{3.10}$$

They also present *LoRA*, a technique aimed at reducing the memory requirements of an LLM during the training process (Hu et al. (2021)). This is achieved by using a small set of adjustable parameters, called adapters, without updating the entire set of model parameters, which remain fixed. During stochastic gradient descent, gradients are propagated through the pre-trained model weights that do not change, and these gradients are used to update the adapters to optimize the loss function. LoRA extends a linear projection by an additional factorized projection.
Given a projection **XW** = **Y** with $\mathbf{X} \in \mathbb{R}^{b \times h}$ and $\mathbf{Y} \in \mathbb{R}^{h \times o}$, LoRA computes:

$$\mathbf{Y} = \mathbf{X}\mathbf{W} + s\mathbf{X}\mathbf{L}_1\mathbf{L}_2 \tag{3.11}$$

where $\mathbf{L}_1 \in \mathbb{R}^{h \times r}$ and $\mathbf{L}_2 \in \mathbb{R}^{r \times o}$ and s is a scalar.

In summary, QLoRA, which combines the two components described above, is an effective method for optimizing LLMs by balancing model efficiency and performance. By introducing low-rank adapters and quantization techniques, QLoRA significantly reduces model size and computational requirements. The advantage of QLoRA is its ability to achieve these efficiency gains with only a small performance degradation. In practice, this means that larger models can be deployed in resource-constrained environments or applications without compromising the model's ability to perform certain tasks well. QLoRA provides a compelling solution to make modern language models more viable and efficient, so that they can be used more broadly in a range of real-world scenarios.

3.5 Training

To fine-tune the base models, we first obtain their weights from Hugging Face, a widely used platform where the AI community shares their open-source models and contributes to the development and progress of AI research (Wolf et al. (2020)).

3.5.1 Objective function and optimization

The training consists of reconstructing the dialog extracts according to a classical task used for generative auto-regressive models. Each extract is transformed into a sequence of tokens $U = (u_1, \ldots, u_n)$ and the parameters Θ of the model are optimized by minimizing the inverse of the log-likelihood function *(negative log-likelihood)*:

$$L(U, \Theta) = -\sum_{i=1}^{n} \log P(u_i | u_1, \ldots, u_{i-1}, \Theta) \tag{3.12}$$

To minimize the objective loss function $L(U, \Theta)$ from equation (3.12), we use an optimization technique called *AdamW* (Zhuang, Liu, Cutkosky, and Orabona (2022), Zhou, Xie, and YAN (2023)), a variant of *Adam*, an optimization technique commonly used for training machine learning and deep learning models.

Adam was first proposed by Kingma and Ba (2017). The paper presents Adam as an optimization algorithm that takes advantage of *stochastic gradient descent* with *momentum* and *RMSprop*. It introduces a mechanism to customise the *learning rates* for each parameter during training, which is achieved by maintaining moving averages of the first and second moments of the gradients. This learning rate makes Adam well suited for dealing with *sparse gradients* and *noisy data*. In addition, one of the most important features of Adam is its efficient *bias correction*, which helps to mitigate initial biases in the moment estimates, especially in the early training phases. An iteration of

Adam is given by:

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha \mathbf{m}_k \odot \mathbf{v}_k \tag{3.13}$$

where $\alpha > 0, \mathbf{m}_k, \mathbf{v}_k \in \mathbb{R}^d$ and $\odot$ denotes the component-wise division operator:

$$\mathbf{m}_k \odot \mathbf{v}_k := \begin{bmatrix} (m_k)_1 \cdot (v_k)_1 \\ (m_k)_2 \cdot (v_k)_2 \\ \vdots \\ (m_k)_d \cdot (v_k)_d \end{bmatrix} := [\frac{[\mathbf{m}_k]_i}{[\mathbf{v}_k]_i}]_{i=(1,\ldots,d)} \tag{3.14}$$

where

$$\mathbf{m}_k = (1-\beta_1)\frac{\sum_{j=0}^{k} \beta_1^{k-j}\mathbf{g}_j}{1-\beta_1^{k+1}} \tag{3.15}$$

and

$$\mathbf{v}_k = \sqrt{(1-\beta_2)\frac{\sum_{j=0}^{k} \beta_2^{k-j}\mathbf{g}_j \otimes \mathbf{g}_j}{1-\beta_2^{k+1}}}. \tag{3.16}$$

