BERT Methods for German Language Sentiment Classification

Master's Thesis

Submitted by:	**Sana Ali Warsi**
Address:	**Universitätsring 8b, zimmer:210,Trier**
Email:	**s4sawars@uni-trier.de**
Matriculation Number:	**1511452**

Master's program
Data Science

Supervisors :

Prof. Dr. Jan Pablo Burgard

M.Sc. Christopher Caratiola

Trier, 7th January 2024

ERKLÄRUNG ZUR BACHELORARBEIT / MASTERARBEIT

Hiermit erkläre ich, dass ich die Bachelorarbeit / Masterarbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt und die aus fremden Quellen direkt oder indirekt übernommenen Gedanken als solche kenntlich gemacht habe.
Die Arbeit habe ich bisher keinem anderen Prüfungsamt in gleicher oder vergleichbarer Form vorgelegt. Sie wurde bisher nicht veröffentlicht.

Datum: 7th January 2024 Unterschrift:

Abstract

In the contemporary landscape of sentiment analysis, the Bidirectional Encoder Representations from Transformers (BERT) and Cross-lingual Language Model - Robustly Optimized BERT (XLM-RoBERTa) models have emerged as powerful tools for understanding and classifying sentiments in text. This thesis delves into their test and application in the context of German language sentiment classification. Testing has been carried out for four distinct labeled datasets and application to an unlabeled dataset related to nuclear energy has been done. In coping with noisy input, specifically spelling mistakes by swapping the characters in the word using nlpaug library, a comprehensive comparative analysis is conducted, pitting BERT against XLM-RoBERTa, across four labeled datasets. Notably, the datasets, characterized by varying text lengths, exhibit differing sensitivities to spelling errors. XLM-RoBERTa demonstrates superior performance in two datasets, showcasing its adaptability to varied linguistic differences. Conversely, BERT outperforms XLM-RoBERTa in one dataset, emphasizing the significance of considering specific dataset characteristics.
Expanding the analysis to unlabeled dataset related to nuclear energy, XLM-RoBERTa predominantly predicts neutral sentiments, whereas BERT includes negative sentiments also, though with almost negligible percentage of positive tweets in both models.

Contents

List of Abbreviations

S.no	Acronym	Full Forms
1	AI	Artificial Intelligence
2	ML	Machine Learning
3	NB	Naive Bayes
4	SVM	Support Vector Machine
5	LR	Logistic Regression
6	DT	Decision Tree
7	LSTM	Long Short-Term Memory
8	MLM	Masked Language Model
9	NSP	Next Sentence Prediction
10	LM	Language Model
11	GRUs	Gated Recurrent Units
12	CNN	Convolutional Neural Networks
13	NLP	Natural Language Processing
14	GPT	Generative Pre-trained Transformer
15	ELMo	Embeddings from Language Models
16	sota	state-of-the-art
17	BERT	Bidirectional Encoder Representations from Transformers
18	ABSA	Aspect-Based Sentiment Analysis
19	ASC	Aspect-Based Sentiment Classification
20	MSA	Multimodal Sentiment Analysis
21	NER	Named Entity Recognition
22	TL	Transfer Learning
23	RoBERTa	Robustly Optimized BERT
24	XLM-R	Cross-lingual Language Model - RoBERTa
25	mBERT	multilingual BERT
26	NLI	Natural Language Inference
27	RNN	Recurrent Neural Network
28	BiLSTM	Bidirectional Long Short-Term Memory
29	BiGRU	Bidirectional Gated Recurrent Unit
30	DistilBERT	Distilled BERT
30	PTMs	Pre-trained Models

List of Figures

Chapter 1

Introduction

"The rapid proliferation of internet-dependent applications, such as social media platforms and blogs, has led to an influx of commentary and assessments pertaining to everyday activities. Sentiment analysis is the systematic process of extracting and assessing individuals' opinions, beliefs, and perceptions on a wide array of topics, products, subjects, and services. The viewpoints of individuals can serve as a valuable resource for entities such as organizations, governments, and individuals, enabling them to gather insights and base their decisions on these perspectives. Sentiment analysis employs techniques such as natural language processing and text mining to identify and extract subjective data from textual content" (Wankhade, A. Rao, and Kulkarni (2022)).
"Traditional machine learning approaches such as Naive Bayes, Decision Trees, and Support Vector Machines are used in sentiment analysis algorithms, as are deep learning models such as Recurrent Neural Networks. When analysing long sequences of data, such as text, RNNs suffer from vanishing gradients difficulties" (Cheruku et al. 2023). They also fail to gather data with long-term dependencies. The LSTM and GRU models were proposed to address these concerns. "Both models include gating techniques that allow the network to selectively recall or forget information from previous inputs, allowing the model to better reflect long-term interdependence. The Transformer model, which uses the attention mechanism, has recently demonstrated remarkable performance in natural language processing. The attention mechanism enables the model to weigh different parts of the input sequence selectively in order to generate informative embeddings" (Tan, Lee, and Lim 2023).

Sentiment analysis, a vital component of natural language processing, plays a pivotal role in discerning and understanding the emotional tone conveyed within textual data. As the demand for sentiment classification models intensifies, particularly in multilingual settings, the utilization of advanced language models becomes imperative. This thesis delves into the realm of sentiment classification for the German language, employing the BERT (Bidirectional Encoder Representations from Transformers) methodology and its variant, XLM-RoBERTa (Cross-lingual Language Model - RoBERTa).

The research hinges on the evaluation of these models across four diverse labeled datasets. To enrich the analysis, a controlled introduction of spelling errors was incorporated into all four datasets using the ***nlpaug*** Python library. This augmentation aimed to simulate real-world scenarios where text imperfections are inherent. The datasets vary not only in their labeled sentiments but also in the length of the texts they encompass. In addition to the labeled datasets, an unlabeled dataset

focusing on nuclear energy was introduced, providing a practical application of the models. BERT and XLM-RoBERTa were employed to predict sentiments within this unannotated dataset, revealing distinctive patterns.

Natural Language Processing (NLP) constitutes a significantly explored field of research, underpinning a diverse range of applications. These include, but are not limited to, translation systems, voice-enabled assistants, and conversational interfaces such as chatbots. By using BERT and XLM-RoBERTa models for sentiment analysis, we can examine for given time frames in section 4.1, how public sentiment around nuclear energy has changed around these time frames.

This thesis unfolds a comprehensive exploration of sentiment classification in the German language, unraveling the intricacies of model performance under the influence of spelling errors and diverse dataset characteristics. The findings presented herein contribute to the advancement of sentiment analysis methodologies, offering valuable insights into the application of BERT and XLM-RoBERTa in the context of German language sentiment classification.

Confronted with the requirement to classify German language texts, we are going to investigate the following research questions in the Master's thesis:

1. Which model (BERT or XLM-RoBERTa) is performing better for sentiment analysis in German language?

2. Which of these models is more sensitive to text sizes?

3. Which of these models is more sensitive to spelling error?

4. For practical application can we identify strong differences between the output of these models.

The rest of the thesis is structured as follows.

Chapter 2 provides relevant literature review for theoretical background and development in the field of Sentiment analysis , including current studies and research. In chapter 3 we give, ideas on different methodology used for sentiment analysis, different types of neural networks and transformer models. It describes all aspects of the BERT model, and gives an overview of more recent model architectures based on BERT, it also discuss other alternative models for German language sentiment analysis.
Chapter 4 discusses different data sources and steps involved in data preparation. In this chapter we test BERT and XLM-RoBERTa models for German sentiment analysis also we talk about different metrices for performance assessment and selecting best BERT model for German sentiment analysis. This chapter also provides detailed application for unlabelled data discussing the result of experiment.
In Chapter 5 finally, the thesis ends with a short conclusion and provides some basic insights on future work.

Chapter 2

Literature Review

2.1 Background and Development in Sentiment Analysis

Definition of sentiment analysis: "Sentiment analysis, often known as opinion mining, is a distinct topic within the field of Natural Language Processing (NLP). The main purpose of this system is to recognize and extract subjective content from sources of textual data. Sentiment analysis aims to ascertain the overarching emotional disposition or sentiment of a textual piece, such as a review, tweet, or news article. The practice of obtaining and analyzing people's ideas, thoughts, and perceptions about various themes, products, subjects, and services is known as sentiment analysis. People's opinions may help organizations, governments, and people gather information and make decisions based on their opinions" (Wankhade, A. Rao, and Kulkarni (2022)).

Sentiment analysis levels: According to Joshi and Itkat 2014, sentiment analysis strategies have been categorized in different dimensions in their meta analysis of methods related to sentiment analysis. In this scheme, they present three different levels of sentiment analysis.

1. Document-level sentiment analysis

2. Sentence-level sentiment analysis

3. Aspect-level sentiment analysis

The main job at the document level is to determine whether an entire opinion piece exhibits a favourable or unfavourable attitude. This level of analysis is based on the assumption that each document represents ideas about a particular entity. At the sentence level, the primary task is to ascertain whether each sentence conveys a positive, negative, or neutral sentiment. This level of scrutiny is intimately connected with subjectivity classification, which distinguishes between objective sentences that express factual information and subjective sentences that convey personal thoughts and opinions. Studies conducted at the document and phrase levels, however, do not provide insights into the specific aspects that individuals found appealing or unappealing. Aspect level analysis is more fine-grained. Aspect level examines the opinion itself rather than language constructs (documents,

sentences, paragraphs, clauses, or phrases) (Devika, Sunitha, and Ganesh 2016).

The choice of sentiment analysis type depends on the specific requirements of the task. Both BERT and XLM-RoBERTa, with their contextualized embeddings and pre-training on diverse data, are well-suited for various sentiment analysis tasks, including document-level, sentence-level, aspect-based and multilingual sentiment analysis.

Applications of sentiment analysis: According to Wankhade, A. Rao, and Kulkarni (2022), sentiment analysis has various applications, including evaluating customer opinions and patient mental health status based on social media posts. Furthermore, technology advancements such as Blockchain, IoT, Cloud Computing, and Big Data have expanded the spectrum of sentiment analysis applications, allowing it to be applied in virtually any discipline. Figure 2.1 depicts a few of the most commonly utilized applications in sentiment analysis. The following are some of the most important domains and sectors where sentiment analysis is used.

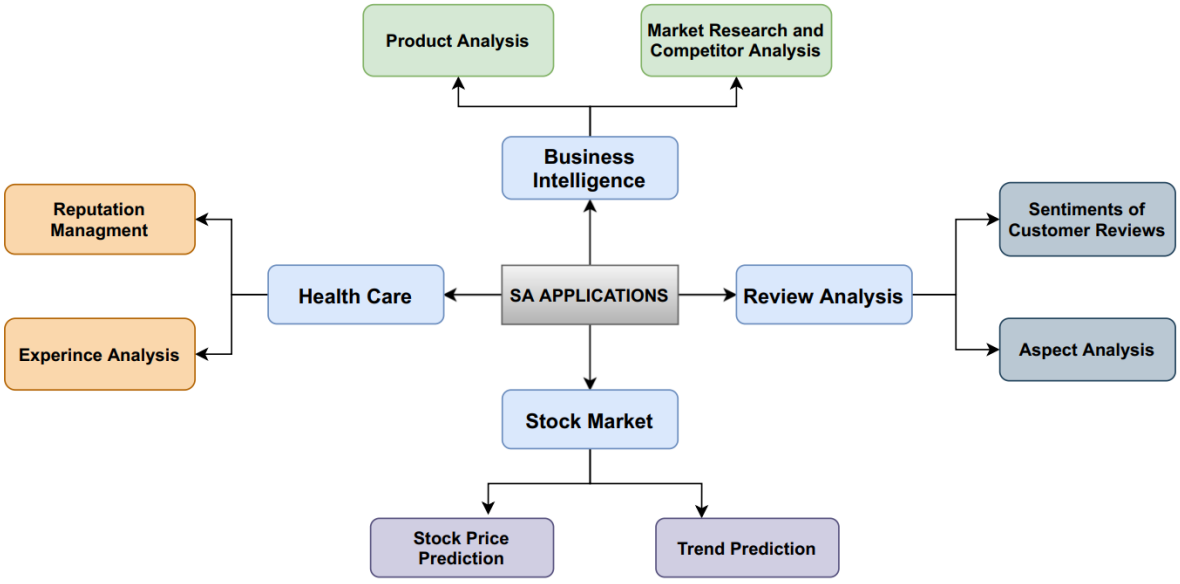

Figure 2.1: Applications of sentiment analysis by Wankhade, A. Rao, and Kulkarni (2022) (p: 5762)

Challenges in sentiment analysis: The following are a few important issues in sentiment analysis in accordance to Mohammad 2017, which are being faced by almost every researcher doing sentiment analysis:

1. **Challenges in Annotating for Sentiment:** For high-quality annotations, clear and concise instructions are essential. This is true even for relatively basic annotation jobs like sentiment annotation, in which instances are labeled as positive, negative, or neutral. Some sentence kinds that are particularly difficult to annotate with sentiment (employing either the conventional sentiment questionnaire or the questionnaire based on semantic roles for sentiment analysis.) are as follows: *Sarcasm and ridicule*, *Rhetorical questions*, *Quoting somebody else or re-tweeting*, *Speaker's emotional state* etc.

2. **Challenges in Multilingual Sentiment Analysis:** Addressing emotions in German requires language-specific nuances; thus, multilingualism is essential. A comparison analysis was done to examine the accuracy of a sentiment analysis system that utilized original English texts vs a system trained on machine translated texts in languages such as German, Spanish, and French. The investigators concluded that the machine translation quality is enough for doing sentiment analysis on mechanically translated texts. However, certain aspects of multilingual sentiment analysis remain underexplored. These include strategies for preserving the degree of sentiment during text translation, understanding the functional differences of sentiment modifiers such as negators and modals across languages, and comprehending how automatic translations vary from manual translations in terms of sentiment.

3. **Negated Expressions:** Negation, often articulated through negative indicators or negator terms such as 'not' or 'never', can significantly influence the sentiment within its scope. A deeper comprehension of negation's impact on sentiment is crucial for enhancing the efficacy of robotic sentiment analysis. However, numerous facets of negation remain unexplored, warranting further investigation. For instance, it is unclear whether negators can be hierarchically arranged based on their average influence on the sentiment of their scopes. This involves discerning which negators exert a more profound impact on sentiment and which ones have a lesser effect. Additionally, the context-dependent nature of these negators is yet to be fully understood. It is essential to identify the circumstances in which a single negator can either amplify or diminish the sentiment of its scope. Moreover, the usage of negations varies across different communities and cultures, a factor that has not been thoroughly examined. Understanding these cultural and communal nuances can provide valuable insights into the diverse applications of negation. Lastly, the usage of negations in sentiment expressions is another area that requires further exploration. This could involve studying how negations can alter the sentiment conveyed by certain expressions, thereby adding a layer of complexity to sentiment analysis.

4. **Emojis:** Emojis have become a part of everyday life and are more efficient than words at expressing one's emotions. However, because sentiment analysis methods rely on textual words, emojis cannot be reliably recognised and are consequently excluded from many assessments. As a result, one is left with a partial analysis (Yılmaz 2023).

Some More Details on Sentiment Analysis: In the following section all the concepts are discussed in accordance to Yuan et al. 2020:

Two decades ago, the inception of sentiment analysis was marked by the simple task of determining whether a word carried a positive or negative connotation. This rudimentary assignment laid the foundation for an extensive and rigorous study in sentiment analysis. The field has since seen a proliferation of research studies and practical applications. This section primarily concentrates on fundamental undertakings such as sentiment classification and aspect-based sentiment analysis. Additionally, it explores some emerging areas of interest including implicit sentiment analysis, multimodal sentiment analysis, and sentiment generation, as depicted in the figure 2.2.

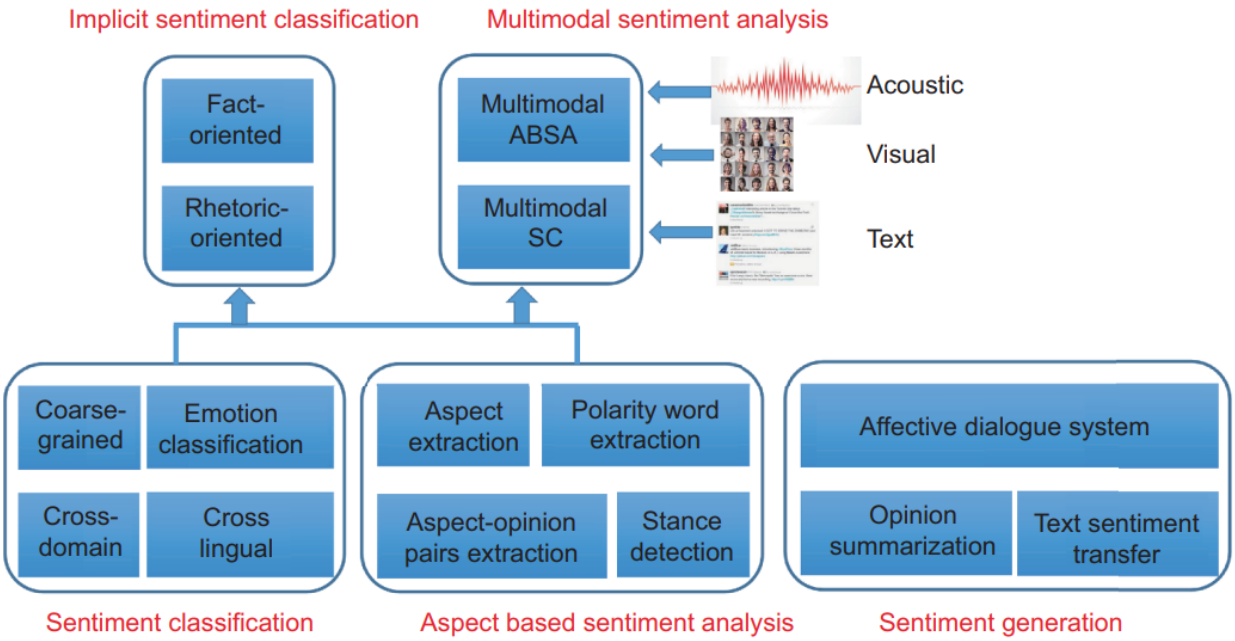

Figure 2.2: The framework of sentiment analysis by Yuan et al. 2020 (p: 1948)

Sentiment classification: Sentiment classification, a foundational issue in sentiment analysis, is primarily concerned with the categorization of the sentiment orientation inherent in a given textual sequence. Such a sequence could range from a solitary word to a complete sentence, or even extend to an entire document. Because of this variation in text length, the complexity of semantic composition varies. Furthermore, the output label might be a sentiment label (positive, negative, or neutral), a star rating, or an emotion label (happy, sad, furious, surprised, etc.).

Sentiment classification with domain transfer: Owing to the divergence across domains, sentiment classifiers, once trained in a specific domain, may not exhibit optimal performance when applied to new domains. This necessitates the development of more advanced algorithms to bridge such feature disparities. Two well-researched tasks that address this challenge are cross-domain and multi-domain sentiment categorization. The former concentrates on the learning of a transferable feature extractor, with little or no reliance on target domain data. In contrast, multi-domain sentiment classification endeavors to construct a model trained on data sourced from multiple domains, with the objective of achieving superior average performance across all domains.

Sentiment classification with language transfer: Given that deep learning methodologies are significantly reliant on annotated data, and considering the variation in sentiment resources across languages, it becomes imperative to adapt sentiment resources from resource-rich languages to those languages that are relatively resource-poor in comparison to English. Consequently, the task of cross-lingual sentiment classification seeks to address the challenge of constructing an effective sentiment classifier for the target language, even in the absence of labeled data in that particular language. In a similar vein, multi-lingual sentiment classification is concerned with the development of a sentiment classifier that exhibits robust performance, on average, across all languages of interest.

Emotion classification: The task of emotion classification is conceived with the aim of reflecting the diverse emotional states of individuals as manifested on social media platforms. This task presents a multi-label classification problem, wherein a classifier is required to assign multiple labels, such as "happy", "sad", "anger", "disgust", "surprise", or "fear", to a given text fragment. Given that a single sentence may encompass one or more emotions, the task of multi-label emotion classification is typically approached either as a series of binary classification problems or by selecting labels that exceed a predetermined probability threshold.

2.2 Current Studies and Research

In recent years, researchers have been developing different models as a variant of BERT and XLM-RoBERTa models to improve sentiment analysis.

"Recent years have witnessed the remarkable success and subsequent milestone status achieved by large-scale pre-trained models (PTMs) such as BERT and GPT in the realm of artificial intelligence (AI). These large-scale PTMs have demonstrated their ability to effectively extract information from substantial volumes of both labeled and unlabeled data, courtesy of their advanced pre-training objectives and extensive model parameters. Following the advent of GPT and BERT, several enhancements have been proposed, including but not limited to RoBERTa and ALBERT, with RoBERTa representing a successful adaptation of BERT. Beyond RoBERTa and ALBERT, a plethora of PTMs have been proposed in recent years with the objective of improving knowledge extraction from unlabeled data. Certain research efforts have focused on enhancing model architectures and exploring innovative pre-training tasks, as exemplified by models such as XLNet, SpanBERT, ELECTRA, MASS, UniLM, among others" (Han et al. 2021).

As per the research conducted by Kotelnikova et al. 2021, it has been observed that there has been a significant enhancement in the performance of sentiment analysis methodologies in recent years. This improvement can be attributed to the utilization of several models that are based on the Transformer architecture, particularly BERT. However, the training of deep neural network models poses a challenge due to the requirement of vast volumes of data. Additionally, interpreting the results generated by these models presents another hurdle. An alternative to these models are the rule-based (or lexicon-based) approaches, which rely on sentiment lexicons. These approaches are advantageous due to their speed, lack of training requirement, and ease of comprehension. Despite these benefits, lexicon-based approaches have seen a decline in their usage with the rising popularity of deep learning. The study compares RuBERT, a fine-tuned deep neural network model, with SO-CAL and SentiStrength, two lexicon-based methodologies developed for the Russian language. While RuBERT outperforms both lexicon-based methods, SO-CAL surpasses RuBERT in four out of 16 corpora. This observation generally fosters optimism regarding the potential of the lexicon-based approach.

Language models have become a staple in contemporary Natural Language Processing (NLP), with their multilingual capabilities garnering significant attention in

recent times. As per the authors Barbieri, Anke, and Camacho-Collados 2021, their current analysis has been concentrated on the (multilingual variants of) standard benchmarks. In their scholarly article, they introduce XLM-T, a model specifically designed for the training and evaluation of multilingual language models in the context of Twitter.

XLM-T: Multilingual Language Models for Sentiment Analysis in Twitter: Multilingual LMs learn general-purpose multilingual representations while integrating streams of multilingual textual data. A clear indication of this situation is the widespread use of multilingual versions derived from established monolingual language models within the NLP community. As an illustration, mBERT derived from BERT. Nevertheless, the utilization of social media data, namely Twitter, seems to have been curiously disregarded in the widespread practice of extensive multilingual pretraining. This phenomenon may arise from the discursive and platform-specific attributes, including but not limited to out-of-distribution samples, misspellings, slang, vulgarisms, emoji usage, and multimodality. These factors, together with the widely recognized lack of curation, contribute to the observed behavior.

