Master Thesis

Evaluating Historical Language Models for literary research

Maja Syrek

Supervisor	Antske Fokkens, Eleanor Smith
2nd reader	Hennie Van Der Vliet

a thesis submitted in fulfillment of the requirements for the degree of

MA Linguistics

(Text Mining)

Vrije Universiteit Amsterdam

Computational Linguistics and Text-Mining Lab
Department of Language and Communication
Faculty of Humanities

Date	August 15, 2025
Student number	2866223

Abstract

This thesis investigates the performance of transformer-based models, specifically BERT and TuringBERT. The models are evaluated on sentiment analysis task across modern and historical-literary corpora. The research focuses on token-level and sentence-level classification using two benchmarks: a dialogue-based Friends dataset and a historical Sherlock Holmes corpus. Token-level experiments revealed challenges with ambiguous and inconsistent annotations in the historical corpus. This lead the BERT model to fail at assigning positive labels. Sentence-level experiments demonstrated that while both models perform well on the modern corpus, their performance significantly declines on the historical text. Furthermore, BERT surprisingly outperforms TuringBERT. Error analysis indicates suggests that shared misclassifications likely stem from linguistic complexity and contextual dependencies. The difference in mistakes made indicates that the historical training of TuringBERT has an impact on interpretation of data. The findings, however are inconclusive, as the sizes of training sets significantly differ between the benchmarks, which might have lead to the poorer performance on the Sherlock Holmes corpus. Key limitations include include imbalanced corpus sizes, the restricted scope of the sentiment analysis task, and annotation inconsistencies. Future work should focus on incorporating additional NLP tasks, balancing the dataset sizes and expanding on the genres of texts used.

Declaration of Authorship

I, Maja Syrek, declare that this thesis, titled *Evaluating Historical Language Models for literary research* and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a MA Linguistics degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Date: 15.08.2025

Signed:

Acknowledgments

I would like to express my gratitude to my supervisors, Prof. Dr. Antske Fokkens and Eleanor Smith, for their unwavering support and guidance throughout my research. Their expertise and encouragement were invaluable for the completion of this thesis.

I am also grateful to Francis Bond for making the NTU-M corpus available and Luis Morgado da Costa for his guidance in navigating the data. Their help was instrumental to the research process.

Finally, I would like to thank everyone who supported me during the past year of the Master's program. I couldn't have completed the thesis without them.

List of Figures

List of Tables

Contents

Chapter 1

Introduction

1.1 Problem Definition

Studying a historical-literary text often poses a challenge to an expert researcher. Historical language can differ substantially from its modern counterpart in terms of vocabulary, spelling, syntax, and semantics. However, in the era of digitization this process, like many other academic ventures, entered an automated space. Now it lies in the peculiar nexus of textual analysis, the historical domain and NLP.

This transition enables large-scale analysis across multiple texts that was previously deemed impractical or impossible. Instead of one expert analyzing one document at a time, automated processing allows the researchers to cover years of literature and this way discover larger trends. This, however, requires specialized tools that can account for the diachronic language change. A regular language model, trained on contemporary corpora, may fail to interpret archaic forms, rare expressions, or domain-specific meanings. To handle historical material specialized language models have been developed and trained on historical corpora. The aim of such models like TuringBERT is to improve performance on tasks involving older linguistic forms.

A key question is how well these historical-domain models perform compared to their general-domain counterparts. This thesis addresses that question by evaluating a historical transformer-based model, TuringBERT, against its widely used general-domain counterpart, BERT. Both models are evaluated at the sentiment analysis task. By comparing them across sentence-level experiments, the study seeks to determine whether domain-specific training provides a tangible advantage when working with historical texts.

1.2 Research Question

This thesis seeks to establish how TuringBERT and BERT perform and compare in domain-specific sentiment analysis task. More specifically, the leading research question is:

Does TuringBERT outperform BERT on a sentiment analysis task when applied to a Sherlock Holmes stories corpus?