Here $\beta_1, \beta_2 \in (0,1)$ and $\otimes$ denotes the Hadamard or component-wise product:

$$\mathbf{g}_k \otimes \mathbf{g}_k = [[\mathbf{g}_k]_i^2]_{i=1}^d \tag{3.17}$$

AdamW is a variant of Adam and shows a remarkable superiority in generalization over vanilla Adam. In particular, the main difference between the two optimization techniques is that in the original Adam optimizer, *weight decay* is directly integrated into the optimization process. It is included in the update step together with the gradient. AdamW, on the other hand, decouples the weight decay from the optimization process. It applies the weight reduction separately after it has calculated the gradient, but before updating the parameters, which makes it more effective as a regularization technique.

In summary, the choice of optimization technique is a crucial factor when it comes to reducing loss and ensuring that the model learns effectively. For example, in this project we use AdamW. This method has a significant impact on how efficiently the model learns and achieves its goals. So it is important to choose the right optimization technique carefully, as it strongly influences the success and performance of the model in research and real-world applications.

3.5.2 Evaluation metrics

Once the training phase has finished, we assess our models' performance using two crucial metrics: the *Perplexity* and the *BLEU* score.

Perplexity assesses how accurately a model predicts the probability distribution of the following word in a sequence. A lower perplexity score indicates that the model understands the data better, leading to more accurate predictions. We use perplexity to assess how well the model captures the underlying linguistic coherence and patterns. The minimum value of perplexity is 1 and is calculated as in equation (3.18).

$$PP(\Theta, \mathbf{U}_N) = \sqrt[N]{\prod_{i=1}^{N} \frac{1}{P(u_i|u_1, \ldots, u_{i-1})}} \tag{3.18}$$

Perplexity alone means nothing and is usually used to compare the performance of different models. Also note that the perplexity corresponds to the closest exponential value of the loss function $L(U, \Theta)$ that is minimized during training.

The BLEU score was proposed by Papineni, Roukos, Ward, and Zhu (2002). The authors define the BLEU score as a metric for translation models that evaluates a candidate in relation to a collection of references. It measures the similarity between a candidate text and a reference text by comparing common token sequences (n-grams). In some cases, such as sequence-to-sequence tasks, there may be more than one reference for a single candidate. The BLEU score ranges from 0 to 1 and is more accurate the closer the value is to 1. It is impossible to achieve a score of 1, and a score of more than 0.4-0.5 is usually considered a good result.

$$BLEU = BP.exp \sum_{n=1}^{N} W_n \log MP \tag{3.19}$$

where

$$BP = \begin{cases} 1 & \text{if can} > \text{ref} \\ e^{(1-\text{ref/can})} & \text{if can} \leq \text{ref} \end{cases} \tag{3.20}$$

BP is the Brevity Penalty and is used to penalize the predictions that are too short when compared to the references. N is the n-gram order that is used for calculation. In general, it is 4 i.e., uni, bi, tri, and tetra grams are all considered for calculation. They are weighted by W_1, W_2, ... which add up to 1 based on N i.e., for N=4, $W_1 = W_2 = W_3 = W_4 = \frac{1}{4}$. Finally, MP is the modified precision.

To summarise, in addition to perplexity and the BLEU score, there are numerous other evaluation metrics in the field of NLP to assess the quality of the generated text. However, for our work, we have mainly focused on these two generally recognised metrics. It is important to note that while quantitative metrics provide valuable insights, one of the most important evaluation methods in the NLP field is human evaluation. AI researchers are highly dependent on human evaluation to assess the overall performance of models, as they offer a better understanding of text quality and its alignment with human preferences and expectations.

3.6 Inference

The next crucial step in our work is inference which evaluates the model's performance in producing text and making predictions after the training phase is completed and the best model is selected based on the previously established criteria. *Top-p*, *top-k*, and *temperature* are three important metrics we use in this phase to evaluate the capability of LLMs. These metrics give us important information about how the model generates text, which helps us to adjust its behavior and maintain the quality of the output.

Top-p sampling, also known as *nucleus sampling*, is a metric used to achieve a balance between diversity and the inclusion of high probability words when selecting tokens. In this approach, the tokens are selected from the most probable top p whose total probability exceeds a certain threshold, denoted by p. This ensures that the generated output remains diverse while maintaining relevance in the given context. The value of top-p is a hyperparameter that needs to be adjusted depending on the specific application and the desired quality of the output. In practice, it is recommended to increase the threshold p if the generated text is too narrow and lacks variety. Conversely, if the generated text is excessively diverse and contains irrelevant words, reducing the probability threshold p should be considered.