The authors of the paper Barbieri, Anke, and Camacho-Collados 2021 address this gap by creating a comprehensive set of tools for evaluating multilingual language models specifically designed for Twitter. The framework they offer to the NLP community consists of a large multilingual language model specifically designed for Twitter. This language model is developed using XLM-R checkpoints and serves as the foundation for their initial set of baseline findings in different situations. Their proposed XLM-R-Twitter model surpasses robust baseline models like RoBERTa-base and XLM-R, which do not utilize Twitter datasets. It also beats RoBERTa-Twitter, which is exclusively trained on Twitter datasets. This exemplifies the resilience of their multilingual approach in language-specific situations.

A recent study by the author Talaat 2023 has shown that the BERT model has achieved impressive outcomes in the domain of sentiment analysis. Nevertheless, there remains a need for further enhancement in the accuracy of sentiment analysis. Hence, the scholars propose four sophisticated deep learning architectures that integrate BERT with the Bidirectional Long Short-Term Memory (BiLSTM) and Bidirectional Gated Recurrent Unit (BiGRU) algorithms. The research primarily relies on pre-trained word embedding vectors, which are essential for enhancing the model. The study presents techniques that seek to improve precision by integrating layers of BIGRU and BILSTM on both Bert models (DistilBERT, RoBERTa) for text sentiment classification tasks, with and without the presence of emojis.

Chapter 3

Methodology

3.1 Methodology for Sentiment Analysis

The *lexicon based method, machine learning approach,* and *hybrid approach* are the three most often utilized approaches for sentiment analysis. Furthermore, researchers are always looking for improved ways to complete the work with greater precision and at a reduced computing cost. Figure 3.1 depicts an overview of the main methodologies utilized in sentiment analysis in Wankhade, A. Rao, and Kulkarni (2022).

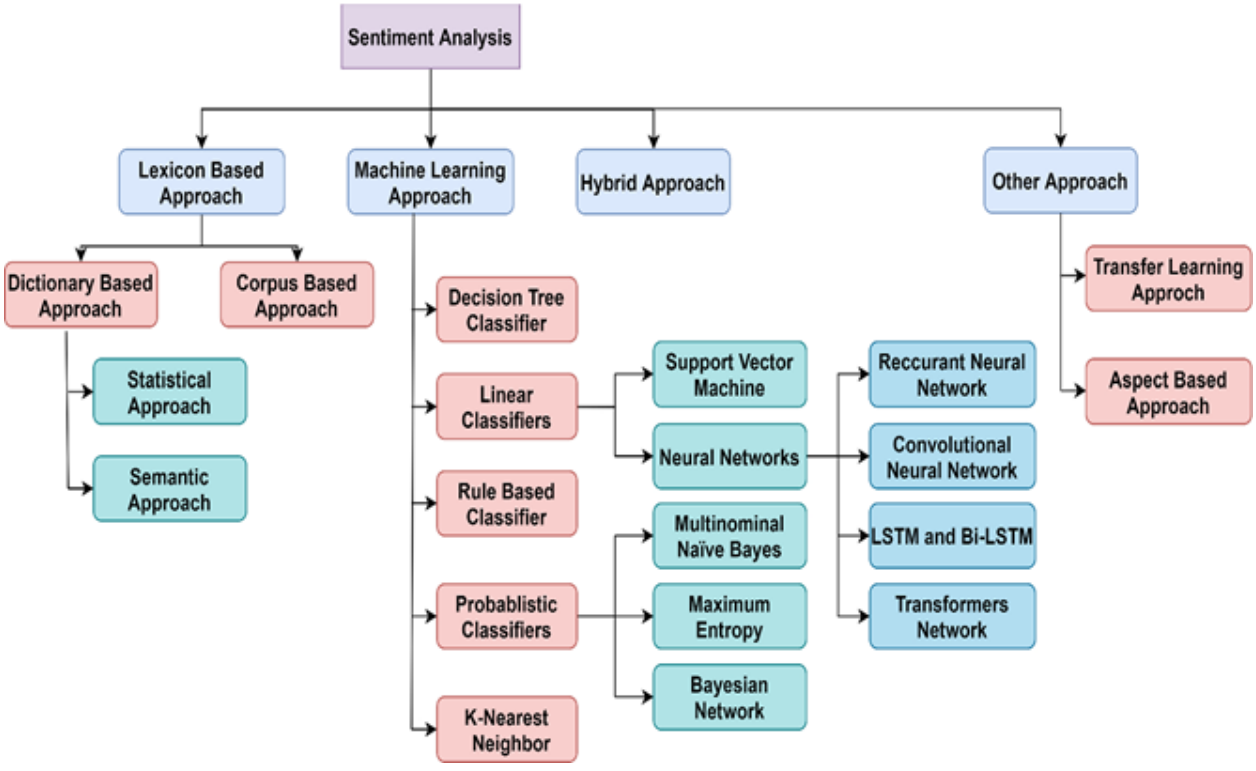

Figure 3.1: Approaches of sentiment analysis by Wankhade, A. Rao, and Kulkarni (2022) (p: 5743)

Lexicon Based Approach: Lexicon-oriented methodologies primarily depend on a sentiment lexicon, which is essentially a compendium of pre-established sentiment terms, phrases, and occasionally idioms, curated for conventional modes of

communication such as the Opinion Finder lexicon. However, more intricate structures, such as ontologies or dictionaries that gauge the semantic orientation of words or phrases, can also be employed for this objective (Serrano-Guerrero et al. 2015). Two sub-categories are discernible in this context: dictionary-based and corpus-based techniques. The former methodology is generally predicated on the utilization of an initial set of phrases, or 'seeds', which are manually collated and annotated. This set proliferates through the exploration of a dictionary's synonyms and antonyms. The inherent limitation of these techniques is their incapacity to adapt to domain and context-specific orientations. Conversely, the objective of corpus-based approaches is to generate dictionaries pertinent to a specific domain. These dictionaries are derived from a set of seed opinion keywords that evolve through the pursuit of associated words using either statistical or semantic methodologies. Statistical methods such as Latent Semantic Analysis (LSA) or the simple frequency of term occurrence within a corpus of documents can be employed (Serrano-Guerrero et al. 2015).

Machine Learning Approach: The sentiment analysis machine learning approach is mostly associated with supervised classification in general and specifically with text classification approaches. Consequently, it is referred to as "supervised learning." For a machine learning-based categorization, you need two sets of documents: a training set and a test set. A diverse range of machine learning methodologies were employed to categorize the evaluations. The topic of text classification has experienced significant progress through the utilization of machine learning techniques, such as Naive Bayes (NB) and Support Vector Machines (SVM) (Vinodhini and Chandrasekaran 2012).

1. **Naive bayes:** The Naive Bayes (NB) classification algorithm is rudimentary yet efficient. The Naive Bayes approach is frequently used for document categorization. The primary concept is utilizing the joint probabilities of words and categories to assess the probabilities of categories based on a test text. The assumption of word independence is the simplistic component of such a paradigm. Due to the straightforward nature of this assumption, the computation of the Naive Bayes classifier is considerably more efficient (Vinodhini and Chandrasekaran 2012).

2. **Support vector machine:** The primary idea behind support vector machine (SVM) is to categorise information separately using hyperplanes in order to maximise the margin between them. SVM is a powerful ML technique for regression, classification, and sentiment detection. The goal of SVM is to find the best classification function for distinguishing between members of the two classes in the training data. SVM is commonly used in the real world to solve a wide range of issues, including intrusion detection, image processing, text categorization, and so on. SVM was initially intended to classify binary classes. The examples were later expanded to include numerous classes (AlBadani, Shi, and Dong 2022).

Hybrid Approach: As articulated in the study by Dang, Moreno-García, and De la Prieta 2021, hybrid methodologies have been shown to be potent models for mitigating sentiment errors in increasingly complex training data. The study utilized

eight separate textual datasets from various domains to develop and assess hybrid deep sentiment analysis models. These models incorporated Long Short-Term Memory (LSTM) networks, Convolutional Neural Networks (CNN), and Support Vector Machines (SVM). These hybrid models were compared against SVM, LSTM, and CNN, which are three independent models. Every approach was assessed according to its dependability and computational time. Across all categories of datasets, the hybrid models surpassed the standalone models in terms of sentiment analysis precision, especially when deep learning models were amalgamated with SVM. The reliability of the latter was significantly superior.

As delineated in the scholarly article by Garrido-Merchan, Gozalo-Brizuela, and Gonzalez-Carvajal 2023, BERT has achieved exceptional outcomes in a wide range of Natural Language Processing (NLP) tasks, sparking a debate about the ongoing effectiveness of traditional NLP methods compared to the superior performance of BERT and similar models. BERT can be seen as an extension of the traditional NLP pipeline, augmenting its flexibility. A contention supporting traditional Machine Learning (ML) NLP methodologies posits that the BERT methodology necessitates voluminous quantities of text to generate precise results. BERT stands as the most proficient deep transfer learning strategy, surpassing the performance of leading traditional ML models.

3.2 Neural Networks for NLP

In the context of NLP, three major classes of neural networks appear predominant. In the following subsection, I will describe recurrent neural networks.

3.2.1 Recurrent Neural Network

Recurrent Neural Networks (RNNs) are unique structures that incorporate temporal data. The computation of outputs at a given time t in an RNN is determined by the hidden state at the same time t, which assimilates information from the input at time t as well as the activations from hidden units at the preceding time t-1, as illustrated in figure 3.2. This endows the RNN with the ability to remember, or the capability to retrieve previous inputs and outputs (Shastri 2020).

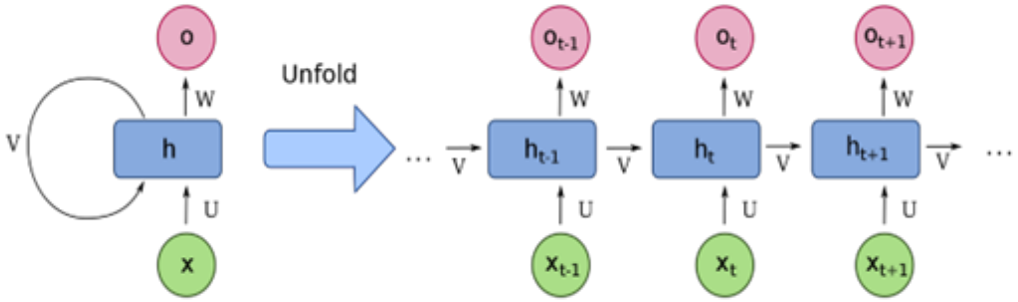

Figure 3.2: Recurrent Neural Network (Shastri 2020)

Contrary to conventional feed-forward neural networks, Recurrent Neural Networks represent a type of network architecture that accommodates fluctuating inputs and mutable outputs. They also possess the capability to assess variable-length inputs, like frames in a video, and execute decisions corresponding to each frame (Feng 2023).

RNN can be constructed as depicted exemplary in figure 3.3.

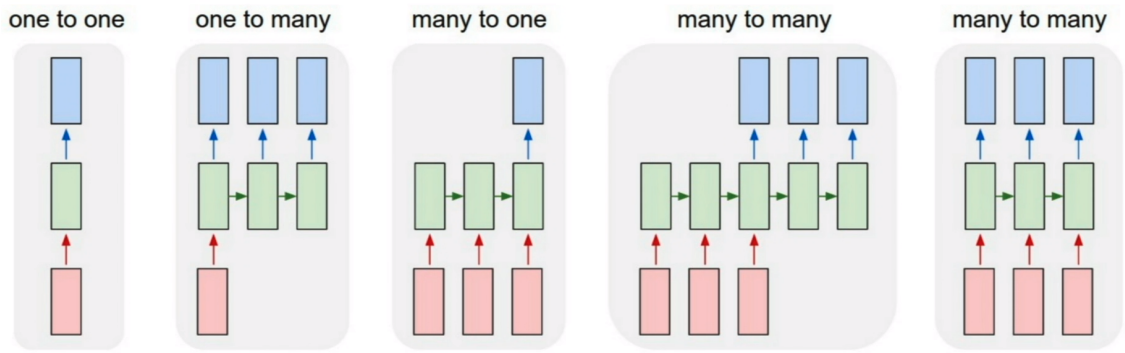

Figure 3.3: The many relations of RNNs (Feng 2023)

One-to-one: The conventional feed-forward neural network architecture is characterized by a unidirectional flow of information, typically from the input layer to the output layer, without any loops. This structure comprises a single input and a single output (Feng 2023).

One-to-many: This paradigm is typically exemplified by the process of image captioning. In this scenario, a single image, which constitutes a fixed-size input, is used to generate variable-length textual descriptions, phrases, or sentences as output. This approach essentially translates the visual content into a linguistic representation, thereby bridging the gap between visual perception and language understanding (Feng 2023).

Many-to-one: This approach is commonly employed in the field of sentiment analysis. The expected input for this process is typically a sequence of words or potentially a paragraph of text. The output generated is typically a regression result, which provides continuous values. These values represent the probability of the expressed sentiment being positive. This method essentially quantifies the emotional tone behind a series of words, used to gain an understanding of the attitudes, opinions, and emotions expressed within the input text (Feng 2023).

Many-to-many: This paradigm finds its application in areas such as machine translation, a notable example being Google Translate. In this context, the input is typically a sentence in English of variable length, and the output is a translation of that sentence into a foreign language, also of variable the "many-to-many" model is a paradigm that finds its application in machine translation tasks, such as those performed by Google Translate. The input in this scenario is typically a sentence in English of variable length, and the corresponding output is a translation of that sentence into a foreign language, also of variable length. This model can also be

extended for tasks such as frame-level video classification. In this application, each frame of a video is fed into the neural network, which then generates an immediate output. However, given the interdependence of video frames, the network needs to transfer its hidden state from one frame to the next. This necessitates the use of recurrent neural networks for such tasks, as they are designed to handle sequential data by maintaining a 'memory' of previous inputs in their hidden layers (Feng 2023).

Limitations of RNN: Recurrent Neural Networks (RNNs) are known to grapple with the issue of Vanishing Gradient Descent. This essentially implies that as the sequence of timestamps lengthens, the RNN tends to lose the information it has acquired from earlier timestamps. This phenomenon, often referred to as 'forgetfulness', significantly hampers the network's ability to learn and retain long-term dependencies (Ghosh 2021).

3.2.2 Long Short Term Memory Network

Long Short Term Memory Networks, often known as "LSTMs," are a specific type of Recurrent Neural Networks (RNNs) that are able to learn long-term dependencies. They are frequently abbreviated as "LSTMs." These networks were first introduced by Hochreiter and Schmidhuber in 1997, and their development and popularization have been the collective effort of many researchers in the years that followed (Oinkina and Hakyll 2015). LSTMs have proven to be highly effective across a broad spectrum of problems and are now widely utilized in the field of machine learning and artificial intelligence.

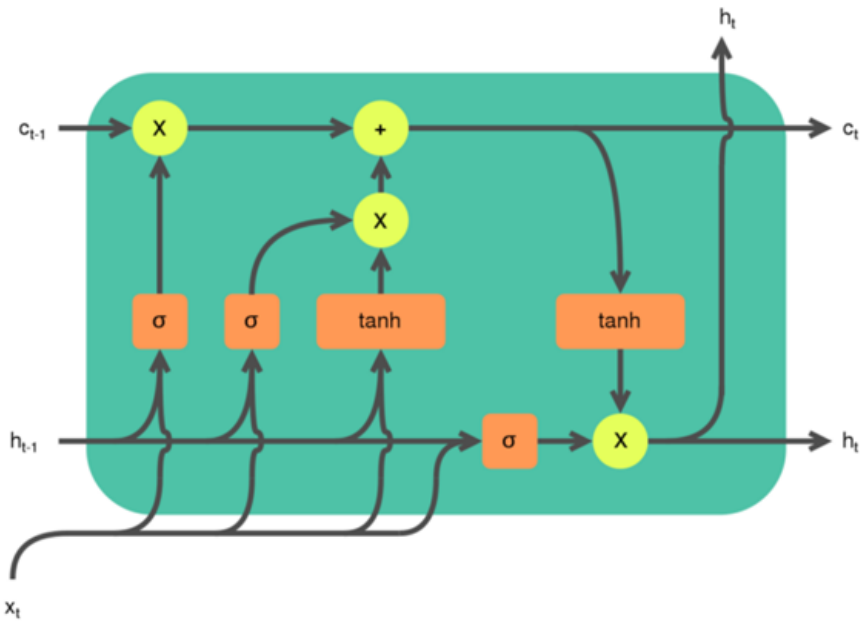

Figure 3.4: LSTM architecture (Shastri 2020)

Figure 3.4 demonstrates that all recurrent neural networks are structured as a sequence of repeated neural network modules. The recurrent module in regular RNNs often consists of a straightforward structure, such as a solitary tanh layer (Oinkina and Hakyll 2015).
tanh: It is a function to overcome the vanishing gradient problem. The second

derivative of the function can maintain a substantial range prior to approaching zero.

Sigmoid: A sigmoid function, capable of producing outputs of either 0 or 1, can serve as a mechanism for either retaining or discarding information.

There are three types of gates in a LSTM network (Dolphin 2020).

As depicted in figure 3.5, the forget gate is responsible for assessing the relevance of specific bits of the cell state, which represents the network's long-term memory, in light of both the preceding hidden state and the present input data.

1.Forget gate: Within the forget gate, the network is conditioned to yield an output approaching 0 when a component of the input is deemed insignificant, and conversely, an output nearing 1 when the component is considered significant. In essence, the forget gate's role is to ascertain which elements of the long-term memory should be de-emphasized or 'forgotten', taking into account both the preceding hidden state and the current data point in the sequence (Dolphin 2020).

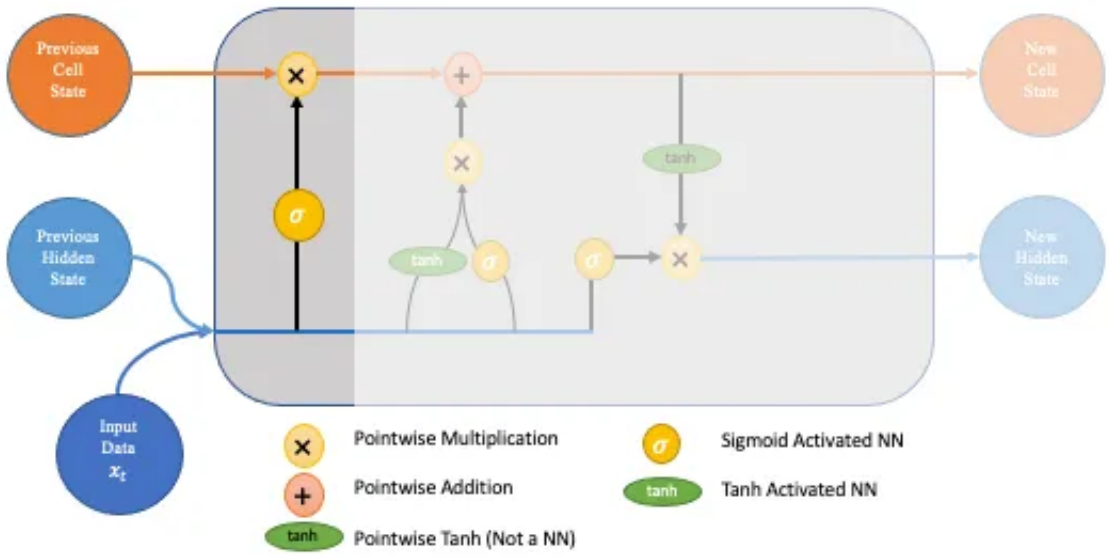

Figure 3.5: Forget Gate (Dolphin 2020)

2.Input gate: As illustrated in figure 3.6, the input gate, activated by a sigmoid function, operates as a filter. It determines which components of the 'new memory vector' should be preserved. The 'new memory vector' is generated by a tanh-activated neural network that has been trained to amalgamate the preceding hidden state and new input data to create a 'new memory update vector'. Given the sigmoid activation, this network will produce a vector of values within the range [0,1], enabling it to act as a filter through pointwise multiplication. An output close to zero, akin to what we observed in the forget gate, signifies that we do not intend to update that portion of the cell state (Dolphin 2020).

3.Output gate:

The step-by-step procedure for the output gate, as depicted in figure 3.7 (Dolphin 2020), is as follows:

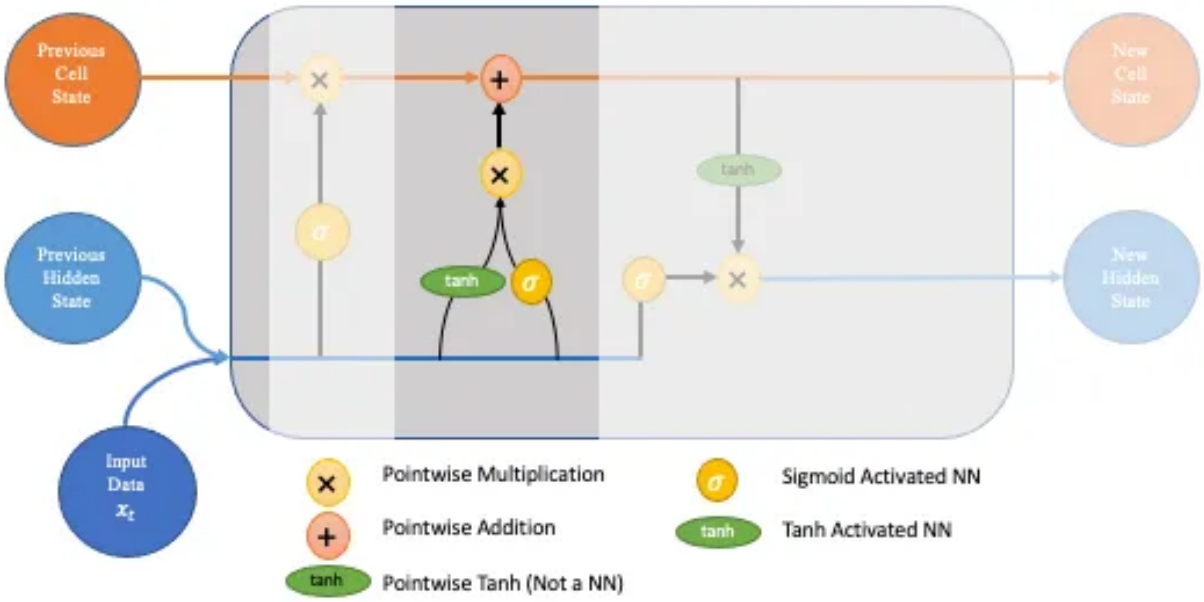

Figure 3.6: Input Gate (Dolphin 2020)

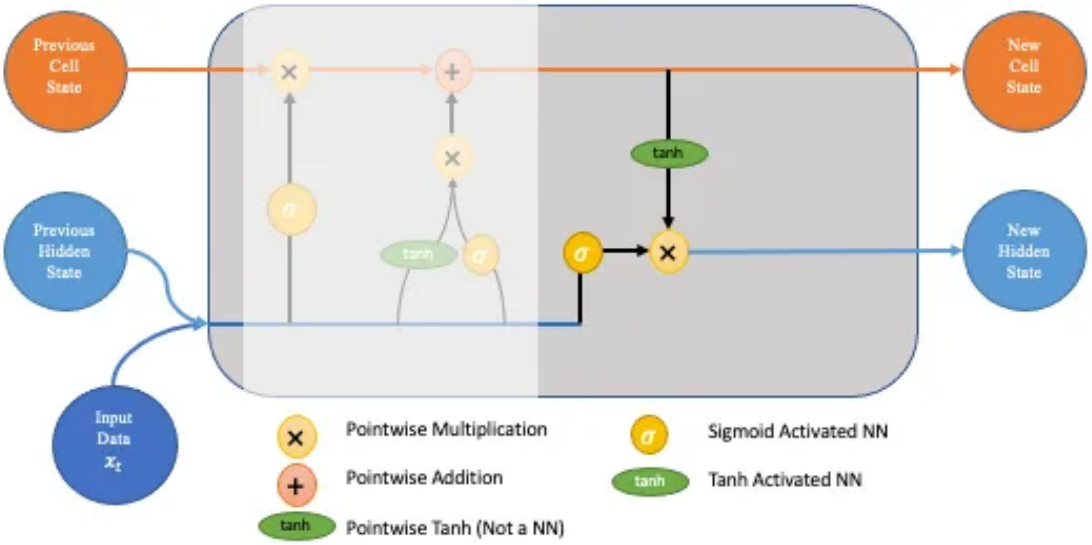

Figure 3.7: Output Gate (Dolphin 2020)

1. Apply the tanh function pointwise to the current cell state to obtain the compressed cell state, which now lies within the range [-1,1].