Furthermore, the top-k sample involves selecting the k tokens with the highest probability of being the subsequent tokens in a sequence, as described by the output distribution of the language model. Then, k tokens are selected from this subset of top k tokens through a stochastic sampling process. The top-k sampling technique proves to be advantageous in the context of text generation as it allows the creation of texts that are characterized by diversity and creativity. This is achieved by allowing the language model to explore alternative potential predictions for the next token, rather than just favoring the most likely option. The exact value of k can be varied according to the desired level of creativity or variety for the generated text.

On the other hand, temperature is a hyperparameter that affects how the LLM generates tokens by changing the probability distribution. By manipulating the temperature we can regulate the variety and ingenuity of the text generated. Setting the temperature to higher values produces more varied output, while lower temperature settings produce more focused and predictable text. The temperature is integrated into the softmax function, which converts the raw values into probabilities.

$$P(x_i) = \frac{e^{\frac{x_i}{T}}}{\sum_j^V e^{\frac{x_j}{T}}} \tag{3.21}$$

where

- $P(x_i)$ is the probability of generating token i
- x_i is the raw score for token i
- T is the temperature
- $\sum_j^V$ is the sum over all tokens in the vocabulary

As can be seen in equation (3.21), the probabilities are more concentrated on the tokens with the highest probability at a low temperature. The generated text is more deterministic and less diverse. At a high temperature, however, the probabilities are more evenly distributed, resulting in a more diverse and creative selection of words.

To conclude this chapter, we present the methodology used in our research project. We use two widely recognized LLMs, namely LLaMA2 and Mistral. These models are fine-tuned using two different datasets: one comprising 907 plays by German playwrights and the other comprising 29 plays by Bertolt Brecht. Considering the hardware limitations, we use the QLoRA technique to fine-tune our models. We then evaluate their performance based on perplexity and BLEU score. Finally, we fine-tune the top-p, top-k, and temperature parameters during the inference process, customizing them to the relevance and creativity desired in the output, aligning with the specific requirements of our application.

CHAPTER 4
RESULTS AND INTERPRETATION

4.1 Experiment Settings

4.1.1 Dataset

In our research, we collect two different datasets. The first dataset consists of 907 German plays, while the second dataset focuses on the works of Bertolt Brecht and includes 29 plays. Our experiments start with the pre-processing of the data, where we extract the cues to drive the model's responses. After this step, we have a total number of 542,474 cues from the German plays and 17,740 cues from the Brecht dataset.

Our approach involves a two-stage fine-tuning process. In the first training session, we fine-tune the LLMs using the German plays dataset. We then perform a further fine-tuning of the resulting model using the Brecht dataset. We opt for this two-stage approach because the Brecht dataset is relatively small. To make the dataset more diverse, we include the German plays that show stylistic similarities with Brecht's works. For the first step of fine-tuning, we split the data into a training set (80%) and a validation set (20%). However, in the second fine-tuning step, we use a split of 90% for training and 10% for validation. More detailed information about the split of the dataset is provided in table 4.7

	Number of cues	**Training set**	**Validation set**
German plays	542,474	433,979	108,495
Brecht plays	17,740	15,966	1,774

Table 4.1

BrAIcht datasets.

4.1.2 Models

Since the initial release of LLaMA and Mistral base models, the open source and academic research community has seen rapid growth in the creation of more advanced language models. These recent improvements have allowed LLaMA2 and Mistral models to compete with OpenAI's ChatGPT, which is based on GPT-3.5, and in some cases even with the more powerful GPT-4. A major drawback, however, is that most of these groundbreaking advances are concentrated primarily in the English language. This limitation is mainly due to the fact that large open-source models were mainly trained on monolingual English data. While there have been some attempts to refine these models for second languages or multilingual purposes, the resulting models often have limited capabilities and are influenced by the US-centric bias of the English data.