2. To generate the filter vector, process the previous hidden state and current input data through the sigmoid-activated neural network.

3. Apply this filter vector to the compressed cell state through pointwise multiplication.

4. Output the new hidden state.

3.3 Transformer Network

In 2017, Google proposed a notable type of neural network known as Transformers in their paper titled 'Attention is all you need'. The architecture of the Transformer is entirely predicated on the attention mechanism, which establishes global dependencies between input and output. This feature of the Transformer facilitates significantly enhanced parallelization (Menon 2021).

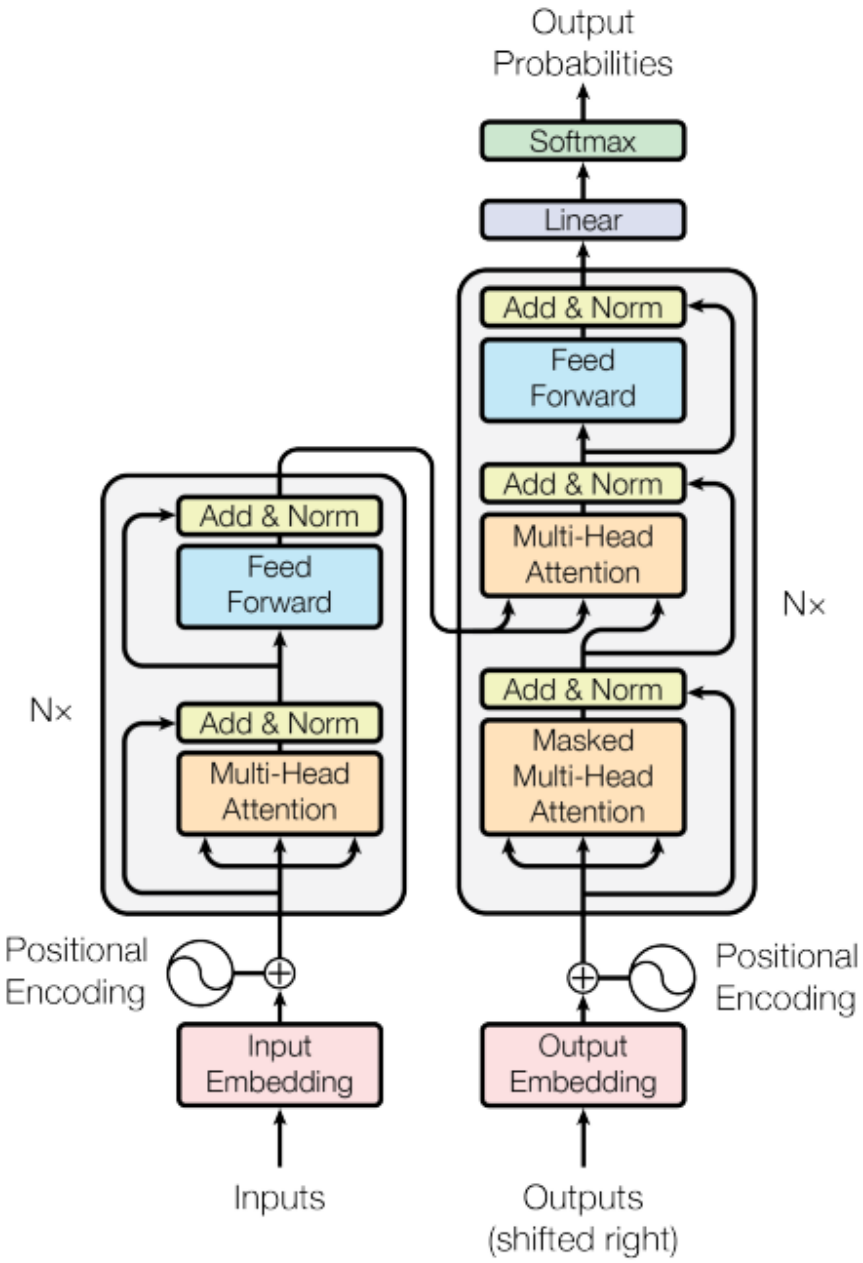

Figure 3.8: The Transformer Model architecture (Vaswani et al. 2017 (p: 3))

Recurrent Neural Networks (RNNs) were employed in an encoder-decoder structure to tackle sequence-to-sequence problems. However, the RNN seq2seq model encounters limitations when dealing with extensive sequences. When used with

long sequences, these models tend to lose the capacity to retain information from earlier elements, leading to a loss of context. The hidden state at each stage of the encoder is typically linked to a specific word in the input sentence, often one of the most recent. Consequently, if the decoder only has access to the encoder's latest hidden state, it risks losing crucial information about the earliest elements of the sequence. To mitigate this issue, the concept of the attention mechanism was introduced (Menon 2021).

The transformer design maintains the Encoder-Decoder framework from the original attention networks ("self attention network, explained in detail in 3.3.2"): given an input sequence, build an encoding depending on the context, and decode that context-based encoding to the output sequence. Figure 3.8 depicts the transformer architecture. Transformers, similar to the majority of neural machine translation models, possess an encoder-decoder architecture. The system utilizes stacked encoders with attention layers and Feed Forward Neural networks (Menon 2021).

3.3.1 Encoder

As illustrated in Figure 3.9, each encoder layer is divided into two sub-layers.

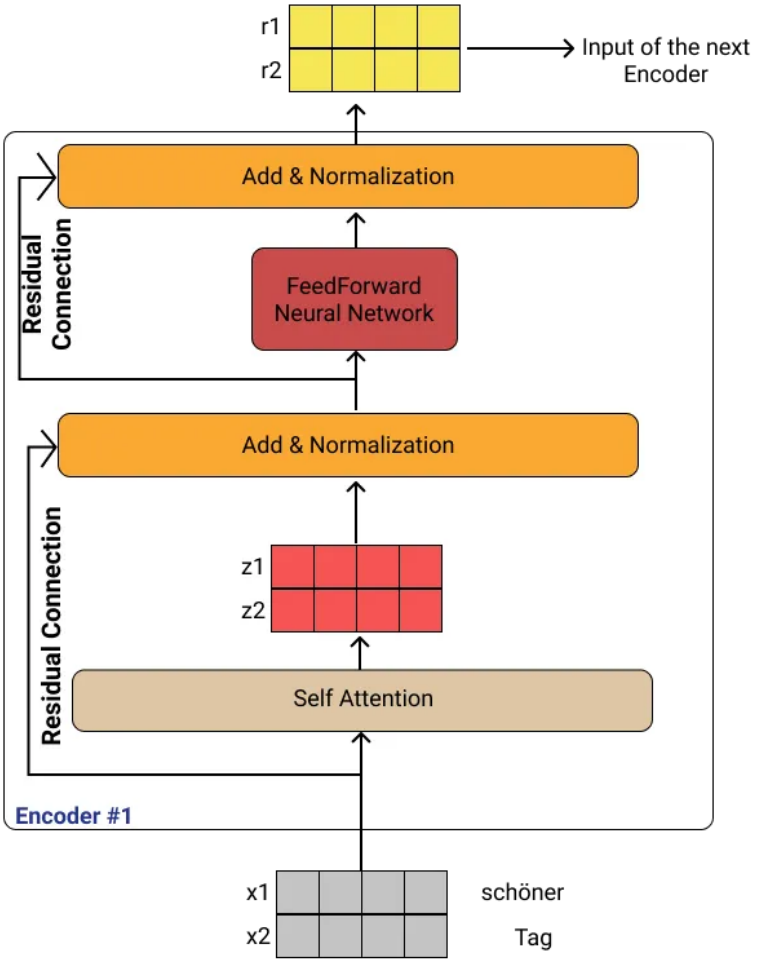

Figure 3.9: Encoder architecture (Özateş 2021)

1. **Multi-head Self-attention Layer**

2. **Feed-forward Neural Network**

In the context of a transformer model, the encoder receives an array of vectors as its input. These vectors are subsequently processed by a self-attention mechanism, which is responsible for identifying the inter-dependencies within the input. Following this, the vectors are forwarded to a Feed-forward Neural Network (FFNN) for further processing. The output from the FFNN is then relayed to the subsequent encoder in the sequence. The final encoder in the sequence is unique in that it transmits its output to all decoders in the model. The quantity of encoders utilized in this procedure is a hyper-parameter, signifying that it can be adjusted based on the specific requirements of the task (Özateş 2021).

3.3.2 Self-Attention

The transformer uses self-attention to improve word encoding by recognizing the context and the position of other relevant words in the input sequence. To compute self-attention, we first construct three vectors from the encoder's input vectors. Consequently, we generate a query vector, a key vector, and a value vector for every word. The generation of these vectors is achieved by performing matrix multiplication between the embedding and three matrices that have been acquired through training (Alammar 2018b).

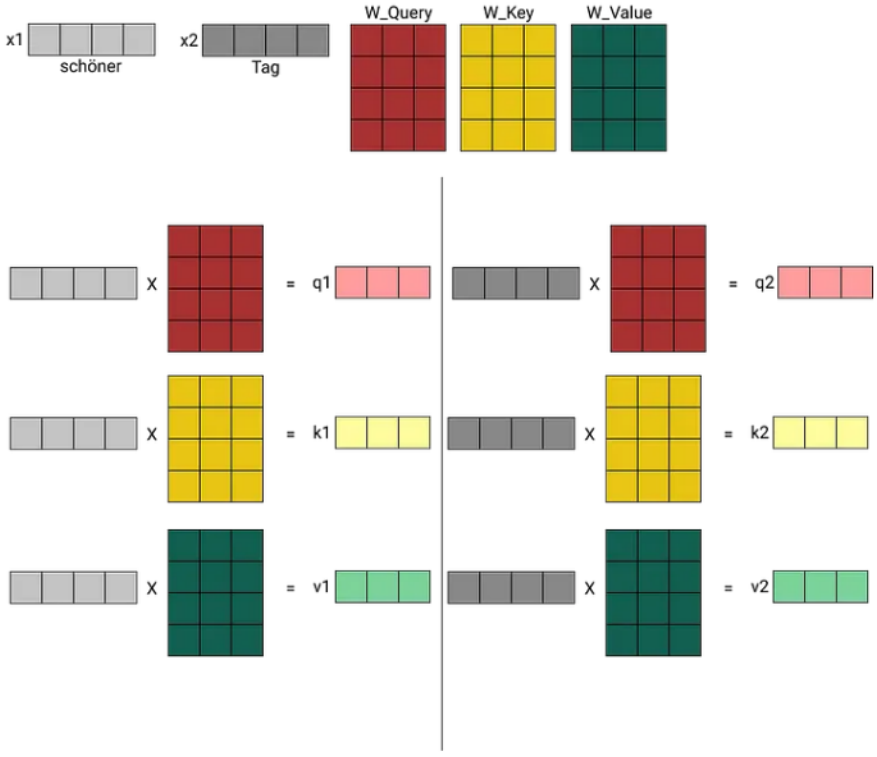

Figure 3.10: Self Attention Mechanism a (Özateş 2021)

Once these vectors have been created, the query (Q) and key (K) vectors are multiplied and then processed by dividing by the dimensions value d_k and squashing

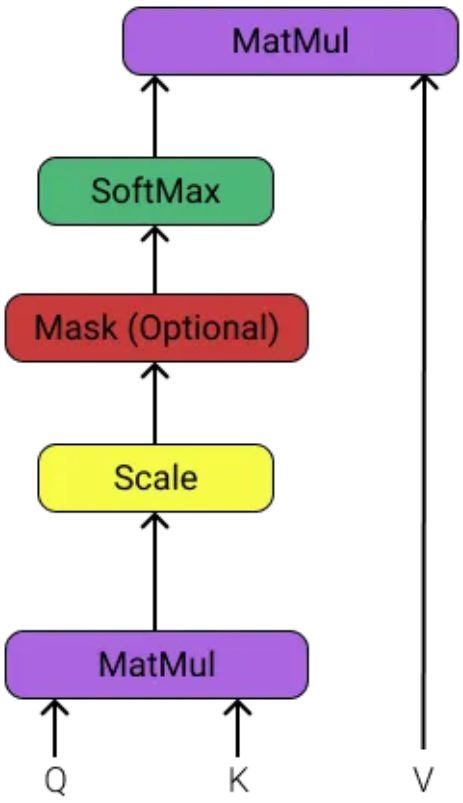

Figure 3.11: Self Attention Mechanism b (Özateş 2021)

with the softmax function. This yields an attention filter vector with values ranging from 0 to 1. This attention filter is then multiplied by the value vector (V) to form the **attention matrix** displayed in figure 3.10.

With attention, self-attention with multiple heads improves the model's capacity to focus on distinct features of the input text. Transformers are typically implemented with 8 heads, and the outputs are concatenated and multiplied by a weight matrix, as illustrated in the figure 3.11 (Özateş 2021).

$$Attention(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{3.1}$$

where:
Q = Query vector
K = Key vector
V = Value vector

3.3.3 Decoder

The architecture of the decoder mirrors that of the encoder to a considerable extent, with one notable distinction. The decoder layer incorporates an additional sublayer, which consists of a "masked" multi-head attention mechanism. This mechanism ensures that the known outputs at position < "i" are being attended to during

the prediction process. The Transformer model operates on an auto-regressive principle. This implies that it generates predictions sequentially, using the outcomes of previous predictions to inform subsequent ones. This approach allows the model to dynamically adjust its predictions based on the information it has processed up to a given point (Özateş 2021).

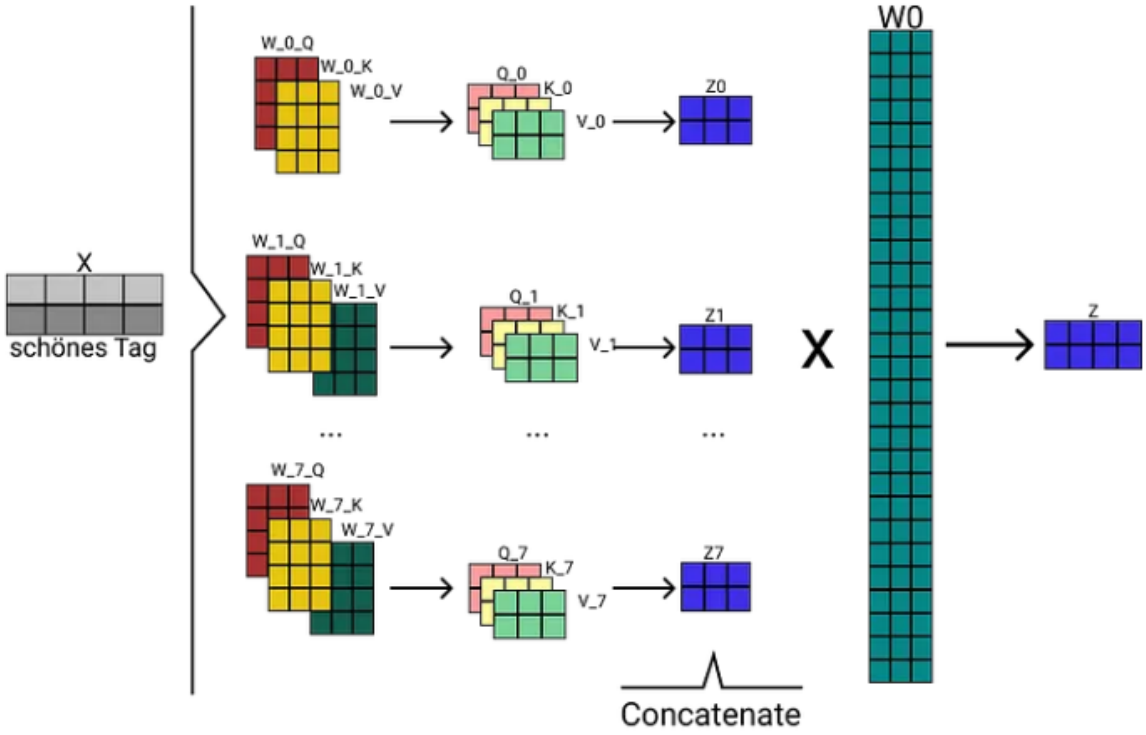

Figure 3.12: Multi headed Mechanism (Özateş 2021)

Masked multi-head attention layer: In order to avoid the calculation of attention scores for words that are yet to be encountered in the sequence, a specific mechanism is required. This mechanism is known as masking. More specifically, a look-ahead mask is employed to ensure that the decoder does not have access to future information. This mask is applied at two stages: prior to the execution of the softmax function and subsequent to the scaling of the scores. This dual application of the mask ensures that the decoder's predictions are based solely on the information it has processed up to the current point in the sequence (Menon 2021).

A look-ahead is nothing more than a matrix of 0's and negative infinities the same size as the attention matrix. The scaled matrix and the look-ahead matrix are then concatenated to produce the masked scores matrix depicted in figure 3.13. The masked scores matrix is then subjected to a softmax function, yielding a masked attention. When we apply softmax to masked scores, the negative infinities are canceled out, leaving zero attention scores for future tokens.

Positional Encoding: Because the Transformer model does not use recurrence, the model input contains no position information (time step t). It's just a bunch of words. We inject (add) some information vector (positional encoding) to each embedding to supply this critical information to the model. These vectors of positional encoding have the same dimension as embeddings (Özateş 2021).

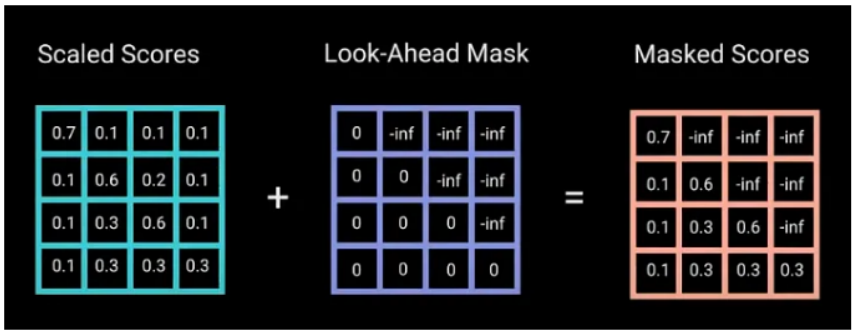

Figure 3.13: Masking the scaled scores (Phi 2020)

$$PE_{(pos,2i)} = \sin(pos/10000^{2i/d_{model}}) \tag{3.2}$$

$$PE_{(pos,2i+1)} = \cos(pos/10000^{2i/d_{model}}) \tag{3.3}$$

where:
pos = position of the token
i = Indices of the vector
d = dimensions of the vector

Transfer Learning: Language models, like as BERT, show how knowledge learnt from one dataset can be transferred to another for specific tasks (Emrich 2020). Since I have chosen BERT and XLM-RoBERTa as my models for sentiment analysis and these models are part of transfer learning models. Therefore I would like to provide some information on Transfer Learning.

Natural language processing (NLP), audio recognition, picture processing, and other domains have all seen recent advances in machine learning (ML).With cutting-edge results, machine learning (ML) approaches are being used to solve an increasing number of real-world issues that conventional statistical learning techniques are unable to address. Traditional machine learning typically requires enormous amounts of training data. One essential assumption is made: the testing and training datasets are derived from the identical distribution. Nevertheless, in numerous practical scenarios, this assumption does not consistently hold. Because of this, the three primary problems with most traditional machine learning methods are distribution mismatch, insufficient data, and mismatched computing capacity. First, a number of solutions, including distributed learning, cloud computing, data synthesis, and data argumentation, have been put out to deal with the first two issues. However, all of these proposed solutions have inherent limitations, particularly in terms of their cost, efficacy, and security. Transfer learning (TL) has lately emerged as a potential option for addressing all three of these concerns (Niu et al. 2020)

The contrast between the learning methodologies for transfer and conventional learning approaches is seen in figure 3.14. Transfer learning techniques aim to leverage knowledge from previous tasks to improve performance on a target task with limited high-quality training data, while traditional machine learning techniques aim to learn each task independently from the ground up.

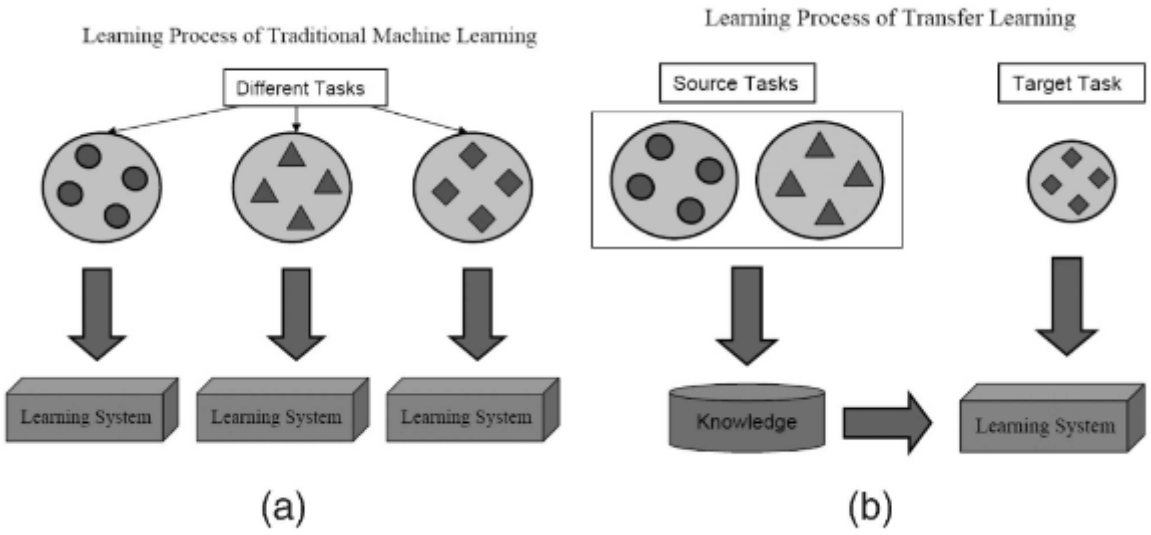

Figure 3.14: Different learning processes between (a) traditional machine learning and (b) transfer learning (Pan and Yang 2009)(p: 1346)

Why Transfer Learning ? The following are the primary advantages of transfer learning for machine learning (Castillo 2021):

1. While using pre-trained model huge amount of training time is not required and also high computational resources are not required.