We first try LLaMA2-7B and Mistral-7B but we end up with average results. This is because the fact that our dataset is in German poses a major challenge. LLaMA2 and Mistral are primarily pre-trained on English corpus. To address this problem, we find a German-focused language model called LeoLM, which is built on LLaMA2. LeoLM is developed by LAION and HessianAI and is available in two versions, 7B and 13B, both trained with a context length of 8k.[1] It uses techniques such as *linear RoPE scaling* and *Flash Attention 2* to improve training efficiency. To enhance the model in German, they perform a second stage of pre-training using LLaMA2 weights and a German corpus with 65B tokens. This approach significantly improves the performance of the model in German compared to the LLaMA2 baseline models.

So once we have selected the appropriate model for our specific use case, we start fine-tuning with the German dataset as described in section 4.1.1. The technical details, including the BitsAndBytes configuration, LoRA, and the training parameters used in this first stage of fine-tuning, are listed in table 4.2 and table 4.3 respectively.[2]

The parameter *lora_alpha* in table 4.2 affects the rank of the low-rank adaptation matrix used during the fine-tuning process. A higher value for *lora_alpha* means that the matrix has a higher rank. If the matrix has a higher rank, it means that more parameters will be adjusted during fine-tuning, which may help the model to better adapt to new data. On the other hand, a lower value of *lora_alpha* results in a matrix with a lower rank, which means that fewer parameters are changed and more of the original structure of the pre-trained model is preserved during the fine-tuning process.

[1] `https://huggingface.co/LeoLM/leo-hessianai-7b`

[2] `https://laion.ai/blog/leo-lm/`

Parameter	Value
Load_in_4bit	True
Bnb_4bit_quant_type	"nf4"
Bnb_4bit_compute_dtype	"float16"
Use_nested_quant	False
LoRA_r	32
LoRA_alpha	16
LoRA_dropout	0.2

Table 4.2

BitsAndBytes and LoRA parameters.

Parameter	Value
Epochs	2
Weight_decay	0.01
Train_batch_size	1
Validation_batch_size	1
Optimizer	"paged_adamW_32bit"
Learning_rate	2e-4
Warmup_ratio	0.03
Maximum_gradient_norm	0.3
Maximum_sequence_length	1024

Table 4.3

1st*-stage training parameters.*

In addition, the parameter *lora_r* shown in table 4.2 determines the rank of the low-rank matrix used for the adaptation during fine-tuning. Using lower ranks reduces the number of parameters that need to be fine-tuned during the adaptation. This can result in faster training and less memory usage, which is beneficial for devices with limited computing resources. Additionally, restricting the rank of the adaptation matrix can also prevent the model from adapting too much to the fine-tuning data, thereby preserving its ability to generalize.

Finally, the *lora_dropout* is the dropout rate applied during the fine-tuning process. Dropout is a regularization technique in which a portion of neurons are randomly deactivated or "turned off" during training to counteract over-fitting. A higher dropout rate means that a larger proportion of neurons are deactivated in each training iteration. While the dropout rate effectively prevents over-fitting, a dropout rate that is too high can lead to under-fitting, where the model learns inadequately from the training data. Conversely, a very low dropout rate can cause the model to over-memorize the training data, reducing its ability to generalize when confronted with new, unseen data.

In the second stage of fine-tuning, we use the Brecht dataset together with the same Lora and BitsAndBytes hyperparameters. However, we make some changes to the training settings, which are listed in the table 4.4 below.

Parameter	**Value**
Epochs	4
Weight_decay	0.01
Train_batch_size	1
Validation_batch_size	1
Optimizer	"paged_adamW_32bit"
Learning_rate	2e-4
Warmup_ratio	0.03
Maximum_gradient_norm	0.3
Maximum_sequence_length	1024

Table 4.4

2nd*-stage training parameters.*

4.2 Results and Analysis

After the initial training phase, in which the German LLaMA2 model is fine-tuned using 907 German plays, we evaluate the perplexity of the model before and after fine-tuning. The results show a significant reduction in perplexity, dropping from 15.96 to 10.19, which is a considerable improvement. In the second training session, in which we further fine-tune the resulting model using 29 plays by Bertolt Brecht, we observe an even more significant reduction in perplexity, reaching a value of 3.57. The results are presented in table 4.5 below.

	German_plays	Brecht_plays
Perplexity	10.19	3.57

Table 4.5

Perplexity.