2. Enhancing the efficiency of developing and deploying machine learning models.

3. An expansive methodology for resolving computer issues that utilizes diverse algorithms to tackle novel challenges.

4. Simulations can be used to teach models instead of using real-world scenarios.

"Scientists conducted a study on the latest advancements in transfer learning in the domain of natural language processing, analyzing modern techniques in machine learning and deep learning, as well as new approaches in transfer learning. Transfer learning approaches introduced novel dimensions to numerous NLP problems. Transfer learning is particularly effective in domains with limited training data. Empirical evidence has shown that transfer learning models surpass other cutting-edge NLP methods. BERT undergoes training utilizing Book Corpus, text corpora, and Wikipedia. Although it has remarkable performance in certain domains of natural language processing, there is still room for enhancement. There is a deficiency in domain and task expertise in multiple areas. Enhancements are required at this location" (Qasim et al. 2022).

3.4 BERT Explained: State of the Art Language Model for NLP

Bidirectional Encoder Representations from Transformers is usually known as BERT. BERT is widely utilized in natural language processing (NLP) tasks and

produces state-of-the-art outcomes. The objective is to develop deep bidirectional representations from unlabeled text by taking into account both the preceding and following context. By simply appending an additional output layer, a pre-trained BERT model can be augmented, leading to the development of state-of-the-art models (Devlin et al. 2018).
The remarkable efficacy of BERT can be due to two crucial elements. Initially, there are two primary pre-training tasks in the beginning: Masked Language Model (MLM) and Next Sentence Prediction (NSP). Moreover, the process of training BERT requires a significant amount of data and computational resources (Suleiman Khan 2019).

The presence of bidirectionality in a model is crucial for gaining a comprehensive comprehension of the semantic nuances of a language. Let us analyze an example to demonstrate this. The provided example comprises two sentences, both of which contain the term "bank" as illustrated in figure 3.15 :

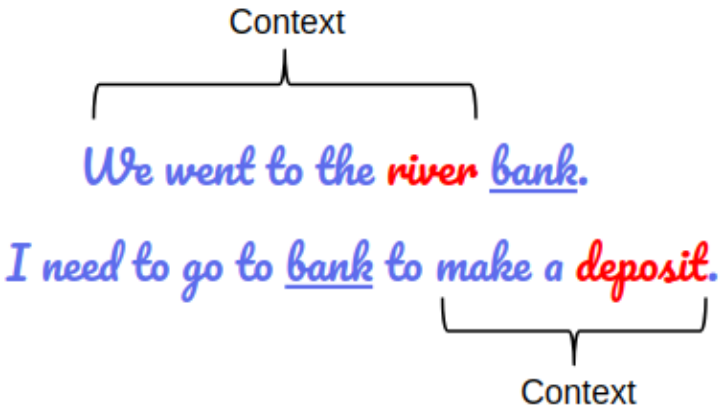

Figure 3.15: Both the left and right contexts are captured by BERT (Mohdsanadzakirizvi 2019)

Attempting to infer the meaning of the term "bank" solely from the previous or following context will inevitably lead to an erroneous interpretation in at least one of the two possibilities. An effective approach involves examining both the preceding and following circumstances prior to formulating a forecast. That is exactly the function of BERT. Ultimately, the most remarkable characteristic of BERT. To enhance its performance, we can optimize it by incorporating additional output layers to create state-of-the-art models for various natural language processing tasks (Mohdsanadzakirizvi 2019).

3.4.1 Model Architecture

The BERT design, seen in figure 3.16, is essentially a transformer architecture featuring an encoder stack. A transformer-based encoder network leverages the self-attention mechanism of the encoder. Every encoder layer is composed of two sublayers. The first layer is a self-attention layer with several heads, while the second layer is a feed-forward network with two layers. This network is applied to

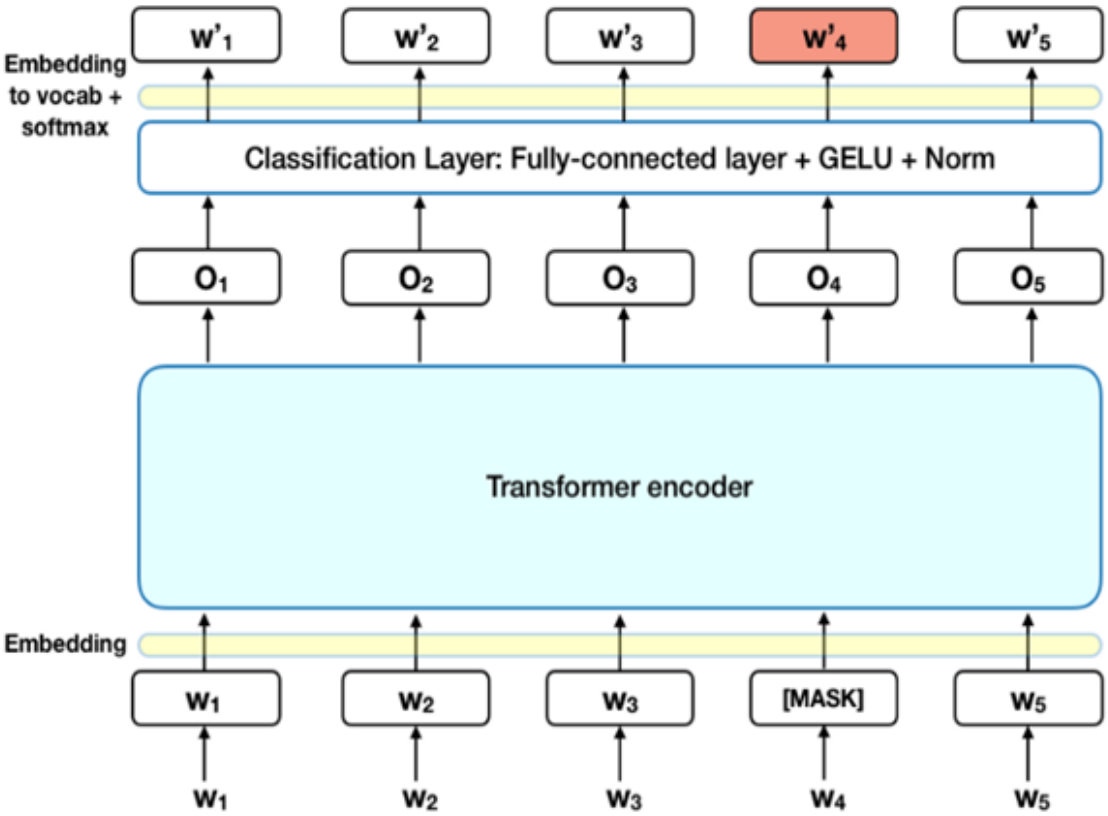

Figure 3.16: Bert Architecture (Horev 2018)

each site individually and utilizes GELU activation, which has been proven to be more effective than ReLU activation in the setting of a transformer encoder. Every sublayer is surrounded by a residual connection, which is subsequently normalized. Residual connections aid in retaining the initial information that would otherwise be lost after modifications, hence mitigating the problem of gradient vanishing during training. Layer normalization aids in stabilizing the training process by mitigating the issue of exploding gradients.

The study introduces two different sizes of BERT models (Alammar 2018a):

BERT BASE - Approximately the same size as the OpenAI Transformer, so we can compare their performance.
BERT LARGE - An exceptionally big model that achieved the most advanced results reported in the research.

The BERT models come in two sizes, both of which have a substantial number of encoder layers. The Base version has twelve encoder layers, while the high version has twenty four. Additionally, both variants include larger feed forward networks with 768 and 1024 hidden units, as well as more attention heads with 12 and 16 respectively. In contrast, the default configuration in the original paper's Transformer reference implementation consists of 6 encoder layers, 512 hidden units, and 8 attention heads (Alammar 2018a).

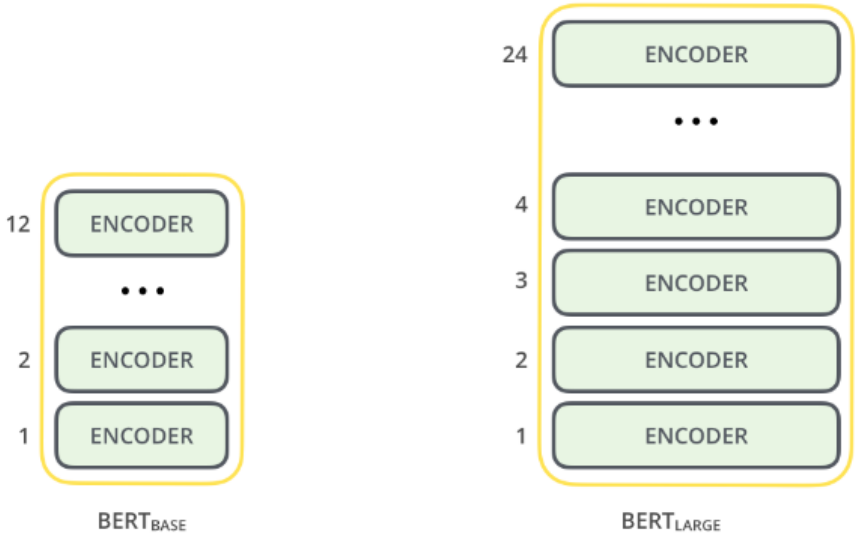

Figure 3.17: BERT Base and BERT Large (Alammar 2018a)

3.4.2 How does BERT operate?

Input-Output Format: As shown in figure, 3.18 the entire input for BERT must be provided as a single sequence. BERT use unique tokens [CLS] and [SEP] to effectively comprehend input. The [SEP] token should be placed at the end of a single input. The [SEP] token is utilized in tasks that involve multiple inputs, such as Natural Language Inference (NLI) and Question-Answering (Q-A) tasks. Its purpose is to enable the model to distinguish between the conclusion of one input and the commencement of another within the same sequence input. [CLS] is a special classification token, and the final hidden state of BERT ($h_{[CLS]}$) is utilized for tasks involving classification (Kumar 2021).

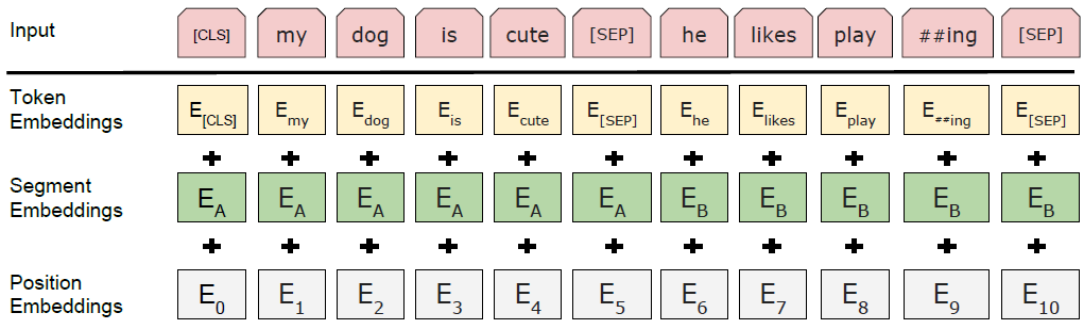

Figure 3.18: Input representation of BERT (Devlin et al. 2018)

Token Embeddings: The pre-trained embeddings for various words are known as token embeddings. The text is tokenized using a technique called WordPiece tokenization in order to get these pretrain token embeddings. This data-driven tokenization approach seeks to strike a healthy balance between large and uncommon words in the vocabulary (T n.d.).

Segment Embeddings: In essence, segment embeddings are vectors encoded with the sentence number. In BERT, the model needs to be able to determine if a given token is part of sentence A or sentence B. This is accomplished by creating two fixed tokens, one for sentence A and one for sentence B, which together are referred to as the segment embedding (T n.d.).

Position Embeddings: The word's vectorized position within the sentence is known as a position embedding. Despite being a very significant and helpful embedding, the token embeddings are unable to provide information about the token's location within a sentence. Therefore, a different embedding known as the position embeddings is employed to solve it. To represent how a token at one place attends to another token at a different position, utilize the absolute position embedding (T n.d.).

Pretraining and Fine Tuning:
The BERT architecture consists of two stages: pre-training and fine-tuning, as seen in figure 3.19. The model is trained using unlabeled data in several pre-training tasks during the pre-training phase. The BERT model is fine-tuned by leveraging labeled data from downstream activities, following the initialization with pre-trained parameters. Although the basic pre-trained parameters remain unchanged, each successive task employs its own refined model (Devlin et al. 2018).

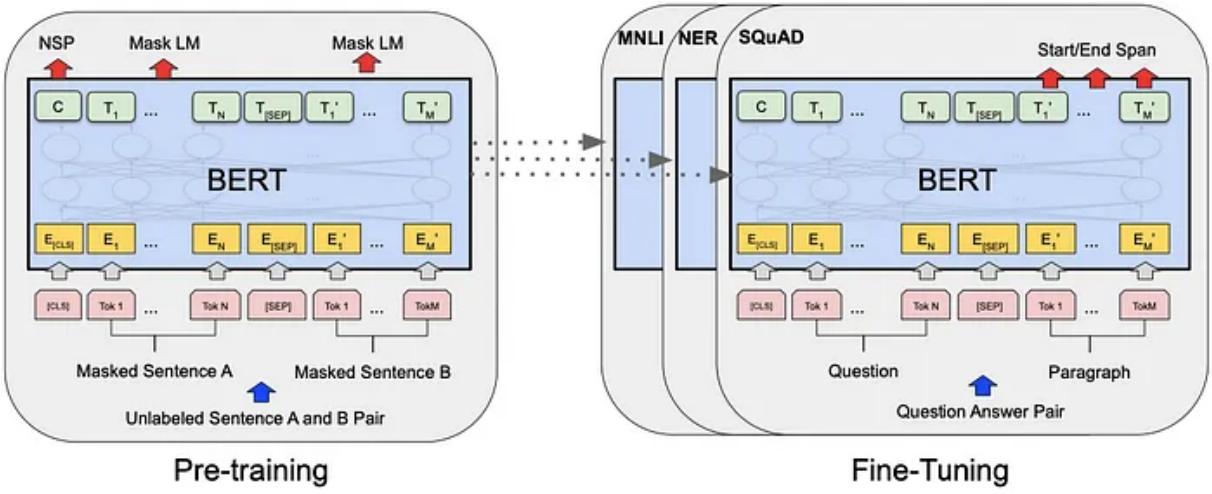

Figure 3.19: The general methodology for pre-training and fine-tuning BERT (Devlin et al. 2018)

The BERT model underwent training by employing language modeling techniques on the unsupervised Wikipedia and Bookcorpus datasets. The Masked Language Model (MLM) and Next Sentence Prediction (NSP) tasks have been completed.

Masked LM: In order to train a deep bidirectional representation, we employ a straightforward approach of randomly masking a portion of the input tokens and subsequently predicting those masked tokens as shown in figure 3.20. The term used to describe this is a "masked language model" (MLM) (Devlin et al. 2018).

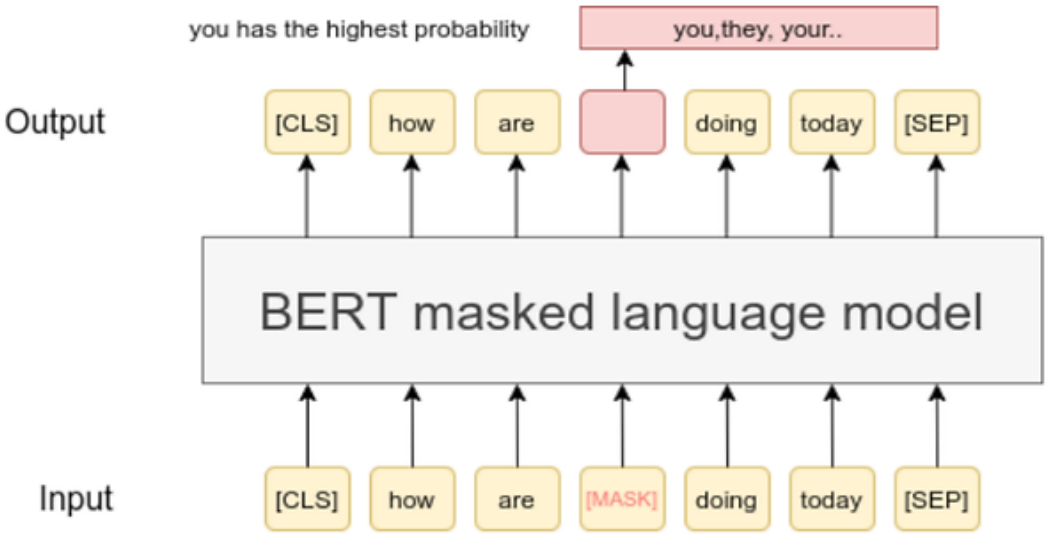

Figure 3.20: Masked Language Modeling (Chauhan 2022)

Next Sentence Prediction (NSP): Understanding the correlation between two sentences is vital for other significant subsequent tasks, such as Question Answering (QA) and Natural Language Inference (NLI). Nevertheless, language modeling does not directly depict this link. We do pre-training for a binarized next sentence prediction task, which can be easily obtained from any monolingual corpus. The purpose of this pre-training is to develop a model that comprehends the links between sentences as shown in figure 3.21 (Devlin et al. 2018).

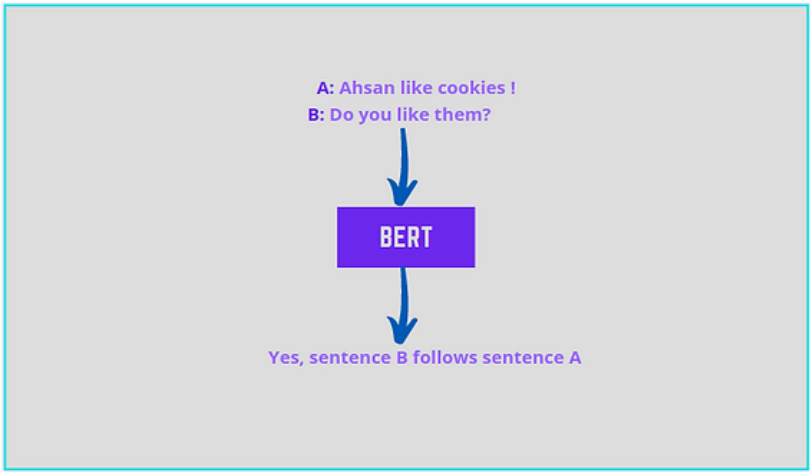

Figure 3.21: Next Sentence Prediction(NSP) (Chauhan 2022)

Except for the batch size, learning rate, and number of training epochs, the majority of model hyper-parameters remain unchanged during the process of fine-tuning after pre-training. The likelihood of dropping out was always set to 0.1. The appropriate hyper-parameter values vary depending on the task (Alammar 2018a).

The BERT paper demonstrates a variety of applications for BERT.

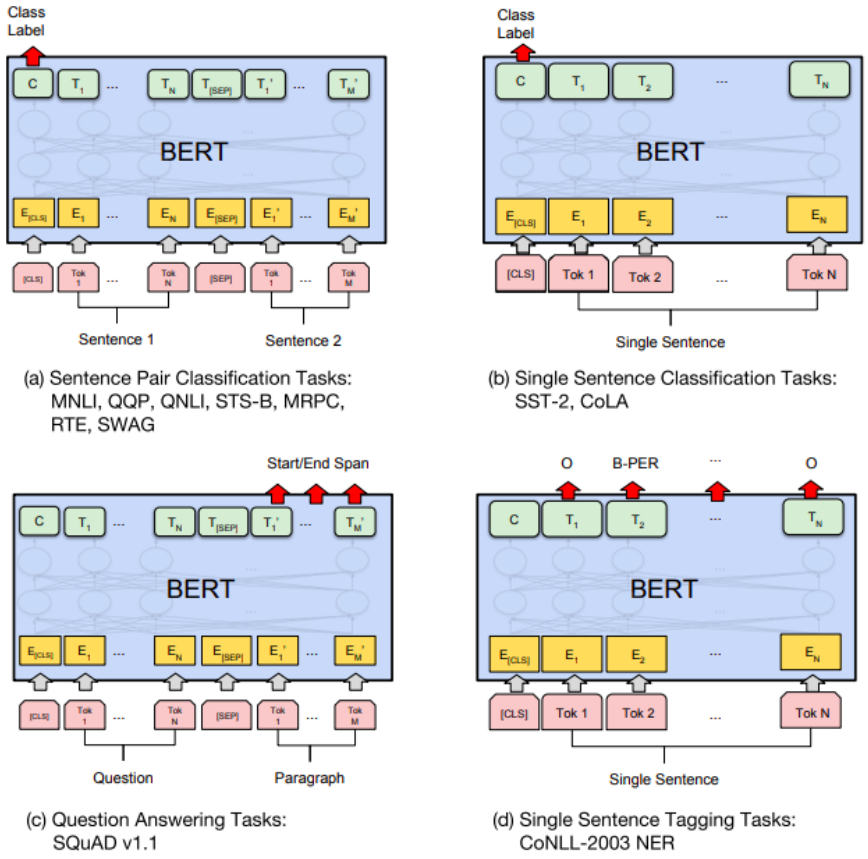

Figure 3.22: Illustrations of Fine-tuning BERT on Different Tasks (Alammar 2018a)

Illustration shown in figure 3.22 depicts the process of fine-tuning BERT for different job tasks. The construction of our task-specific models involves the integration of BERT with an extra output layer, which leads to a limited number of parameters that need to be acquired from the beginning. Problem (a) and problem (b) are at the level of sequences, but problem (c) and issue (d) are at the level of individual tokens. The image depicts the input embedding as E, the contextual representation of token i as T_i, [CLS] as the symbol used for classification output, and [SEP] as the symbol used to separate non-consecutive token sequences.

3.5 Alternative Models

Several models have undergone substantial pre-training and possess the ability to process multiple languages. Google Research has developed a multilingual variant of their BERT language model called mBERT. This model has been trained on 104 languages and shares the same structure as BERT (Yan, Li, and X. Qiu 2020). DistilBERT is a pre-trained variant of BERT that is smaller and faster, being 40% smaller and 60% faster, while still maintaining 97% of the language understanding skills (Sanh et al. 2019). In addition, at the beginning of 2019, Facebook released the Cross-Lingual Language Model (XLM) along with other pre-trained models. Constructing multilingual sentence embeddings is often challenging due to various variables, including heightened technological demands and limited training data for numerous languages.

mBERT: "Multilingual BERT" also known as mBERT, which is a Google-made language model that has already been trained. mBERT is a version of BERT that can understand more than one language. It was trained on a huge amount of text in many languages. This means that mBERT can read and write text in a lot of different languages without needing training data in those languages. Recently, it was discovered that multilingual BERT (mBERT) learns rich cross-lingual representations that help with language transfer. It also does very well at zero-shot language transfer for a variety of tasks while being taught without any parallel control (Gonen et al. 2020).

mBERT has demonstrated effectiveness in various NLP tasks, including text classification, named entity identification, and machine translation. It has been extensively utilized in both academic research and industrial applications. Furthermore, it is becoming more frequently employed as a contextual representation in several multilingual applications, including dependency parsing, cross-lingual natural language inference (XNLI), and named-entity recognition (NER) (Libovickỳ, Rosa, and Fraser 2019).

"Pretraining multilingual language models at scale sometimes leads to significant performance improvements for various cross-lingual transfer tasks. A masked language model based on Transformers is trained on over two terabytes of filtered CommonCrawl data, encompassing one hundred languages. XLM-R, our model, outperforms multilingual BERT (mBERT) on various cross-lingual benchmarks" (Conneau et al. 2019).

DistilBERT: DistilBERT, as the name suggests, is a distilled or compressed version of the original BERT model. In 2019, researchers at Hugging Face proposed a method to enhance the efficiency of BERT, a language model, without compromising its remarkable performance. DistilBERT was employed to autonomously acquire and extract significant and intricate textual representations from incoming input (Adel et al. 2022).

"As demonstrated in figure 3.23, DistilBERT leverages knowledge distillation to decrease the number of parameters in the BERT base model (BERT-base-uncased) by 40%, resulting in a 60% improvement in inference speed. The primary principle of distillation is utilizing a smaller model, such as DistilBERT, to approximate the whole output distributions of the BERT model. As a result, the BERT base now has six transformer layers (encoders) instead of the original 12. The pre-trained model encompasses a total of 66 million parameters that are subject to modification during the training process, whereas the BERT basic model comprises 110 million parameters. DistilBERT required 3.5 GPU (8xV100) days for training, while the BERT base model took 12 GPU days (8xV100) to train" (Adel et al. 2022).

"The DistilBERT model undergoes training with a dataset of 16 GB, encompassing information from the Toronto books corpus and the English Wikipedia. This dataset is identical to the one utilized for training the BERT basic model. DistilBERT is trained using a significant batch size of 400, with the use of gradient accumulation. This process entails aggregating gradients from multiple mini-batches inside a local context prior to updating the parameters at each iteration. Furthermore, the training technique lacks the incorporation of learning objectives such as next sentence prediction (NSP) and segment embeddings. Inference use dynamic masking to substitute the fixed masking employed in the BERT base model" (Adel

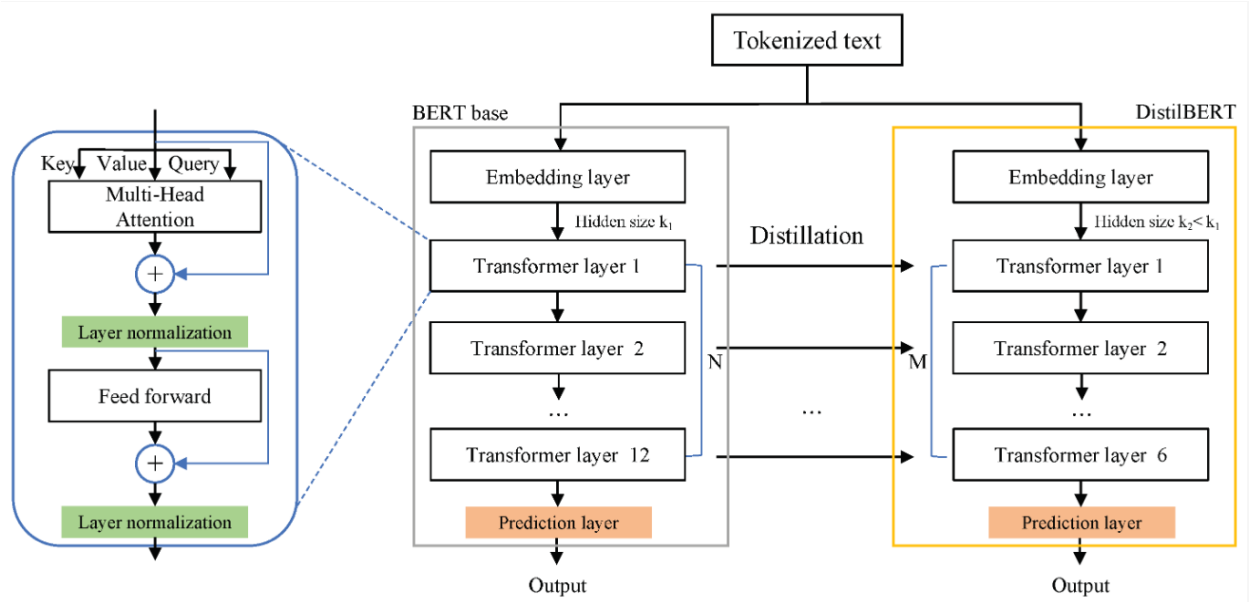

Figure 3.23: The DistilBERT model architecture and components (Adel et al. 2022) (p: 6)

et al. 2022).

XLM-RoBERTa (Cross-lingual Language Model - RoBERTa): "XLM-RoBERTa is a transformer-based language model that enhances the capabilities of RoBERTa to include a wide range of languages. The model has been pre-trained using large quantities of monolingual and parallel data from many languages, enabling it to acquire cross-lingual representations. This suggests that it can be customized and enhanced for a particular language before to its application to other languages, without requiring a substantial quantity of language-specific training data. In order to create a Named Entity Recognition (NER) model that is tailored to a certain language, it is possible to refine the XLM-RoBERTa-base model by utilizing a labeled NER dataset in that particular language. The procedure entails employing the XLM-RoBERTa tokenizer to turn textual tokens into numerical representations. These token embeddings are then fed into a neural network classifier, such as a linear layer, to make predictions regarding the named entity labels" (Höfer and Mottahedin 2023).

Multilingual Model Approach: A multilingual model is pre-trained on text from various languages, rather than treating each language individually. XLM-R underwent pre-training using data from Wikipedia and Common Crawl, encompassing 100 diverse languages. In contrast, the original BERT model was pre-trained using English Wikipedia and BooksCorpus, a compilation of self-published books, as depicted in figure 3.24. Instead of training 100 separate models on 100 different languages, we use a single BERT-type model that is pre-trained on all of this content simultaneously (McCormick 2020).

The introduction of XLM-Roberta aligns with the widespread availability of non-English models such as Finnish BERT, French BERT (also known as CamemBERT), and German BERT. In November 2019, the Facebook AI team introduced XLM-RoBERTa as an enhancement to their initial XLM-100 model. Both models are

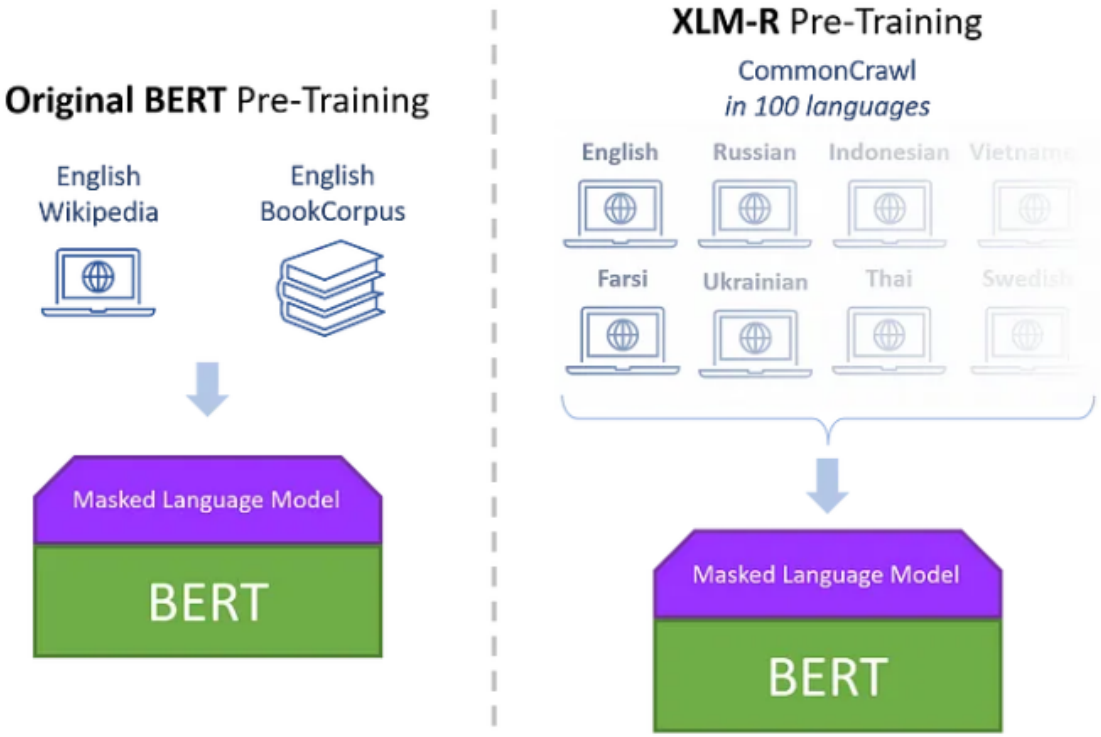

Figure 3.24: Monolingual vs Multilingual Pre-Training (McCormick 2020)

built on transformers and utilize the Masked Language Model goal. They have the ability to analyze text from a wide range of 100 languages. XLM-Roberta offers a substantial enhancement compared to the previous version by providing a substantially larger volume of training data. The processed Common Crawl data used for training requires a substantial 2.5 terabytes of storage capacity. The size of the corpus used to train its predecessor is significantly smaller than the current one, which is many orders of magnitude larger. This increase in scale is particularly evident in languages with limited resources. The term "RoBERTa" is derived from its adherence to the training procedure of the monolingual RoBERTa model (Chan 2020).

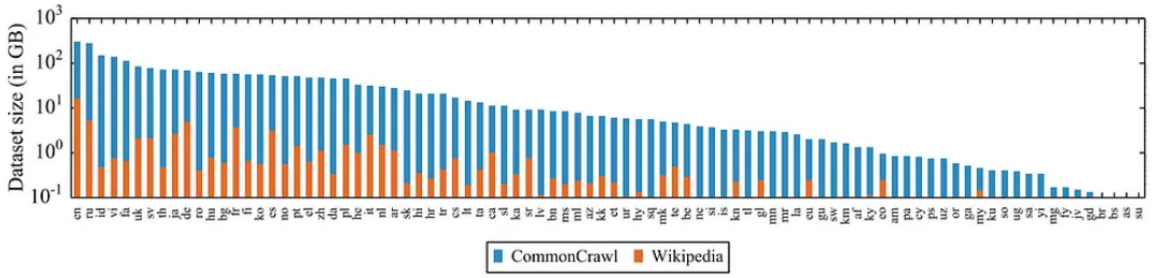

Figure 3.25: The increase in size of the CommonCrawl dataset over Wikipedia per language (Conneau et al. 2019)

Figure 3.25 displays the iso codes for 88 languages that are common to both XLM-R and XLM-100. It also shows the difference in size between the Wikipedia Corpus used by mBERT and XLM-100, as well as the CommonCrawl Corpus (Con-

neau et al. 2019).

Dynamic Masking: As illustrated in the figure 3.26, the training dataset is replicated ten times, thus each sequence is only masked in ten possible ways. Given that BERT has 40 training epochs, each sequence with the identical masking is sent to BERT four times. According to the researchers, it is somewhat preferable to employ dynamic masking, which means that masking is produced uniquely each time a sequence is sent to BERT. Overall, this results in less duplicated data during training, allowing a model to operate with more diverse data and masking patterns (Efimov 2023).

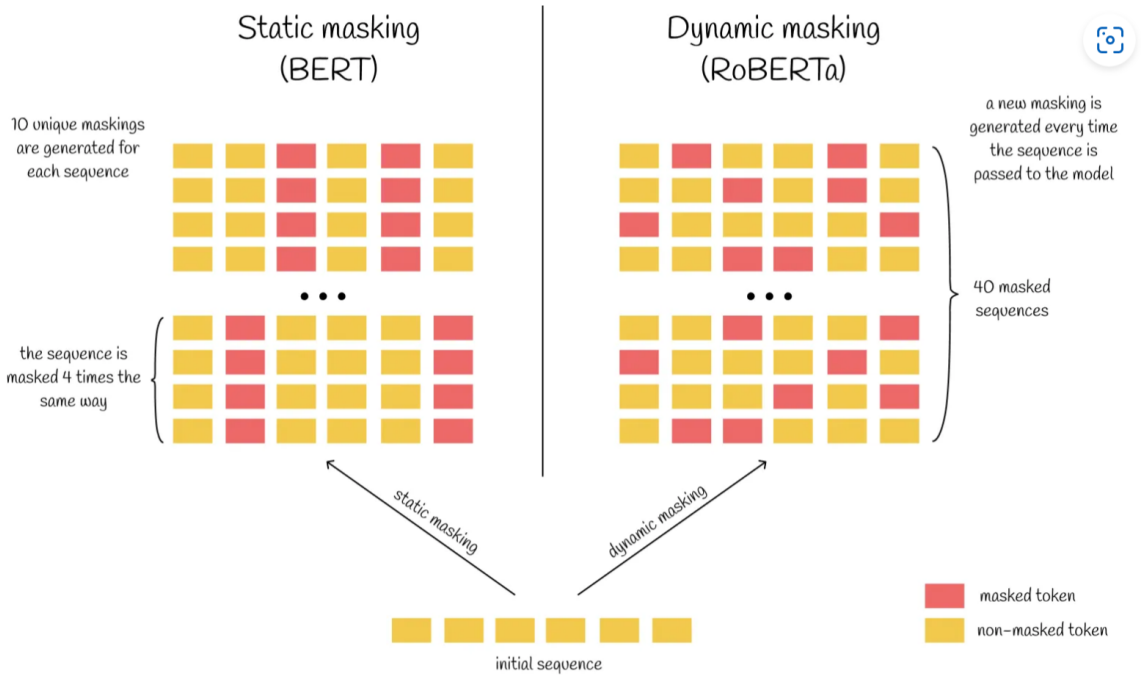

Figure 3.26: Static masking vs Dynamic masking (Efimov 2023)

Structure of BERT and XLM-RoBERTa: "Both BERT and XLM-RoBERTa comprise three primary structural components: Embedding Layers, Transformer Encoders, and a downstream structure, as seen in figure 3.27. The model produces an output vector for the initial identifier [CLS] in a single comment. The Fully Connected Layer is used to create connections, and the Sigmoid activation function is employed to transform the output results into probabilities. The cross-entropy loss is calculated by integrating the actual label of the phrase as the goal function for training XLM. The topic of discussion is RoBERTa and BERT" (Xu and Zhai 2021).

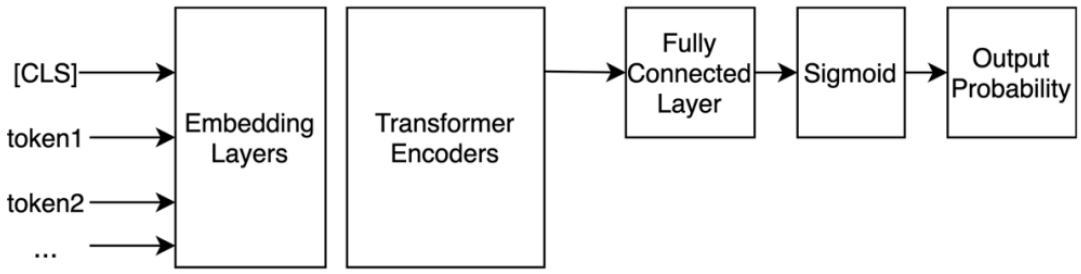

Figure 3.27: Structure of BERT and XLM-RoBERTa (Xu and Zhai 2021) (pg: 4)

Chapter 4

Application

4.1 Datasets

In this reserach I used four datasets to test the performance of BERT and XLM-RoBERTa model, namely germeval2017, Scare, SB10K and PotTS. The specifics of datasets will be discussed in section 4.1.1. Since these datasets were not preprocessed so I preprocessed them as a first step.
The data for application part of this research is the data from Twitter in the form of a raw json file, which is scrapped from Twitter using keywords **"#AKW OR #Atomkraft OR #Nuklear OR #Atomkraftwerk"**. For the rest of the thesis, we refer to the dataset by the name Twitter data.
The dataset consists of data from different time frames as shown if figure 4.1 below.

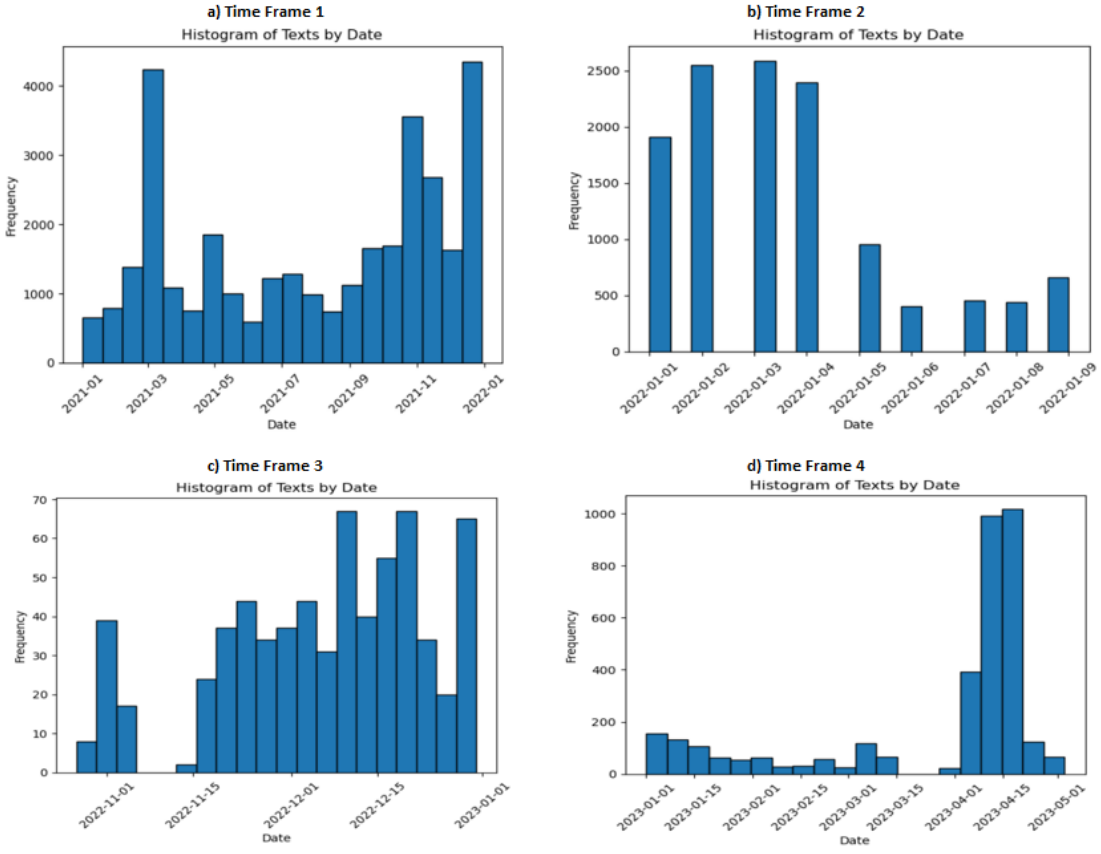

Figure 4.1: Amount of raw Twitter data for given time periods

Data is not clean and it does not have any labels as shown in the below figures 4.2, 4.3, 4.4, 4.5 for given time frames.
Time Frame 1: The data collected from Twitter include over 33,000 tweets spanning the entire year of 2021. Out of all the columns provided, the only significant ones are the "text" and "created_at" columns. These columns provide information about the tweets and the timestamps indicating when these tweets were created.

	text	created_at
0	RT @HonkHase: #KRITIS Sektor #Energie\n\nTrotz...	2021-01-09 06:25:01+00:00
1	RT @HonkHase: #KRITIS Sektor #Energie\n\nTrotz...	2021-01-08 23:17:20+00:00
2	RT @HonkHase: #KRITIS Sektor #Energie\n\nTrotz...	2021-01-08 22:40:37+00:00
3	RT @HonkHase: #KRITIS Sektor #Energie\n\nTrotz...	2021-01-08 22:39:17+00:00
4	RT @HonkHase: #KRITIS Sektor #Energie\n\nTrotz...	2021-01-08 22:28:47+00:00
...	...	...
33287	RT @Oliver_Krischer: Dass #Atomkraft Versorgun...	2021-12-30 08:42:46+00:00
33288	RT @jdoeschner: "Nukleare Denkverbote"?\nUmgke...	2021-12-30 08:41:57+00:00
33289	RT @sascha_m_k: Wenn Bundeskanzler @olafscholz...	2021-12-30 08:40:45+00:00
33290	RT @Oliver_Krischer: Dass #Atomkraft Versorgun...	2021-12-30 08:40:00+00:00
33291	RT @jdoeschner: "Nukleare Denkverbote"?\nUmgke...	2021-12-30 08:39:09+00:00

33292 rows × 2 columns

Figure 4.2: Example of raw Twitter data in json format for Time Frame 1

Time Frame 2: Approximately 12,000 tweets from January 2022 make up the data set retrieved from Twitter.

	text	created_at
0	@RainerReelfs #Nuklear bald zurück in #Deutsch...	2022-01-01 14:35:08+00:00
1	RT @eysvog3l: Nachdem aktuell in Deutschland d...	2022-01-01 14:34:54+00:00
2	RT @BerlinVic: Wir brauchen ein #Europa, das f...	2022-01-01 14:34:54+00:00
3	Noch keine Flocke Schnee im #winter22, aber di...	2022-01-01 14:34:45+00:00
4	RT @BerlinVic: Wir brauchen ein #Europa, das f...	2022-01-01 14:34:37+00:00
...	...	...
12349	RT @MarkusBecker: Öko-Etikett für #Kernenergie...	2022-01-09 11:39:31+00:00
12350	Richtig, der Blick d. PLAGE bzgl. #Atomkraft &...	2022-01-09 11:39:13+00:00
12351	RT @Alice_Weidel: Ob es ums #Klima, die #Migra...	2022-01-09 11:39:02+00:00
12352	RT @watch_union: Vorschlag zur Güte:\n\nWir la...	2022-01-09 11:38:11+00:00
12353	Billig und zuverlässig? Das Gegenteil ist der ...	2022-01-09 11:38:06+00:00

12354 rows × 2 columns

Figure 4.3: Example of raw Twitter data in json format for Time Frame 2

Time Frame 3: The dataset is sourced from Twitter and comprises 665 tweets collected between October and December 2022.

	text	created_at
0	souverän bei Illner: Robert #Habeck\n\nzu viel...	2022-11-22 00:22:15
1	Eine gute Idee aus FRANKREICH \n-- aber bei so...	2022-11-22 03:51:56
2	Ukraine aktuell 22.11.22 #AKW #Cherson #Kiew ...	2022-11-22 11:46:05
3	Ukraine aktuell 22.11.22 #AKW #Cherson #Kiew ...	2022-11-22 11:38:10
4	Ukraine aktuell 22.11.22 #AKW #Cherson #GTSOU...	2022-11-22 15:44:45
...	...	...
660	Ich verlassen diesen Ort und ziehe zu #Mastodo...	2022-11-21 11:40:32
661	Die Ergebnisse der Regierung Merkel sowie der ...	2022-11-21 13:37:40
662	Soviel zum Weiterbetrieb der hochsicheren und ...	2022-11-21 19:11:27
663	Liebe #Bundesregierung @GrueneBundestag @spdde...	2022-12-05 13:49:23
664	Die Intelligenz ist sehr hoch, die Weisheit is...	2022-11-18 22:13:48

665 rows × 2 columns

Figure 4.4: Example of raw Twitter data in json format for Time Frame 3

Time Frame 4: The data collected from Twitter include approximately 3.5 thousand tweets spanning from January to May 2023.

	text	created_at
0	Großbritannien, Finnland, Japan und viele ande...	2023-03-07 22:00:31
1	Top bezahlte und krisensichere Jobs für top au...	2023-03-08 07:12:29
2	Die Pro-Atom-Kampagne von FDP, Union und AfD u...	2023-03-08 06:30:09
3	ZUM GLÜCK STEHT DEM NOCH EIN GESETZ ENTGEGEN! ...	2023-03-08 08:42:33
4	Sind die #AKW eigentlich schon aus? Ist doch s...	2023-03-08 10:31:07
...	...	...
3504	Herausragende Dokumentation von \u2066@ZDF\u20...	2023-05-02 17:32:44
3505	#Krieg #Nuklearmacht #Eskalation \n\neine groß...	2023-01-09 00:39:48
3506	https://t.co/Z4bDMkRtQz \n\nehemaliger General...	2023-01-11 05:05:40
3507	#Luetzerath \n\nWo kann ich bitte einen #Nukle...	2023-01-15 00:26:20
3508	Abgesehen davon, @@Ricarda_Lang: natürlich kan...	2023-01-15 21:21:51

3509 rows × 2 columns

Figure 4.5: Example of raw Twitter data in json format for Time Frame 4

4.1.1 Data Sources

Explanation for four different data type sources used for simulation is taken from (Guhr et al. 2020).