Next, we focus on the BLEU score. It is important to emphasise that the BLEU score can be significantly affected by parameters such as top-p, top-k, temperature, and the number of candidates generated. To simplify the calculation of the BLEU score, we decide to generate only one candidate for each reference and set top-p to 50. As mentioned before, using the BLEU score alone for interpretation may be limited. Therefore, we use the BLEU score for model comparisons. In our approach, we compare the German base model, which is not yet fine-tuned, with both the German plays model and the Brecht plays model. For this comparison, we use the validation set of the Brecht dataset to evaluate the ability of each model to generate Brecht's plays. We take samples of 100, 300, 500, and 1000 from the dataset. For each sample, we calculate the BLEU score three times and then average the results. The final results are presented in table 4.6.

	German_base	German_plays	Brecht_plays
1st trial (n=100)	0.11	0.46	0.7
2nd trial (n=300)	0.10	0.22	0.65
3rd trial (n=500)	0.10	0.18	0.57
4th trial (n=1000)	0.09	0.17	0.74
BLEU score	0.1	0.26	0.665

Table 4.6

BLEU score.

To illustrate this, we perform three BLEU score trials for each sample size, then average the three scores. We repeat this process for all sample sizes. At the end, we average all the calculated averages to obtain the final BLEU score which is presented in table 4.6 and which clearly shows a significant improvement in the BLEU score.

Finally, to present the results of our models with a user-friendly interface, we develop a web application. To create this web platform, we use several cutting-edge technologies. On the client side, we use HTML, CSS, and JavaScript to design the front end of the application and ensure a visually appealing and interactive user experience. On the server side, we use Flask, a powerful web framework, to host and run the web application, enabling efficient server-side operations. We also integrate a SQLite database to carefully record and store the entire history of user interactions. This feature is important as we are creating scenes similar to those of Bertolt Brecht, so the history of the conversation needs to be preserved for a coherent and meaningful experience.

4.3 Other Improvements

In parallel to the further development of BrAIcht, we are turning our attention to the improvement of *MoliAIre*, a theatrical bot developed for the creation of plays influenced by the famous French playwright Molière. This initiative is an evolution of a previous project presented by Guillaume Grosjean, Tristan Cazenave, and Baptiste Rozière (Guillaume Grosjean (2022)). To achieve our goal, following the approach taken in the development of BrAIcht, we choose a model adapted to the French dataset, as highlighted in Müller and Laurent (2022). This model is currently the predominant language model developed for the French language and guarantees a considerable degree of competitiveness and applicability for our project. Cedille, which is built on the GPT-J framework with about 6B parameters, serves as the architectural basis for the model. Cedille is trained on 78B tokens of French text from the C4 dataset.[3] The configuration of the model is shown in table 4.7.

Similar to the development of BrAIcht, we use two different datasets for fine-tuning the LLM. The first dataset consists of 539 French plays from the 17th century that have a similar style to Molière. The second dataset consists of 32 plays written specifically by Molière. The cues are extracted from both datasets, and we apply consistent preprocessing and encoding techniques as in BrAIcht (see section 3.3). In a two-stage fine-tuning approach similar to BrAIcht, the model is fine-tuned in the first training session using the French pieces, which are divided into an 80%-20% train-test ratio. In the second training session, the resulting model is then fine-tuned again using Molière's

[3] `https://huggingface.co/Cedille/fr-boris`

plays in a ratio of 90%-10%. Due to hardware limitations, the QLoRA technique is again used for fine-tuning (see section 3.4).

	Number of cues	**Training set**	**Validation set**
French plays	190,351	152,280	38,071
Molière plays	15,197	13,677	1,520

Table 4.7

MoliAIre datasets.

The details concerning BitsAndBytes, LoRA, and training parameters for both the 1st and 2nd training sessions can be found in Appendix A.

CHAPTER 5
CONCLUSION AND RECOMMENDATIONS

5.1 Summary of the Findings

The aim of this research project is to develop a CA capable of writing plays in the distinctive style of the famous playwright Bertolt Brecht by utilising the capabilities of LLMs. Among the investigated LLMs, the German LLaMA2 model proves to be the most suitable for tackling German language tasks. Our approach for fine-tuning this model follows a two-step process.

First, we fine-tune the model using an extensive dataset consisting of 907 German plays. Moreover, in the second phase, we re-train the resulting model using 29 plays written exclusively by Bertolt Brecht. The training process is carried out on the LAMSADE server, using an A6000 graphics processor with 48GB RAM to ensure efficient and effective model training. The results obtained are promising and show that the model is capable of generating plays that reflect Bertolt Brecht's distinctive style. The results are also encouraging as the loss drops to an impressive level. Our model achieves a BLEU score of around 0.67 and a perplexity of 3.57, both of which serve as indicators of its promising performance.

Furthermore, to make our CA accessible, we deploy it on LAMSADE server with a user-friendly web application. We also set up a database system to store and manage users' dialogues with our CA for a seamless and engaging user experience. Thus, this project is a successful fusion of cutting-edge language modelling, fine-tuning, and user-friendly application design with the goal of bringing Bertolt Brecht's art into the digital era.

5.2 Limitations of the Research

The aim of this research is to contribute to the field of AI, especially in the field of art and theater. With our work, we aim to offer valuable insights to the AI community actively researching in this field. However, as with any other research, there are some limitations. One of the main limitations we face is related to our available computational resources. Although our GPU is robust, it cannot handle the training of very large models. This limitation significantly affects our choice of model, so we work with models with 7B parameters. While these models provide impressive results, we have to admit that using even larger models could have led to even more remarkable results. The number of parameters in a model is often linked to its performance, and our hardware limitations required a compromise in this aspect.

Another notable limitation involves the availability of the dataset. In particular, we have limited access to a dataset of plays written by Bertolt Brecht. Therefore, we collect data from other German playwrights who show stylistic similarities with Bertolt Brecht. While this strategy allows us to create a more diverse dataset for fine-tuning, it is important to recognize that the lack of an extensive dataset directly linked to Bertolt Brecht's works could potentially compromise the model's ability to fully capture and replicate his distinctive style. Despite these limitations, our study represents a significant advance in the field of conversational agents capable of mimicking the unique creative style of famous playwrights.

5.3 Suggestions for Future Research

The results of this research and its implications for AI researchers are of great importance and therefore deserve to be investigated in further studies. One of the possible improvements of this research could be the exploration of extending the scale of our model. In order to reduce the current computational limitations, it is advisable to gain access to more powerful hardware resources or cloud-based GPU facilities. Such an approach would allow us to work with larger language models, potentially leading to a more remarkable performance in mimicking Bertolt Brecht's distinctive artistic style. Second, we propose to expand our dataset to include a more comprehensive collection of Bertolt Brecht's works. A specialized and comprehensive dataset would improve the model's ability to accurately capture Brecht's unique style. In this context, we can also utilize data augmentation techniques to enrich our dataset. Another improvement could be the inclusion of human evaluation to assess the quality of the plays produced by our CA. Integrating comprehensive human evaluations into future research projects will facilitate the fine-tuning of the model and provide deeper insights into its performance from a human-centered perspective, ensuring the fidelity and artistic value of the generated content. Finally, one interesting improvement idea involves using an encoder such as BERT to convert our German data to English. Subsequently, we can employ another encoder after the GPT to translate the English output back to German. In this scenario, the base LLaMA2 model, which excels with English data, can be utilized. This might be a beneficial approach, particularly since there isn't a perfect model for German tasks currently. However, it's a bit complex and could require a considerable amount of time to set up, along with significant computing power. So, while it's an exciting idea, careful planning and ensuring we have resources would be essential before implementing it.

REFERENCES

Adiwardana, D., Luong, M.-T., So, D. R., Hall, J., Fiedel, N., Thoppilan, R., ... others (2020). Towards a human-like open-domain chatbot. *arXiv preprint arXiv:2001.09977*.

Bahdanau, D., Cho, K., & Bengio, Y. (2016). *Neural machine translation by jointly learning to align and translate.*

Bangura, M., Barabashova, K., Karnysheva, A., Semczuk, S., & Wang, Y. (2023). *Automatic generation of german drama texts using fine tuned gpt-2 models.*

Bengio, Y., Ducharme, R., & Vincent, P. (2000). A neural probabilistic language model. In T. Leen, T. Dietterich, & V. Tresp (Eds.), *Advances in neural information processing systems* (Vol. 13). MIT Press. Retrieved from `https://proceedings.neurips.cc/paper_files/paper/2000/file/728f206c2a01bf572b5940d7d9a8fa4c-Paper.pdf`

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., ... Amodei, D. (2020). Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, & H. Lin (Eds.), *Advances in neural information processing systems* (Vol. 33, pp. 1877–1901). Curran Associates, Inc. Retrieved from `https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf`

Chang, C., Peng, W.-C., & Chen, T.-F. (2023). *Llm4ts: Two-stage fine-tuning for time-series forecasting with pre-trained llms.*

Dai, Z., Yang, Z., Yang, Y., Carbonell, J., Le, Q. V., & Salakhutdinov, R. (2019). Transformer-xl: Attentive language models beyond a fixed-length context. *arXiv preprint arXiv:1901.02860*.