GermEval-2017: GermEval-2017 was published as a component of the GermEval-2017 shared assignment on aspect-based sentiment analysis. The dataset comprises papers pertaining to the German train operator "Deutsche Bahn" sourced from social media, microblogs, news platforms, and Q & A websites. Two skilled annotators

labelled each document. The documents were gathered between May 2015 and June 2016. For this dataset I removed those texts where length of token exceeded 512 and also there were some missing entries in text column of the dataset so I had to remove those rows where text column does not have any entries.

Scare: The Sentiment Corpus of App Reviews contains Google Play Store application reviews. According to the authors, these reviews are shorter than conventional product reviews and employ colloquial language and a more flexible syntax. All of the reviews were collected between December 2014 and June 2015 and are labelled.

SB10k: SB10k includes labelled German tweets gathered between August and October of 2013. Each tweet tagged with one of five categories: positive, negative, neutral, mixed, and unknown (but in my analysis I am only considering positive, negative, neutral categories). The authors only provided the IDs of the annotated tweets, not the full-text of the tweets. When it was downloaded the tweets from Twitter using the IDs in 2018, a significant chunk of the tweets were no longer accessible. As a result, an earlier collected version2 of the data set was chosen.

PotTS: It includes tweets from the social media network Twitter. Two experts carefully labelled the texts after they were collected in 2013. The authors selected tweets from the following themes using a keyword filter: federal elections in Germany in 2013, papal conclave in 2013, debates about general political concerns, and casual everyday interactions. This dataset is comparatively much noisier.

4.1.2 Data Preprocessing

Noise Removal: The data must be free from noise, therefore the first step after receiving the data is the process of noise removal which involves removal of following Twitter terminology.
Hashtag: A hashtag is any word or phrase that is preceeded by the # symbol. When clicking on a hashtag, other tweets with the same keyword or topic will appear.
@username: A username is how you are identified on Twitter, and it is always preceeded by the @ symbol. Elon Musk, for example, is @elonmusk.
Retweet: RT, a retweet is a tweet that you forward to your followers. Retweets are frequently used on Twitter to share news or other significant findings, and they always retain original attribution.
Emojis: They are composed of special characters, letters, and other symbols and are used to describe feelings succinctly, ";):)..."
Url: Many tweets contains url which needs to be removed like all above terminologies for better sentiment analysis.

4.2 Comparing BERT and XLM-RoBERTa Model on Different Dataset

In this section I have compared BERT and XLM-RoBERTa model for different performance matrices (mainly accuracy) on four different datasets namely: ger-

meval2017, Scare, SB10K, PotTS. These datasets have different text sizes. Dataset germeval2017 has relatively large text size with an average length of the text as 25.26 tokens (after removing texts having tokens sizes greater than 512 token length), where as SB10K and PotTS datasets have medium text size with an average length of the text as 8.18 and 8.01 tokens respectively. And Scare datasets has two parts, both have small text size with an average length of the text as 1.25 tokens for alarm clock dataset and 1.28 tokens for fitness tracker dataset.
I observed that XLM-RoBERTa outperforms BERT for Scare, SB10K dataset since the accuracy of XLM-RoBERTa is higher compared to BERT in both datasets, while BERT outperforms XLM-RoBERTa for PotTS dataset. For the germeval2017 dataset there is no significant difference in the accuracy for BERT and XLM-RoBERTa methods. Also I have tested these datasets after implementing spelling error using different methods such as using ***'nlpaug'*** and ***'textaugment'*** libraries and through ***stemming*** (It is the procedure of eliminating suffixes from words in order to generate a root word, sometimes referred to as the base form.) . According to my findings it can be said that irrespective of the methods used to introduce the spelling error results are almost similar. As per comparison of BERT and XLM-RoBERTa it could be said that XLM-RoBERTa is a better model compared to BERT but not consistantly since for one dataset out of given four datasets its performance is poor. In accordance to (Y. Liu et al. 2019), "XLM-RoBERTa's superior performance may be attributed to several factors. Firstly, the model is trained for a longer duration, using larger batches and more data. Secondly, the next sentence prediction objective is removed, allowing for better training. Thirdly, the model is trained on longer sequences and the masking pattern applied to the training data is dynamically altered."
These findings are discusssed in more details in section 4.2.2 and results of stemming process is being displayed in Appendix.

Time efficiency of BERT and XLM-RoBERTa Models: The two models are being executed on Google Colab which has following machine specifications:
Processor: model name : Intel(R) Xeon(R) CPU @ 2.30GHz
RAM: 12.67 GB
Number of Cores: 1
Logical CPUs: 2

- For Germaneval2017 Dataset BERT model takes around 6-7 minutes to execute for 188 texts whereas for the same dataset and for same amount of texts XLM-RoBERTa model takes only around 57-59 seconds to execute.
- For Scare(alarm clock) Dataset BERT model takes around 25-39 seconds to execute for 588 small texts whereas for the same dataset and for same amount of texts XLM-RoBERTa model takes around 54-62 seconds to execute.
- For Scare(fitness tracker) Dataset BERT model takes around 22-32 seconds to execute for 508 small texts whereas for the same dataset and for same amount of texts XLM-RoBERTa model takes around 49-53 seconds to execute.
- For SB10K Dataset BERT model takes around 36-39 seconds to execute for 200 medium size texts whereas for the same dataset and for same amount of texts XLM-RoBERTa model takes around 32-35 seconds to execute.

- For PotTS Dataset BERT model takes around 15-25 seconds to execute for 98 medium size texts whereas for the same dataset and for same amount of texts XLM-RoBERTa model takes around 20-30 seconds to execute.

From above five points on time efficiency of BERT and XLM-RoBERTa models it can be said for these four datasets that for long text size dataset BERT model is less time efficient compared to XLM-RoBERTa , since it takes longer time to execute . For medium text size dataset both models are equal in terms of their time efficiency and for small text size dataset BERT is relatively better compared to XLM-RoBERTa .

4.2.1 Performance Assessment

In this section concepts are discussed in accordance to (Kanstrén 2020). *Precision*, *Recall*, and *F1-Score* are some basic concepts. These pertain to obtaining a more fine-grained understanding of how well a classifier is performing rather than simply looking at total *accuracy*.

The principles of True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN) provide the foundation of precision, recall, and F1-Score.

Confusion Matrix: A confusion matrix is occasionally used to show classifier performance based on the four values listed above (TP, FP, TN, FN). When these are plotted against one other, a confusion matrix is formed as shown in the figure 4.6:

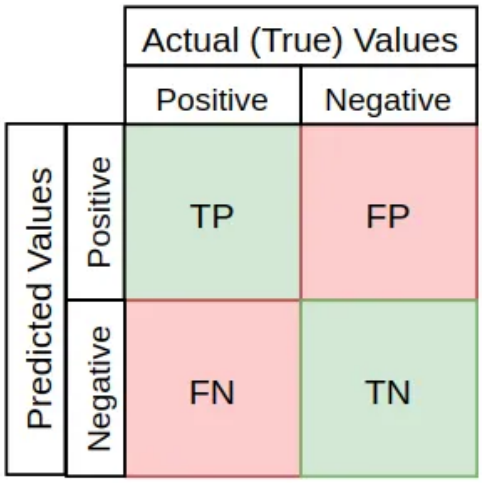

Figure 4.6: Confusion Matrix (Kanstrén 2020)

An overview figure 4.7 is often useful before delving into the details:

Accuracy: The term "accuracy," which refers to the proportion of correct predictions relative to the total number of predictions, is usually utilized as the foundational parameter for the evaluation of modelling:

$$\frac{TP+TN}{TP+TN+FP+FN} = \frac{\text{No. of Correct Predictions}}{\text{No. of all Predictions}} = \frac{\text{No. of Correct Predictions}}{\text{Size of Dataset}}$$

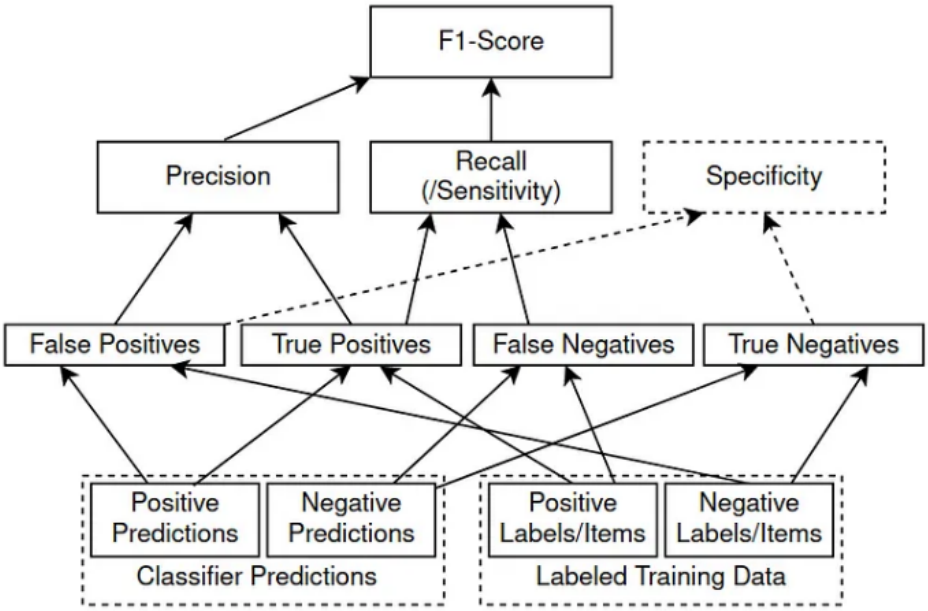

Figure 4.7: Hierarchy of Metrices (Kanstrén 2020)

The formula for calculating accuracy is the same for all three above, but the terminology is different.

Precision: Precision is a metric that quantifies the accuracy of positive forecasts, specifically the number of true positives. The formula is illustrated below:

$$\frac{TP}{TP+FP} = \frac{\text{No. of Correctly Predicted Positive Instances}}{\text{No. of Total Positive Predictions you Made}}$$

Recall / Sensitivity: Recall is a metric that quantifies the proportion of positive cases that the classifier accurately predicted out of the total number of positive cases in the dataset. It is occasionally referred to as Sensitivity.

$$\frac{TP}{TP+FN} = \frac{\text{No. of Correctly Predicted Positive Instances}}{\text{No. of Total Positive Instances in the Dataset}}$$

F1-Score: This score encompasses both precision and recall. The commonly employed word to denote this concept is the harmonic mean of the two numbers. The harmonic mean is an alternate approach of computing the "average" of integers, which is often seen as more suitable for ratios (such as precision and recall) in comparison to the traditional arithmetic mean. The formula for calculating the F1-score is presented below in this particular situation :

$$2 * \frac{Precision * Recall}{Precision + Recall}$$

Macro-average: The macro average of each performance measure is calculated individually (Vani and T. M. Rao 2019):

1. The macro-average precision is obtained by computing the average of precision values calculated for each individual class.
   Macro-average of Precision = $\frac{\sum_{i=1}^{n} P_i}{n}$
   where n = count of classes in the dataset
   P_1, P_2....P_n = Precision calculated for each class

2. The macro-average recall is determined by taking the average of all the recall values calculated for each individual class.
   Macro-average of Recall = $\frac{\sum_{i=1}^{n} R_i}{n}$
   where n= number of classes in the dataset
   R_1, R_2....R_n = Recall calculated for each class

3. The macro-average for the F1-score is calculated as the harmonic median of the macro-average of recall and precision.

Weighted-average: The weighted-average calculation for F1-Score is as follows (Vani and T. M. Rao 2019):
Weighted-average of F1 Score = $\frac{\sum_{i=1}^{n} F_i * N_i}{N}$

where F_1,F_2,....F_n = F1-Score calculated for each class
N_1,N_2,....N_n = count of instances of each class
N = total count of instances and
n = count of classes in the dataset
The weighted average for precision and recall is calculated using the same method.

Libraries: I have used following Python libraries for testing and application of BERT and XLM-RoBERTa on given datasets.

1. *Pandas:* Pandas is a popular open-source Python library used for data manipulation and analysis. It provides data structures and functions for working with structured data, making it an essential tool for data scientists and analysts. We used pandas for tasks like data loading, data cleaning and pre-processing, data visualisation, data manipulation etc.

2. *MatplotLib:* Matplotlib is a popular and widely-used open-source Python library for creating high-quality, static, animated, and interactive visualizations in 2D and 3D. It provides a flexible and versatile framework for generating a wide range of plots and charts, making it a valuable tool for data visualization, scientific plotting, and general graphing. In this research, this library is used specially for plots like data analysis and results.

3. *Seaborn:* Here, seaborn is used to create count plots that display the distribution of sentiment categories ('positive,' 'negative,' and 'neutral') for different time periods ('year_month'). Seaborn allows for customization of colors, figure size, and other plot aesthetics, making the visual representation of data more informative and visually appealing.

4. *germansentiment:* The germansentiment library is presumably a Python library or package that contains functionality related to sentiment analysis for the German language. It's specifically designed for analyzing sentiment in German text.

5. *Hugging Face Transformers Library:* Hugging Face Transformers is an open-source, state-of-the-art library that provides a wide range of pre-trained natural language processing (NLP) models. These models include transformer-based architectures like BERT, GPT, RoBERTa, and many others. We are using this library here to perform text classification tasks, such as sentiment analysis, with pre-trained models provided by the Hugging Face Transformers library.

6. *Natural Language Toolkit (NLTK):* NLTK is a popular Python library for working with human language data, including text and text processing. In our code, NLTK is used for tokenization, lemmatization, and stop word removal.

7. *re:* The 're' library is a built-in Python library used for working with regular expressions, which are powerful tools for pattern matching and text manipulation. Here, regular expressions are used to perform various text cleaning and preprocessing tasks, such as removing HTML tags, URLs, special characters, and white-space.

8. *random:* The random module is a standard Python library for generating random numbers and performing random operations. Here, the random module is used to control the introduction of spelling errors based on a defined probability (error_probability). It helps determine whether to augment text with spelling errors or keep the text as it is.

9. *os:* The os module is a powerful tool for writing platform-independent code that interacts with the underlying operating system. Here, os library is used for interacting with the operating system. Specifically, it is used to execute a shell command to gather information about the system's processor.

10. *pickle:* The pickle module in Python is used for serializing and deserializing objects. Serialization is the process of converting an object into a byte stream, and deserialization is the process of reconstructing the object from that byte stream. In our code it is used for storing and loading Python objects.

11. *nlpaug:* We use the nlpaug library, specifically the nac module, to perform character-level text augmentation by random character swap to introduce spelling errors. The nlpaug package contains fifteen augmentations that can be used to change a given text and generate a new data point. Each augmentation is stochastic, producing a result with various degrees of alteration based on the parameters provided (Lu et al. 2022). How nlpaug character-level text augmentation works is shown in figure 4.8.

Original Input: No, he was accused of being a racist white man.

Character-Based Augmentation		
Augmentation	**Augmented Output**	**Description**
Optical Character Recognition (OCR)	No, he was accu8ed of being a racist white man.	Modify characters in text by simulating OCR errors as defined by mapping.
Keyboard	No, he was accused of being a racist wUite man.	Modify characters in text by simulating typo errors via keyboard distance.
Random Character Insertion	No, he was accusnd of being a racist white man.	Modify text by injecting characters randomly.
Random Character Swap	No, he was accused of being a racist whtie man.	Modify text by swapping adjacent characters randomly.
Random Character Deletion	No, he was accuse of being a racist white man.	Modify text by deleting characters randomly.
Random Character Substitution	No, he was accusnd of being a racist white man.	Modify text by substituting characters randomly.

Figure 4.8: nlpaug Character Based Augmentation (Lu et al. 2022)

4.2.2 Selecting best BERT model for German Sentiment Analysis

For our datasets, we are taking three classes: positive, neutral, and negative. The neutral class is necessary since not all remarks by users are anticipated to be positive or negative. For example the queries in the dataset such as "What is your name?" and "Can you assist me?" have no feeling are considered as neutral (Guhr et al. 2020).
In this section I will be discussing further my results on all four datasets after preprocessing them to extract cleaned texts and introducing different percentages of spelling error. For all the figures on this section, the left hand side represents the development of BERT model and on the right side represents development of XLM-RoBERTa model.

Germaneval2017: Given that the average length of the texts in this dataset is 25.26 words, it can be classified as containing long-sized texts. The figure 4.9 demonstrates that the BERT model achieves an accuracy of 63% on cleaned text. Introducing a small amount of random spelling errors (12.81%) into the datasets does not affect its accuracy, which remains at 63%. However, increasing the random spelling errors to 25.33% slightly decreases the accuracy to 61%. The accuracy of XLM-RoBERTa for cleaned text is 63%. When introducing random spelling errors to the dataset up to a percentage of 12.81, the accuracy slightly increases to 65%. However, raising the spelling error percentage to 25.33 does not affect the accuracy, which remains at 65%. It may be argued that a random spelling error has no impact

on this dataset due to its large size. Additionally, both models perform similarly, as their accuracies are nearly identical.

BERT

Classification Report:

	precision	recall	f1-score	support
negative	0.42	0.63	0.50	46
neutral	0.76	0.70	0.73	125
positive	0.33	0.06	0.10	17
accuracy			0.63	188
macro avg	0.50	0.46	0.44	188
weighted avg	0.64	0.63	0.62	188

XLM-RoBERTa

Classification Report:

	precision	recall	f1-score	support
negative	0.36	0.09	0.14	46
neutral	0.66	0.92	0.77	125
positive	0.00	0.00	0.00	17
accuracy			0.63	188
macro avg	0.34	0.34	0.30	188
weighted avg	0.53	0.63	0.55	188

Cleaned Text

Classification Report:

	precision	recall	f1-score	support
Negative	0.44	0.59	0.50	46
Neutral	0.73	0.73	0.73	125
Positive	0.00	0.00	0.00	17
accuracy			0.63	188
macro avg	0.39	0.44	0.41	188
weighted avg	0.59	0.63	0.61	188

Cleaned Text

Classification Report:

	precision	recall	f1-score	support
negative	0.57	0.09	0.15	46
neutral	0.67	0.95	0.79	125
positive	0.00	0.00	0.00	17
accuracy			0.65	188
macro avg	0.41	0.35	0.31	188
weighted avg	0.58	0.65	0.56	188

12.81% Error

Classification Report:

	precision	recall	f1-score	support
Negative	0.35	0.48	0.41	46
Neutral	0.74	0.74	0.74	125
Positive	0.00	0.00	0.00	17
accuracy			0.61	188
macro avg	0.36	0.40	0.38	188
weighted avg	0.58	0.61	0.59	188

12.81% Error

Classification Report:

	precision	recall	f1-score	support
negative	0.38	0.07	0.11	46
neutral	0.67	0.96	0.79	125
positive	0.00	0.00	0.00	17
accuracy			0.65	188
macro avg	0.35	0.34	0.30	188
weighted avg	0.54	0.65	0.55	188

25.33% Error

25.33% Error

Figure 4.9: BERT and XLM-RoBERTa results for different amounts of spelling errors in Germaneval2017 Dataset

Scare Alarm Clocks DataSet: This dataset consists of texts with an average length of 1.25 words, indicating that the texts are quite short in size. The accuracy of the BERT model can be observed in figure 4.10. For cleaned text, the model achieves an accuracy of 58%. However, when introducing a certain level of random spelling errors (33.16%) into the datasets, the accuracy drops to 48%. Furthermore, with an increased level of random spelling errors (71.44%), the accuracy further decreases to 39%. after using XLM-RoBERTa on cleaned text, the accuracy is 70%. However, after introducing random errors to the dataset up to a percentage of 33.16, the accuracy decreases to 59%. Furthermore, increasing the spelling errors to a percentage of 71.44 results in a further fall in accuracy to 49%. It has been shown that spelling errors have a negative impact on both models. Increasing the number of spelling errors decreases the accuracy of both models. Additionally, XLM-RoBERTa outperforms BERT since its accuracy is consistently greater in all scenarios.

Scare Fitness Tracker DataSet: This dataset contains texts with an average length of 1.28 words, indicating that the texts are very short in size. The figure

BERT

Classification Report:

	precision	recall	f1-score	support
Negative	0.44	0.74	0.56	78
Neutral	0.67	0.02	0.04	213
Positive	0.61	0.92	0.74	308
accuracy			0.58	599
macro avg	0.57	0.56	0.44	599
weighted avg	0.61	0.58	0.46	599

XLM-RoBERTa

Classification Report:

	precision	recall	f1-score	support
negative	0.75	0.27	0.40	78
neutral	0.56	0.97	0.71	213
positive	0.95	0.63	0.75	308
accuracy			0.70	599
macro avg	0.75	0.62	0.62	599
weighted avg	0.78	0.70	0.69	599

Cleaned Text

Classification Report:

	precision	recall	f1-score	support
Negative	0.27	0.76	0.39	78
Neutral	0.67	0.02	0.04	213
Positive	0.60	0.72	0.65	308
accuracy			0.48	599
macro avg	0.51	0.50	0.36	599
weighted avg	0.58	0.48	0.40	599

Cleaned Text

Classification Report:

	precision	recall	f1-score	support
negative	0.60	0.19	0.29	78
neutral	0.48	0.97	0.64	213
positive	0.94	0.43	0.59	308
accuracy			0.59	599
macro avg	0.67	0.53	0.51	599
weighted avg	0.73	0.59	0.57	599

33.16% Spelling Error

Classification Report:

	precision	recall	f1-score	support
Negative	0.17	0.63	0.27	78
Neutral	0.40	0.01	0.02	213
Positive	0.59	0.60	0.59	308
accuracy			0.39	599
macro avg	0.39	0.41	0.29	599
weighted avg	0.47	0.39	0.35	599

33.16% Spelling Error

Classification Report:

	precision	recall	f1-score	support
negative	0.25	0.05	0.09	78
neutral	0.42	0.96	0.58	213
positive	0.91	0.27	0.42	308
accuracy			0.49	599
macro avg	0.53	0.43	0.36	599
weighted avg	0.65	0.49	0.43	599

71.44% Spelling Error

71.44% Spelling Error

Figure 4.10: BERT and XLM-RoBERTa results for different amounts of spelling errors in Scare Alarm Clocks DataSet

4.11 demonstrates that the BERT model achieves an accuracy of 60% when applied to cleaned text. However, when introducing a moderate amount of random spelling errors (30.75%) into the datasets, the accuracy decreases to 50%. Furthermore, increasing the random spelling errors to 60.80% results in a further decrease in accuracy to 41%. After using XLM-RoBERTa on cleaned text, the accuracy is 73%. However, after introducing random errors to the dataset up to a percentage of 30.75, the accuracy decreases to 62%. Furthermore, increasing the spelling errors to a percentage of 60.80 results in a further fall in accuracy to 50%. It is evident that XLM-RoBERTa outperforms BERT due to its significantly higher accuracy across all scenarios. Spelling errors have a negative impact on both models, as increasing the number of spelling errors decreases their accuracy.