Dettmers, T., Lewis, M., Belkada, Y., & Zettlemoyer, L. (2022). *Llm.int8(): 8-bit matrix multiplication for transformers at scale.*

Dettmers, T., Pagnoni, A., Holtzman, A., & Zettlemoyer, L. (2023). *Qlora: Efficient finetuning of quantized llms.*

Frantar, E., & Alistarh, D. (2022). Optimal brain compression: A framework for accurate post-training quantization and pruning. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, & A. Oh (Eds.), *Advances in neural information processing systems* (Vol. 35, pp. 4475–4488). Curran Associates, Inc. Retrieved from `https://proceedings.neurips.cc/paper_files/paper/2022/`

file/1caf09c9f4e6b0150b06a07e77f2710c-Paper-Conference.pdf

Frantar, E., Ashkboos, S., Hoefler, T., & Alistarh, D. (2023). *Gptq: Accurate post-training quantization for generative pre-trained transformers.*

Graves, A. (2013). Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850*.

Guillaume Grosjean, B. R., Tristan Cazenave. (2022). *Moliaire, a theatrical agent which speaks like molière's characters.*

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, *9*(8), 1735-1780. doi: 10.1162/neco.1997.9.8.1735

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., … Chen, W. (2021). *Lora: Low-rank adaptation of large language models.*

Jiang, A. Q., Sablayrolles, A., Mensch, A., Bamford, C., Chaplot, D. S., de las Casas, D., … Sayed, W. E. (2023). *Mistral 7b.*

Kingma, D. P., & Ba, J. (2017). *Adam: A method for stochastic optimization.*

Li, Y., Li, Z., Zhang, K., Dan, R., Jiang, S., & Zhang, Y. (2023). *Chatdoctor: A medical chat model fine-tuned on a large language model meta-ai (llama) using medical domain knowledge.*

Lo, K.-L., Ariss, R., & Kurz, P. (2022). Gpoet-2: A gpt-2 based poem generator. *arXiv preprint arXiv:2205.08847*.

Müller, M., & Laurent, F. (2022). *Cedille: A large autoregressive french language model.*

Nguyen, T. T., Wilson, C., & Dalins, J. (2023). *Fine-tuning llama 2 large language models for detecting online sexual predatory chats and abusive texts.*

Papineni, K., Roukos, S., Ward, T., & Zhu, W.-J. (2002). Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the association for computational linguistics* (pp. 311–318).

Radford, A., Narasimhan, K., Salimans, T., Sutskever, I., et al. (2018). Improving language understanding by generative pre-training.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al. (2019). Language models are unsupervised multitask learners. *OpenAI blog*, *1*(8), 9.

Roller, S., Dinan, E., Goyal, N., Ju, D., Williamson, M., Liu, Y., … others (2020). Recipes for building an open-domain chatbot. *arXiv preprint arXiv:2004.13637*.

Rosa, R., Dušek, O., Kocmi, T., Mareček, D., Musil, T., Schmidtová, P., … Vosecká, K.

(2020). *Theaitre: Artificial intelligence to write a theatre play.*

Shoham, O. B., & Rappoport, N. (2023). *Cpllm: Clinical prediction with large language models.*

Thoppilan, R., De Freitas, D., Hall, J., Shazeer, N., Kulshreshtha, A., Cheng, H.-T., … others (2022). Lamda: Language models for dialog applications. *arXiv preprint arXiv:2201.08239*.

Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., … Lample, G. (2023). *Llama: Open and efficient foundation language models.*

Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., … Scialom, T. (2023). *Llama 2: Open foundation and fine-tuned chat models.*

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., … Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, *30*.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., … Rush, A. (2020, October). Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 conference on empirical methods in natural language processing: System demonstrations* (pp. 38–45). Online: Association for Computational Linguistics. Retrieved from `https://aclanthology.org/2020.emnlp-demos.6` doi: 10.18653/v1/2020.emnlp-demos.6

Yang, Y., Tang, Y., & Tam, K. Y. (2023). *Investlm: A large language model for investment using financial domain instruction tuning.*

Yu, X., Chen, Z., Ling, Y., Dong, S., Liu, Z., & Lu, Y. (2023). *Temporal data meets llm – explainable financial time series forecasting.*

Zhang, Y., Sun, S., Galley, M., Chen, Y.-C., Brockett, C., Gao, X., … Dolan, B. (2019). Dialogpt: Large-scale generative pre-training for conversational response generation. *arXiv preprint arXiv:1911.00536*.