SB10K DataSet: This dataset has an average length of 8.18 words, indicating that the texts are of average size. In the above figure 4.12, the BERT model achieves an accuracy of 59% on cleaned text. However, when a certain quantity of random

BERT

Classification Report:

	precision	recall	f1-score	support
Negative	0.53	0.84	0.65	86
Neutral	0.88	0.08	0.14	185
Positive	0.61	0.90	0.73	250
accuracy			0.60	521
macro avg	0.67	0.61	0.51	521
weighted avg	0.69	0.60	0.51	521

XLM-RoBERTa

Classification Report:

	precision	recall	f1-score	support
negative	0.83	0.34	0.48	86
neutral	0.59	0.98	0.74	185
positive	0.94	0.68	0.79	250
accuracy			0.73	521
macro avg	0.79	0.66	0.67	521
weighted avg	0.80	0.73	0.72	521

Cleaned Text

Classification Report:

	precision	recall	f1-score	support
Negative	0.36	0.81	0.50	86
Neutral	0.88	0.04	0.07	185
Positive	0.58	0.74	0.65	250
accuracy			0.50	521
macro avg	0.60	0.53	0.41	521
weighted avg	0.65	0.50	0.42	521

Cleaned Text

Classification Report:

	precision	recall	f1-score	support
negative	0.77	0.27	0.40	86
neutral	0.50	0.97	0.66	185
positive	0.91	0.49	0.64	250
accuracy			0.62	521
macro avg	0.73	0.58	0.57	521
weighted avg	0.74	0.62	0.61	521

30.75% Spelling Error

Classification Report:

	precision	recall	f1-score	support
Negative	0.27	0.72	0.39	86
Neutral	0.83	0.03	0.05	185
Positive	0.52	0.59	0.55	250
accuracy			0.41	521
macro avg	0.54	0.45	0.33	521
weighted avg	0.59	0.41	0.35	521

30.75% Spelling Error

Classification Report:

	precision	recall	f1-score	support
negative	0.50	0.07	0.12	86
neutral	0.42	0.99	0.59	185
positive	0.93	0.28	0.44	250
accuracy			0.50	521
macro avg	0.62	0.45	0.38	521
weighted avg	0.68	0.50	0.44	521

60.80% Spelling Error

60.80% Spelling Error

Figure 4.11: BERT and XLM-RoBERTa results for different amounts of spelling errors in Scare Fitness Tracker DataSet

spelling errors (17.20%) are introduced into the datasets, the accuracy drops to 50%. Increasing the random spelling errors to 33.75% does not further affect the accuracy, which remains at 50%. After using XLM-RoBERTa on cleaned text, the accuracy is 71%. However, after introducing random errors to the dataset, up to a percentage of 17.20, the accuracy somewhat decreases to 69%. Furthermore, increasing the spelling errors to a percentage of 33.75 results in a further fall in accuracy to 67%. It is evident that XLM-RoBERTa outperforms BERT due to its significantly higher accuracy across all scenarios. Spelling errors have a limited impact on both models, as increasing the number of spelling errors does not significantly decrease their accuracy.

PotTS DataSet: This dataset is having an average length of 8.01 words present which is also average size texts. Figure 4.13 exhibits for cleaned text the BERT model has an accuracy of 43% and XLM-RoBERTa is 33%. Upon giving some amount of random spelling error (21.08%) into the datasets BERT accuracy is decreased to 39% and for XLM-RoBERTa 32%. When increasing random spelling

BERT

Classification Report:

	precision	recall	f1-score	support
negative	0.29	0.71	0.41	31
neutral	0.89	0.60	0.72	130
positive	0.50	0.46	0.48	39
accuracy			0.59	200
macro avg	0.56	0.59	0.54	200
weighted avg	0.72	0.59	0.62	200

XLM-RoBERTa

Classification Report:

	precision	recall	f1-score	support
negative	0.69	0.29	0.41	31
neutral	0.72	0.93	0.81	130
positive	0.63	0.31	0.41	39
accuracy			0.71	200
macro avg	0.68	0.51	0.54	200
weighted avg	0.70	0.71	0.67	200

Cleaned Text

Classification Report:

	precision	recall	f1-score	support
negative	0.22	0.65	0.33	31
neutral	0.82	0.51	0.63	130
positive	0.48	0.36	0.41	39
accuracy			0.50	200
macro avg	0.51	0.50	0.46	200
weighted avg	0.66	0.50	0.54	200

Cleaned Text

Classification Report:

	precision	recall	f1-score	support
negative	0.83	0.16	0.27	31
neutral	0.69	0.96	0.80	130
positive	0.54	0.18	0.27	39
accuracy			0.69	200
macro avg	0.69	0.43	0.45	200
weighted avg	0.68	0.69	0.62	200

17.20% Spelling Error

Classification Report:

	precision	recall	f1-score	support
negative	0.22	0.65	0.33	31
neutral	0.82	0.51	0.63	130
positive	0.48	0.36	0.41	39
accuracy			0.50	200
macro avg	0.51	0.50	0.46	200
weighted avg	0.66	0.50	0.54	200

17.20% Spelling Error

Classification Report:

	precision	recall	f1-score	support
negative	0.60	0.10	0.17	31
neutral	0.67	0.98	0.80	130
positive	0.50	0.05	0.09	39
accuracy			0.67	200
macro avg	0.59	0.38	0.35	200
weighted avg	0.63	0.67	0.56	200

33.75% Spelling Error

33.75% Spelling Error

Figure 4.12: BERT and XLM-RoBERTa results for different amounts of spelling errors in SB10K Dataset

error (33.14%) accuracy in BERT decreases little to 38% and in XLM-RoBERTa accuracy further goes little down to 28%. It is clear to understand here that this is the only case where XLM-RoBERTa performs poorly compared to BERT since accuracy is higher in the case of BERT in all scenarios. And spelling error does affect both the models to a lower extent, since upon increasing the spelling error for both the models accuracy does not go down too much.

BERT

Classification Report:

	precision	recall	f1-score	support
mixed	0.00	0.00	0.00	4
negative	0.31	0.74	0.44	23
neutral	0.47	0.54	0.50	26
positive	0.85	0.24	0.38	45
accuracy			0.43	98
macro avg	0.41	0.38	0.33	98
weighted avg	0.58	0.43	0.41	98

XLM-RoBERTa

Classification Report:

	precision	recall	f1-score	support
mixed	0.00	0.00	0.00	4
negative	1.00	0.13	0.23	23
neutral	0.28	1.00	0.44	26
positive	1.00	0.07	0.12	45
accuracy			0.33	98
macro avg	0.57	0.30	0.20	98
weighted avg	0.77	0.33	0.23	98

Cleaned Text

Classification Report:

	precision	recall	f1-score	support
mixed	0.00	0.00	0.00	4
negative	0.28	0.74	0.41	23
neutral	0.45	0.54	0.49	26
positive	1.00	0.16	0.27	45
accuracy			0.39	98
macro avg	0.43	0.36	0.29	98
weighted avg	0.65	0.39	0.35	98

Cleaned Text

Classification Report:

	precision	recall	f1-score	support
mixed	0.00	0.00	0.00	4
negative	1.00	0.09	0.16	23
neutral	0.28	1.00	0.44	26
positive	1.00	0.07	0.12	45
accuracy			0.32	98
macro avg	0.57	0.29	0.18	98
weighted avg	0.77	0.32	0.21	98

21.08% Spelling Error

Classification Report:

	precision	recall	f1-score	support
mixed	0.00	0.00	0.00	4
negative	0.28	0.65	0.39	23
neutral	0.42	0.50	0.46	26
positive	0.69	0.20	0.31	45
accuracy			0.38	98
macro avg	0.35	0.34	0.29	98
weighted avg	0.49	0.38	0.35	98

21.08% Spelling Error

Classification Report:

	precision	recall	f1-score	support
mixed	0.00	0.00	0.00	4
negative	0.50	0.04	0.08	23
neutral	0.26	0.96	0.41	26
positive	1.00	0.02	0.04	45
accuracy			0.28	98
macro avg	0.44	0.26	0.13	98
weighted avg	0.65	0.28	0.15	98

33.14% Spelling Error

33.14% Spelling Error

Figure 4.13: BERT and XLM-RoBERTa results for different amounts of spelling errors in PotTS Dataset

4.3 Application for Unlabelled Data

The outcomes for the Twitter dataset vary between the BERT and XLM-RoBERTa approaches. I have plotted for development of BERT and XLM-RoBERTa model for the data available in four different time frames which is being discussed in section 4.1.

Time Frame 1: Using BERT model I can see from figure 4.14 that in the year 2021 most of the tweets were having neutral sentiment but some percentage of negative and a very small percentage of positive sentiments were also recorded. But for the same year 2021 XLM-RoBERTa has different predictions, it records high percentages of neutral tweets compared to BERT model and percentages of negative and positive tweets are almost negligible.

Time Frame 2: As shown in figure 4.15 for the January month of 2022 BERT

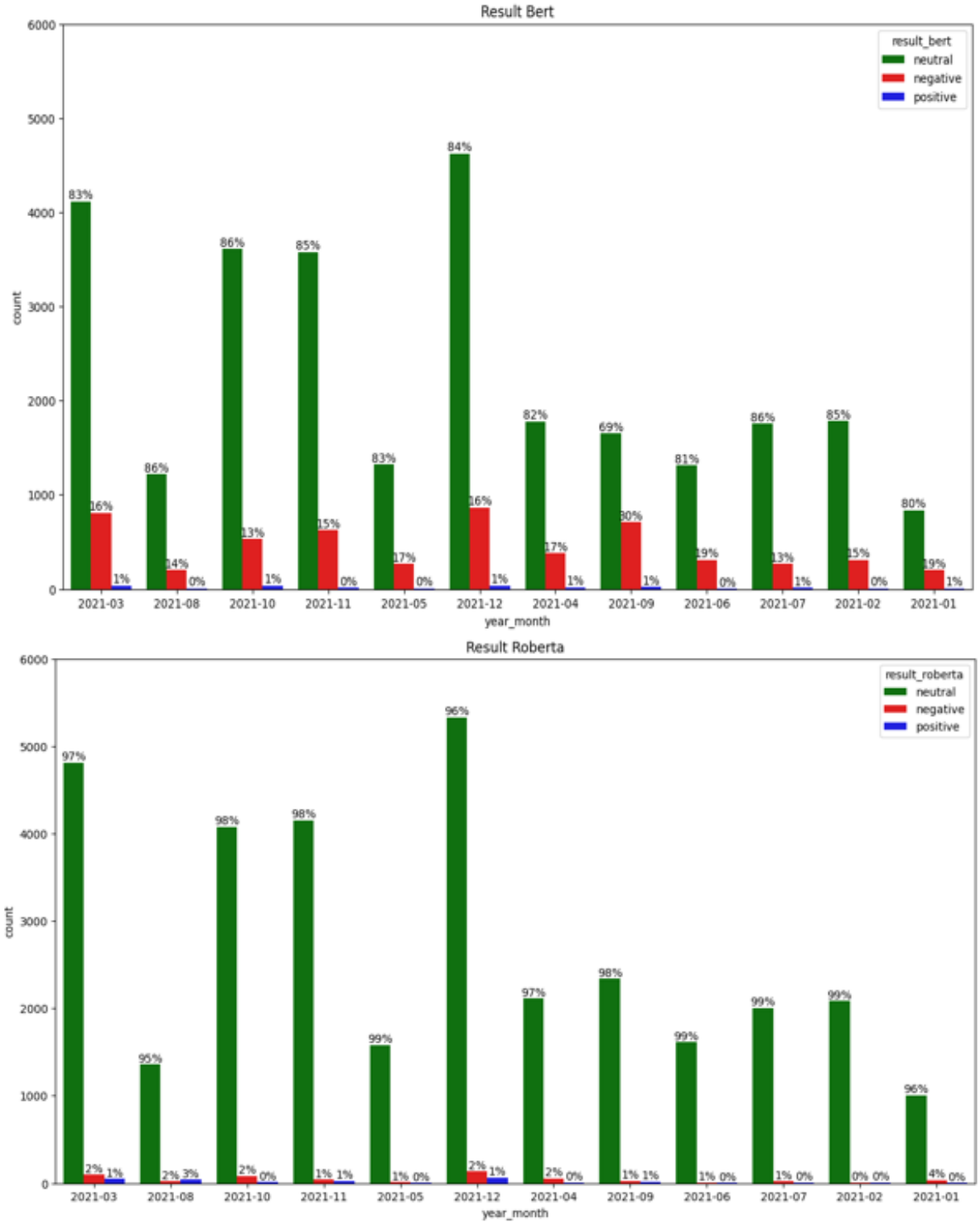

Figure 4.14: BERT and XLM-RoBERTa Sentiment plot for Time Frame 1 dataset

model has predictions such as 83% neutral, 14% negative and 3% positive and XLM-RoBERTa predictions varies such as 98% neutral, 1% negative, 1% positive.

Time Frame 3: It can be seen from figure 4.16 that in October 2022 very few tweets were fetched and in these tweets BERT predictions are 44% neutral, 56% negative and 0% positive but for the same time XLM-RoBERTa predicts all

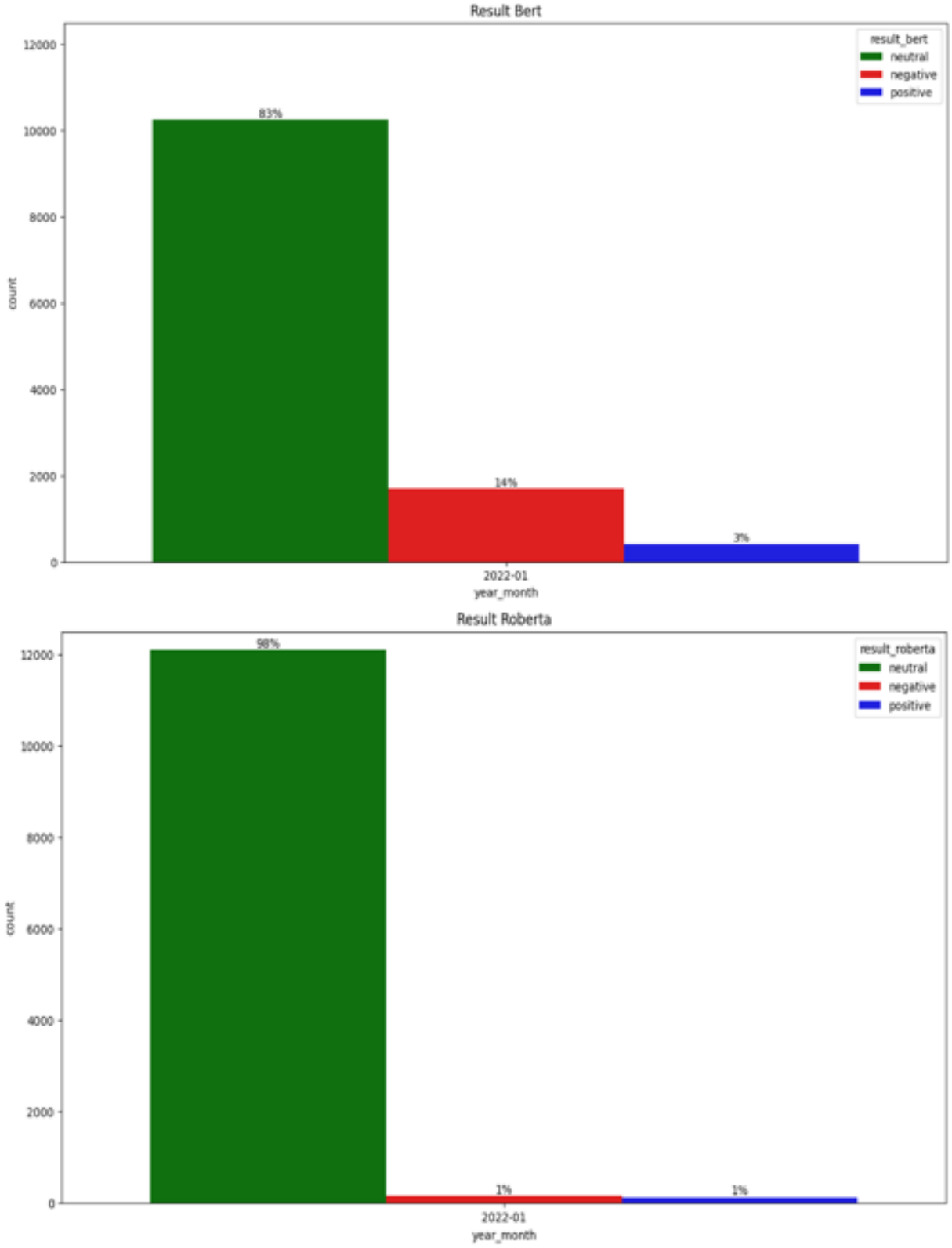

Figure 4.15: BERT and XLM-RoBERTa Sentiment plot for Time Frame 2 dataset

the tweets as neutral (100%). For november 2022, 65% are neutal , 35% negative and 0% positive for BERT similarly for XLM-RoBERTa predictions are almost 96% neutral, 3% negative and 1% positive. And in December 2022 68% neutral , 30% negative and 2% positive predictions were made by BERT and XLM-RoBERTa

predictions are 96% neutral, 3% negative and 1% positive tweets .

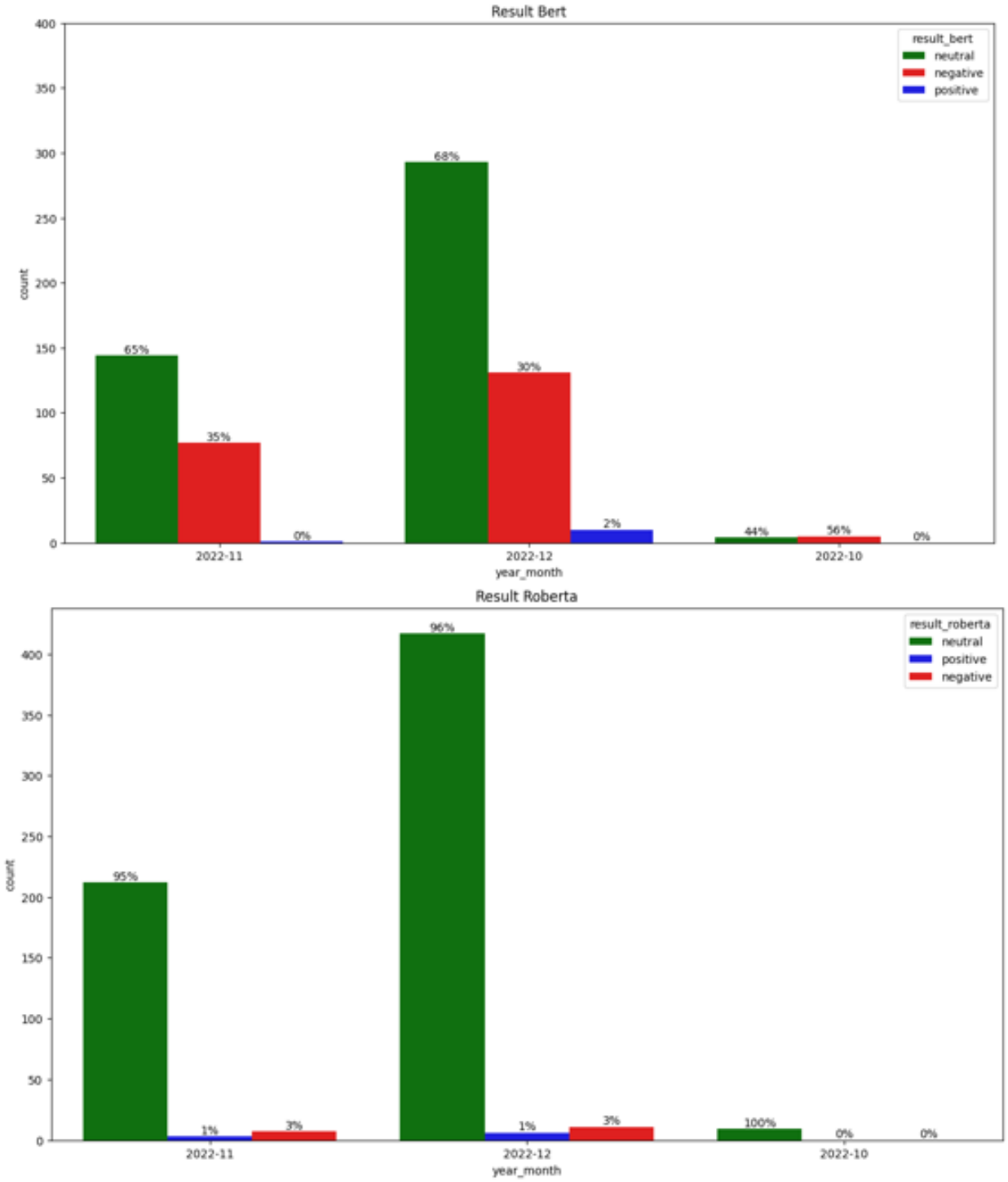

Figure 4.16: BERT and XLM-RoBERTa Sentiment plot for Time Frame 3 dataset

Time Frame 4: From the figure 4.17 difference in results of BERT and XLM-RoBERTa can be seen for the time January to May 2023. Positive, negative, neutrals percentages are counted similar to other time frames. BERT predictions are neutral and negative sentiments but at the same time XLM-RoBERTa predictions are mostly neutral.