Zheng, O., Abdel-Aty, M., Wang, D., Wang, C., & Ding, S. (2023). *Trafficsafetygpt: Tuning a pre-trained large language model to a domain-specific expert in transportation safety.*

Zhou, P., Xie, X., & YAN, S. (2023). *Towards understanding convergence and generalization of adamw.* Retrieved from `https://openreview.net/forum?id=EfTN2tSGlF`

Zhuang, Z., Liu, M., Cutkosky, A., & Orabona, F. (2022). *Understanding adamw*

through proximal methods and scale-freeness.

APPENDIX A

BitsAndBytes, LoRA, and training parameters for Molière

Parameter	Value
Load_in_4bit	True
Bnb_4bit_quant_type	"nf4"
Bnb_4bit_compute_dtype	"float16"
Use_nested_quant	False
LoRA_r	16
LoRA_alpha	16
LoRA_dropout	0.3

Table A.1 BitsAndBytes and LoRA parameters for MoliAIre.

Parameter	Value
Epochs	4
Weight_decay	0.01
Train_batch_size	1
Validation_batch_size	1
Optimizer	"paged_adamW_32bit"
Learning_rate	1e-4
Warmup_ratio	0.03
Maximum_gradient_norm	0.3
Maximum_sequence_length	1024

Table A.2 1st-stage training parameters for MoliAIre.

Parameter	Value
Epochs	4
Weight_decay	0.01
Train_batch_size	1
Validation_batch_size	1
Optimizer	"paged_adamW_32bit"
Learning_rate	1e-4
Warmup_ratio	0.03
Maximum_gradient_norm	0.3
Maximum_sequence_length	1024

Table A.3 2nd**-stage training parameters for MolAIre.**

APPENDIX B

Examples of dialogues evaluated with BrAIcht

User: Sind Sie bereit ?

BrAIcht: Ja.

User: Perfekt. Sollen wir einen Spaziergang machen?

BrAIcht: Nein, ich will hier bleiben.

User: Warum? Du gehst nicht gern mit mir aus?

BrAIcht: Ich habe auch nicht gern allein zu Haus.

User: Soll ich also zu Ihnen nach Hause kommen?

BrAIcht: Ja, das möchte ich gern.

User: Okay, ich treffe dich um 20:00 Uhr bei dir zu Hause. Ist das gut für Sie?

BrAIcht: Ja.

User: Ok wir sehen uns dann.

BrAIcht: Du wirst mir nicht entkommen.

Table B.1 A conversation example of BrAIcht.

User: Are you ready ?

BrAIcht: Yes.

User: Perfect. Shall we take a walk?

BrAIcht: No, I want to stay here.

User: Why? Don't you like going out with me?

BrAIcht: I don't like being home alone either.

User: So should I come to your house?

BrAIcht: Yes, I would like that.

User: Okay, I'll meet you at your house at 8:00 p.m. Is that good for you?

BrAIcht: Yes.

User: Ok see you then.

BrAIcht: You won't escape me.

Table B.2 Translation in English of table B.1.

User: Hey ! Freut mich, Sie kennenzulernen !
BrAIcht: Nun, ich bin auch neugierig, Sie näher kennen zu lernen.
User: Ich bin ein Ingenieur für künstliche Intelligenz und arbeite in einem Forschungslabor. Was ist mit dir?
BrAIcht: Ich bin ein Mann von Welt und reise viel.
User: Sehr schön zu hören. In welche Länder reisen Sie gerne?
BrAIcht: In alle.

Table B.3 An example of BrAIcht conversation.

User: Hey! Nice to meet you !
BrAIcht: Well, I'm also curious to get to know you better.
User: I am an artificial intelligence engineer working in a research lab. What about you?
BrAIcht: I am a man of the world and travel a lot.
User: Very nice to hear. Which countries do you like to travel to?
BrAIcht: In all.

Table B.4 Translation in English of table B.2.