In summary for all the four time frames it can be said that BERT model predic-

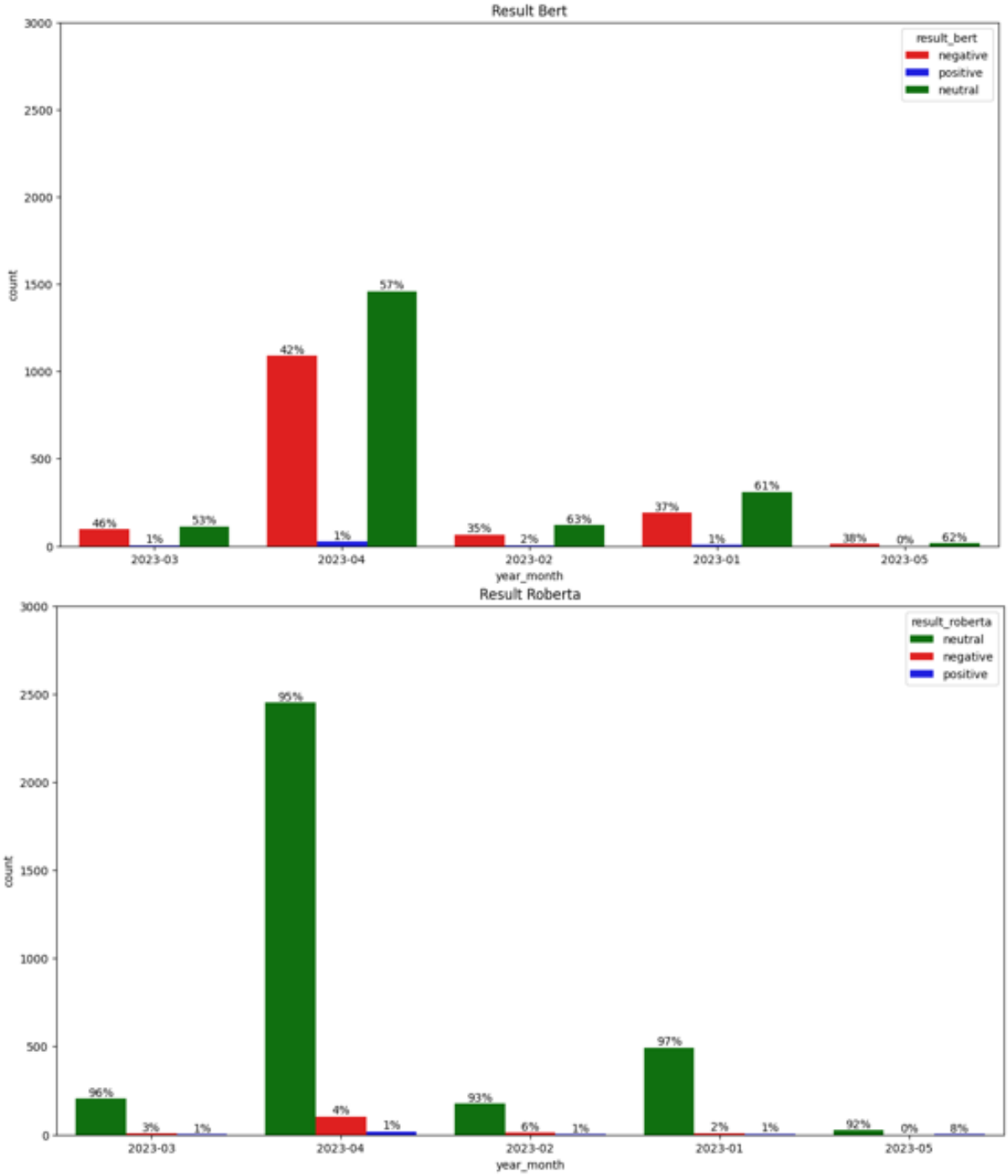

Figure 4.17: BERT and XLM-RoBERTa Sentiment plot for Time Frame 4 dataset

tions are of importance since it includes some good amount of negative predictions compared to XLM-RoBERTa which contains majority of neutral predictions.
Since these tweets are about nuclear energy (as fetched through the keywords), so people might have varying opinions but XLM-RoBERTa predicts almost 95-100% of the tweets as neutral in all time frames, which sounds very much unpractical.
The reason behind poor performance of XLM-RoBERTa could be that its not being trained on Twitter corpora (Barbieri, Anke, and Camacho-Collados 2021) and it does not perform good for noisy data compared to BERT as seen in simulation part for PotTS dataset which is much noisier compared to other datasets.

January month analysis: Out of given all months January month fetched my interest to do analysis of tweets and how sentiment evolved over period of time , since January is one of the peak winter months and for this month I have the data for all three years (2021, 2022 and 2023). As previously discussed XLM-RoBERTa predictions are mostly neutral, I am focusing only on predictions of BERT model and for both the models positive sentiments are almost negligible therefore in my discussion only about negative sentiments are highlighted.
As shown in figure 4.18, it can be seen that in year 2021 only 19% tweets were negative , which further went down to arount 14% in year 2022 and in 2023 amount of negative tweets increased to almost around 41%.

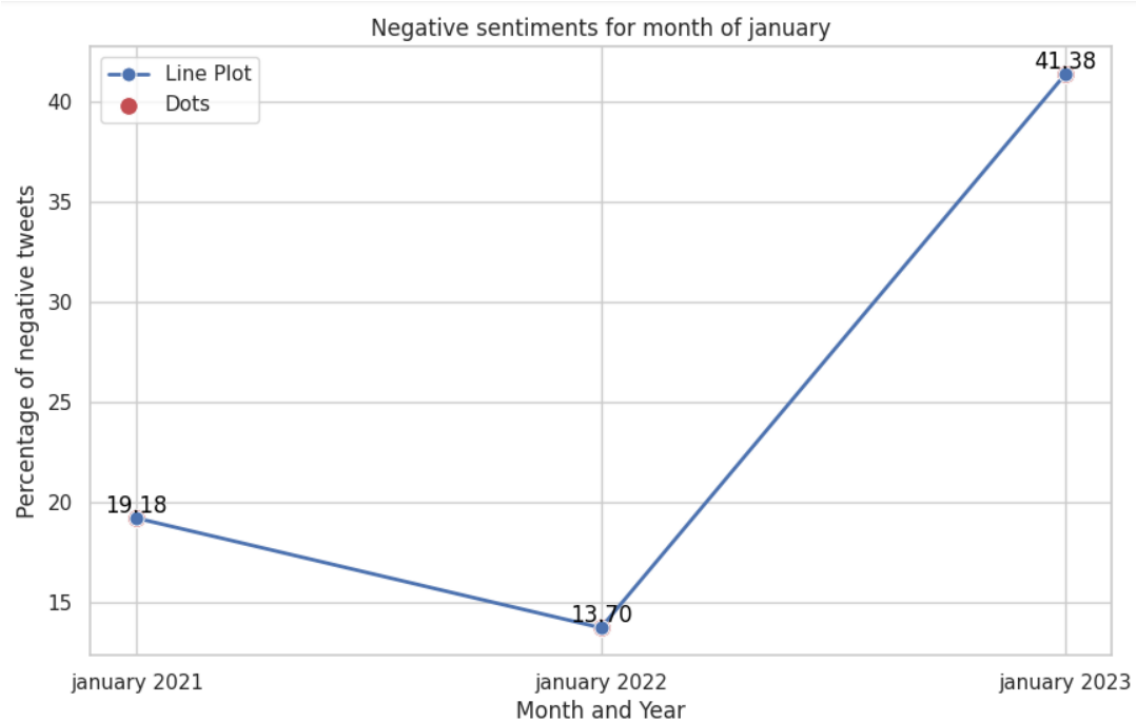

Figure 4.18: Negative sentiments for the month of January

Chapter 5

Conclusions

BERT is unquestionably a milestone in the application of Machine Learning to Natural Language Processing. Because it is simple to use and allows for quick fine-tuning, it is anticipated to have a wide range of practical uses in the future. This thesis has two parts testing and application of BERT and XLM-RoBERTa. For testing (different dataset) purpose my observation for both BERT and XLM-RoBERTa is important since results of both the models are good enough for comparison of sentiment analysis in German language. And for the application part for the Twitter dataset, I observe that XLM-RoBERTa predictions are mostly neutral which is of no interest to check for how the sentiments evolved around nuclear energy. Although BERT has relatively better amount of negative and positive predictions. Therefore I am considering only development of BERT model for my analysis for Twitter dataset.

In conclusion, this thesis delved into the application of the BERT method for German language sentiment classification, utilizing four distinct labeled datasets (GermEval-2017, Scare, SB10k, PotTS). The comparative analysis between BERT and XLM-RoBERTa revealed intriguing insights. XLM-RoBERTa demonstrated superior performance in two of the datasets (Scare and SB10k) , while BERT exhibited better accuracy in one dataset (PotTS). Interestingly, both models performed comparably same in GermEval-2017 dataset.

Moreover, the introduction of random spelling errors using the ***nlpaug*** library added a unique dimension to the evaluation. Across varying text lengths in the datasets, it was observed that the impact of spelling errors on accuracy differed. For dataset of shorter length (Scare), accuracy is decreasing to a greater extent upon increasing the spelling error percentage. Medium size datasets also gets impacted with spelling error but its results are inconsistent to the amount of spelling error, sometimes giving different percentages of spelling error result remain same as in SB10K dataset for both the models and in the case of PotTS dataset in which text size is also of medium length accuracy is decreased compared to cleaned text and almost same with different amount of spelling error in the case of BERT model but for XLM-RoBERTa model accuracy decreases slightly but consistently with increasing spelling error percentage. In the case of dataset (germeval2017) with large text sizes, hardly any change in accuracy is observed upon giving spelling error.

The exploration of the impact of spelling errors reveals a nuanced relationship

with text length, emphasizing the importance of understanding how these models respond to noise in diverse linguistic contexts. As text size decreases, the models' accuracies experience a more pronounced decline, shedding light on the challenges posed by shorter texts and increased error probabilities.

This research contributes valuable insights to the field of sentiment analysis, offering considerations for practitioners seeking to deploy BERT and XLM-RoBERTa in the complex landscape of the German language. The outcomes emphasize the importance of nuanced model evaluation, dataset characteristics, and contextual understanding for effective sentiment analysis applications.

Possible Future Work:

1. Experiment with alternative text augmentation techniques beyond spelling errors using random character swap to diversify the types of data perturbations, such as keyboard error, random character insertion, random character deletion, random character substitution etc. to enhance model robustness.

2. With more data fine-tuning of a sentiment model like BERT-sentiment would be possible on a specific task like sentiment analysis on tweets for the "nuclear energy" topic.

Articles References

Adel, Hadeer et al. (2022). "Improving crisis events detection using distilbert with hunger games search algorithm". In: *Mathematics* 10.3, p. 447.

AlBadani, Barakat, Ronghua Shi, and Jian Dong (2022). "A novel machine learning approach for sentiment analysis on Twitter incorporating the universal language model fine-tuning and SVM". In: *Applied System Innovation* 5.1, p. 13.

Barbieri, Francesco, Luis Espinosa Anke, and Jose Camacho-Collados (2021). "Xlm-t: Multilingual language models in twitter for sentiment analysis and beyond". In: *arXiv preprint arXiv:2104.12250*.

Cheruku, Ramalingaswamy et al. (2023). "Sentiment classification with modified RoBERTa and recurrent neural networks". In: *Multimedia Tools and Applications*, pp. 1–19.

Conneau, Alexis et al. (2019). "Unsupervised Cross-lingual Representation Learning at Scale". In: *CoRR* abs/1911.02116. arXiv: 1911.02116. URL: http://arxiv.org/abs/1911.02116.

Dang, Cach N, María N Moreno-García, and Fernando De la Prieta (2021). "Hybrid deep learning models for sentiment analysis". In: *Complexity* 2021, pp. 1–16.

Devika, M D[a], C[a] Sunitha, and Amal Ganesh (2016). "Sentiment analysis: a comparative study on different approaches". In: *Procedia Computer Science* 87, pp. 44–49.

Devlin, Jacob et al. (2018). "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". In: *CoRR* abs/1810.04805. arXiv: 1810.04805. URL: http://arxiv.org/abs/1810.04805.

Garrido-Merchan, Eduardo C, Roberto Gozalo-Brizuela, and Santiago Gonzalez-Carvajal (2023). "Comparing BERT against traditional machine learning models in text classification". In: *Journal of Computational and Cognitive Engineering* 2.4, pp. 352–356.

Gonen, Hila et al. (2020). "It's not Greek to mBERT: inducing word-level translations from multilingual BERT". In: *arXiv preprint arXiv:2010.08275*.

Guhr, O. et al. (2020). "Training a broad-coverage german sentiment classification model for dialog systems. In Proceedings of the Twelfth Language Resources and Evaluation Conference". In: pp. 1627–1632.

Han, Xu et al. (2021). "Pre-trained models: Past, present and future". In: *AI Open* 2, pp. 225–250.

Höfer, Antonia and Mina Mottahedin (2023). "Minanto at SemEval-2023 Task 2: Fine-tuning XLM-RoBERTa for Named Entity Recognition on English Data". In: pp. 1127–1130.

Joshi, Neha S and Suhasini A Itkat (2014). "A survey on feature level sentiment analysis". In: *International Journal of Computer Science and Information Technologies* 5.4, pp. 5422–5425.

Kotelnikova, Anastasia et al. (2021). "Lexicon-based methods vs. BERT for text sentiment analysis". In: pp. 71–83.

Libovickỳ, Jindřich, Rudolf Rosa, and Alexander Fraser (2019). "How language-neutral is multilingual BERT?" In: *arXiv preprint arXiv:1911.03310*.

Liu, Yinhan et al. (2019). "Roberta: A robustly optimized bert pretraining approach". In: *arXiv preprint arXiv:1907.11692*.

Lu, Helen et al. (2022). "Improved Binary Classification Accuracy in Natural Language Processing via Test-Time Augmentation". In: *MURJ*, p. 19.

Mohammad, Saif M (2017). "Challenges in sentiment analysis". In: *A practical guide to sentiment analysis*, pp. 61–83.

Niu, Shuteng et al. (2020). "A decade survey of transfer learning (2010–2020)". In: *IEEE Transactions on Artificial Intelligence* 1.2, pp. 151–166.

Pan, Sinno Jialin and Qiang Yang (2009). "A survey on transfer learning". In: *IEEE Transactions on knowledge and data engineering* 22.10, pp. 1345–1359.

Qasim, Rukhma et al. (2022). "A fine-tuned BERT-based transfer learning approach for text classification". In: *Journal of healthcare engineering* 2022.

Sanh, Victor et al. (2019). "DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter". In: *arXiv preprint arXiv:1910.01108*.

Serrano-Guerrero, Jesus et al. (2015). "Sentiment analysis: A review and comparative analysis of web services". In: *Information Sciences* 311, pp. 18–38.

Talaat, Amira Samy (2023). "Sentiment analysis classification system using hybrid BERT models". In: *Journal of Big Data* 10.1, pp. 1–18.

Tan, Kian Long, Chin Poo Lee, and Kian Ming Lim (2023). "RoBERTa-GRU: A Hybrid Deep Learning Model for Enhanced Sentiment Analysis". In: *Applied Sciences* 13.6, p. 3915.

Vani, S and TV Madhusudhana Rao (2019). "An experimental approach towards the performance assessment of various optimizers on convolutional neural network". In: pp. 331–336.

Vaswani, Ashish et al. (2017). "Attention is all you need". In: *Advances in neural information processing systems* 30.

Vinodhini, G and RM Chandrasekaran (2012). "Sentiment analysis and opinion mining: a survey". In: *International Journal* 2.6, pp. 282–292.

Wankhade, M., A.C.S. Rao, and C. Kulkarni ((2022)). "A survey on sentiment analysis methods, applications, and challenges. Artificial Intelligence Review". In: *Artificial Intelligence Review 2022 - Springer* 55.7, pp. 5731–5780.

Xu, Jian and Yuqing Zhai (2021). "A Toxic Comment Classification Model Based on Ensemble". In: 1873.1, p. 012080.

Yan, Hang, Xiaonan Li, and Xipeng Qiu (2020). "BERT for monolingual and cross-lingual reverse dictionary". In: *arXiv preprint arXiv:2009.14790*.

Yuan, JianHua et al. (2020). "Recent advances in deep learning based sentiment analysis". In: *Science China Technological Sciences* 63.10, pp. 1947–1970.

Other Sources References

Alammar, Jay (2018a). *The Illustrated BERT, ELMo, and co. (How NLP Cracked Transfer Learning)*. https://jalammar.github.io/illustrated-bert/. [Online; accessed 07-April-2023].

— (2018b). *The Illustrated-transformer*. https://jalammar.github.io/illustrated-transformer/. [Online; accessed 23-May-2023].

Castillo, Dianne (2021). *Transfer Learning for Machine Learning*. https://www.seldon.io/transfer-learning. [Online; accessed 30-November-2023].

Chan, Branden (2020). *XLM-RoBERTa: The alternative for non-english NLP*. https://medium.com/deepset-ai/xlm-roberta-the-multilingual-alternative-for-non-english-nlp-cf0b889ccbbf. [Online; accessed 07-April-2023].

Chauhan, Nagesh Singh (2022). *Google BERT: Understanding the Architecture*. https://www.theaidream.com/post/google-bert-understanding-the-architecture. [Online; accessed 25-May-2023].

Dolphin, Rian (2020). *LSTM Networks — A Detailed Explanation*. https://towardsdatascience.com/lstm-networks-a-detailed-explanation-8fae6aefc7f9. [Online; accessed 24-May-2023].

Efimov, Vyacheslav (2023). *Large Language Models: RoBERTa — A Robustly Optimized BERT Approach*. https://towardsdatascience.com/roberta-1ef07226c8d8. [Online; accessed 27-November-2023].

Emrich, Matin (2020). *Transfer Learning BERT Transformer Functionality*. https://www.alexanderthamm.com/en/blog/transfer-learning-bert-function/. [Online; accessed 3-December-2023].

Feng, Calvin (2023). *Vanilla Recurrent Neural Network*. https://calvinfeng.gitbook.io/machine-learning-notebook/supervised-learning/recurrent-neural-network/recurrent_neural_networks. [Online; accessed 24-May-2023].

Ghosh, Subhadip (2021). *Recurrent Neural Networks: In-depth understanding*. https://medium.com/almabetter/recurrent-neural-networks-in-depth-understanding-84d5a09b396f. [Online; accessed 29-December-2023].

Horev, Rani (2018). *BERT Explained: State of the art language model for NLP*. https://towardsdatascience.com/bert-explained-state-of-the-art-language-model-for-nlp-f8b21a9b6270. [Online; accessed 07-April-2023].

Kanstrén, Teemu (2020). *A Look at Precision, Recall, and F1-Score*. https://towardsdatascience.com/a-look-at-precision-recall-and-f1-score-36b5fd0dd3ec. [Online; accessed 4-October-2023].

Kumar, Raman (2021). *All You Need to know about BERT*. https://www.analyticsvidhya.com/blog/2021/05/all-you-need-to-know-about-bert/. [Online; accessed 26-May-2023].

McCormick, Chris (2020). *How to Apply BERT to Arabic and Other Languages*. https://towardsdatascience.com/how-to-apply-bert-to-arabic-and-other-languages-5c3410ddd787. [Online; accessed 27-November-2023].

Menon, Raghav (2021). *Transformers*. https://raghav-menon.medium.com/transformers-c4e040fc6500. [Online; accessed 22-May-2023].

Mohdsanadzakirizvi (2019). *Demystifying BERT: A Comprehensive Guide to the Groundbreaking NLP Framework*. https://www.analyticsvidhya.com/blog/2019/09/demystifying-bert-groundbreaking-nlp-framework/. [Online; accessed 26-May-2023].

Oinkina and Hakyll (2015). *Understanding LSTM Networks*. https://colah.github.io/posts/2015-08-Understanding-LSTMs/. [Online; accessed 24-May-2023].

Özateş, Muhammet Nusret (2021). *Transformer Architecture: How Transformer Models Work?* https://medium.com/carbon-consulting/transformer-architecture-how-transformer-models-work-46fc70b4ea59. [Online; accessed 07-April-2023].

Phi, Michael (2020). *Illustrated Guide to Transformers- Step by Step Explanation*. https://towardsdatascience.com/illustrated-guide-to-transformers-step-by-step-explanation-f74876522bc0. [Online; accessed 25-May-2023].

Shastri, Akash (2020). *3 neural network architectures you need to know for NLP*. https://towardsdatascience.com/3-neural-network-architectures-you-need-to-know-for-nlp-5660f11281be. [Online; accessed 07-April-2023].

Suleiman Khan, Ph.D. (2019). *BERT Technology introduced in 3-minutes*. https://towardsdatascience.com/bert-technology-introduced-in-3-minutes-2c2f9968268c. [Online; accessed 07-April-2023].

T, Adith Narein (n.d.). *Embeddings in BERT*. https://iq.opengenus.org/embeddings-in-bert/. [Online; accessed 26-December-2023].

Yılmaz, Begüm (2023). *Top 5 Sentiment Analysis Challenges and Solutions in 2023*. https://research.aimultiple.com/sentiment-analysis-challenges/. [Online; accessed 3-December-2023].

Appendix A

Appendix

Classification Report:

	precision	recall	f1-score	support
negative	0.43	0.59	0.50	46
neutral	0.78	0.77	0.78	126
positive	0.50	0.06	0.11	17
accuracy			0.66	189
macro avg	0.57	0.47	0.46	189
weighted avg	0.67	0.66	0.65	189

Figure A.1: BERT result for spelling error through Stemming in Germaneval2017 Dataset

Classification Report:

	precision	recall	f1-score	support
negative	0.44	0.09	0.15	46
neutral	0.68	0.96	0.79	126
positive	0.00	0.00	0.00	17
accuracy			0.66	189
macro avg	0.37	0.35	0.31	189
weighted avg	0.56	0.66	0.56	189

Figure A.2: XLM-RoBERTa result for spelling error through Stemming in Germaneval2017 Dataset

Classification Report:

	precision	recall	f1-score	support
Negative	0.41	0.74	0.53	78
Neutral	0.67	0.02	0.04	213
Positive	0.61	0.89	0.72	308
accuracy			0.56	599
macro avg	0.56	0.55	0.43	599
weighted avg	0.60	0.56	0.45	599

Figure A.3: BERT result for spelling error through Stemming in Scare Alarm Clocks Dataset

Classification Report:

	precision	recall	f1-score	support
negative	0.63	0.15	0.25	78
neutral	0.47	0.98	0.63	213
positive	0.95	0.41	0.57	308
accuracy			0.58	599
macro avg	0.68	0.51	0.48	599
weighted avg	0.74	0.58	0.55	599

Figure A.4: XLM-RoBERTa result for spelling error through Stemming in Scare Alarm Clocks Dataset

Classification Report:

	precision	recall	f1-score	support
Negative	0.46	0.84	0.59	86
Neutral	1.00	0.02	0.03	185
Positive	0.60	0.86	0.70	250
accuracy			0.56	521
macro avg	0.68	0.57	0.44	521
weighted avg	0.72	0.56	0.45	521

Figure A.5: BERT result for spelling error through Stemming in Scare Fitness Tracker Dataset

Classification Report:

	precision	recall	f1-score	support
negative	0.68	0.20	0.31	86
neutral	0.46	0.99	0.63	185
positive	0.96	0.36	0.53	250
accuracy			0.56	521
macro avg	0.70	0.52	0.49	521
weighted avg	0.73	0.56	0.53	521

Figure A.6: XLM-RoBERTa result for spelling error through Stemming in Scare Fitness Tracker Dataset

Classification Report:

	precision	recall	f1-score	support
negative	0.27	0.77	0.40	161
neutral	0.82	0.50	0.62	606
positive	0.52	0.38	0.44	233
accuracy			0.51	1000
macro avg	0.54	0.55	0.49	1000
weighted avg	0.66	0.51	0.54	1000

Figure A.7: BERT result for spelling error through Stemming in SB10K Dataset

Classification Report:

	precision	recall	f1-score	support
negative	0.67	0.20	0.31	161
neutral	0.65	0.97	0.78	606
positive	0.67	0.14	0.23	233
accuracy			0.65	1000
macro avg	0.66	0.44	0.44	1000
weighted avg	0.66	0.65	0.58	1000

Figure A.8: XLM-RoBERTa result for spelling error through Stemming in SB10K Dataset

Classification Report:

	precision	recall	f1-score	support
mixed	0.00	0.00	0.00	4
negative	0.26	0.70	0.38	23
neutral	0.36	0.38	0.37	26
positive	0.75	0.13	0.23	45
accuracy			0.33	98
macro avg	0.34	0.30	0.24	98
weighted avg	0.50	0.33	0.29	98

Figure A.9: BERT result for spelling error through Stemming in PotTS Dataset

Classification Report:

	precision	recall	f1-score	support
mixed	0.00	0.00	0.00	4
negative	1.00	0.09	0.16	23
neutral	0.27	1.00	0.43	26
positive	1.00	0.02	0.04	45
accuracy			0.30	98
macro avg	0.57	0.28	0.16	98
weighted avg	0.77	0.30	0.17	98

Figure A.10: XLM-RoBERTa result for spelling error through Stemming in PotTS Dataset