1.3 Approach

The study begins with identifying a gap in research of the domain-specific language models. A study by Manjavacas and Fonteyn (2022) provides an overview of some of the common historical models. The authors evaluate three differently trained models that were adapted using different historical datasets. Among them, TuringBERT is tested on a selection of five common NLP tasks. In this thesis, the same model is evaluated in a similar fashion to Manjavacas and Fonteyn (2022) in a sentiment analysis task.

Firstly, TuringBERT is compared to BERT on a token-level sentiment analysis task. Both models are trained three times with different random seeds, and their performance is evaluated using macro averaged F1-scores, Cohen's kappa coefficient, and standard deviation across runs. Nevertheless, BERT fails to detect the "POSITIVE" label in the token-level setting. Multiple strategies are tested to address this issue, proving unsuccessful.

The decision is made to perform the task at sentence-level. Both models are trained to assign a "POSITIVE", "NEGATIVE", or "NEUTRAL" label to whole sentences. TuringBERT and BERT are compared to determine whether the domain-specific training provides advantage when dealing with historical text. To further explore the models' abilities they are tested on a modern benchmark. Using the same protocol the models perform the sentiment analysis on a modern corpus to assess whether the historical one poses a bigger challenge.

1.4 Thesis Outline

The structure of this thesis is organized as follows. Chapter 2 reviews previous related work. It focuses on textual analysis, the historical-literary domain, and the challenges historical texts pose in digital humanities. The chapter reviews the most common NLP approaches in the domain and introduces the task of sentiment analysis. Chapter 3 introduces the methodology used: the models evaluated, and the data used to test them. Consequently, Chapter 4 and Chapter 5 present the findings of the token- and sentences-level experiments. Chapter 6 discusses the results and provides insights into the limitations of the study. Lastly, Chapter 7 concludes this thesis project.

Chapter 2

Related Work

This chapter provides the contextual background for the study. Section 2.1 introduces the historical-literary domain and textual analysis, both important aspects of this thesis. The section outlines the domain's relevance within the Digital Humanities field and highlights the motivations for its exploration, as well as describes popular approaches to traditional textual analysis. Section 2.2 focuses on NLP applications within the historical domain and outlines the linguistic and technical challenges associated with historical texts. The section provides an overview of common approaches used in the field, with particular attention to the performance of transformer-based models. It includes a detailed discussion of the study conducted by Manjavacas and Fonteyn (2022) on adapting the BERT model to historical language data. Their work forms a methodological foundation for this thesis, which seeks to further test their findings and address a gap in the evaluation of historically adapted models. Lastly, Section 2.3 discusses sentiment analysis and justified the selection of an appropriate modern benchmark, MELD, for comparative evaluation.

2.1 The Historical-Literary Domain

"The past is a foreign country, they do things differently there" LP Hartley

The new field of digital humanities (DH) has been introduced during the era of digitization (Piotrowski, 2012). As computational research has become more popular, vast collections of books and other texts have incrementally been made available in digital form. The digitized historical material presented an opportunity for convergence of NLP and DH. The NLP tasks that have been applied to modern texts up until that point, now could be administered to previously unavailable datasets.

2.1.1 Motivations for Exploring Historical Literary Texts

The reasons for exploring the historical literary domain are various. For one, analysis of old texts can serve as a basis for diachronic language changes. An example of such study can be Sun and Wang's (2022) paper on 19th century English fiction. Authors notice that a common complaint about the Victorian era literature is difficulty in comprehending the text. One of the reasons for that challenge is archaic vocabulary. Sun and Wang (2022) pose the question of whether words in English fiction have become more abstract or more concrete since the 19th century. To assess that change, authors measure lexical concreteness and imageability. Concreteness indicates whether a word is more concrete or abstract, while imageability reflects the degree of effort required to generate a mental image of something. However, concreteness or abstractness of a word is not the only linguistic change that happens over time. There are a number of studies on other evolutionary indicators. Among some common phenomena are: word frequencies (Feltgen et al. 2017), that one may measure in order to assess vocabulary use over time, and semantic shift (Periti et al., 2025), how words change meaning due to social practices or events.

Apart from investigating diachronic language change, historical research allows researchers to assess the political or societal context of the time period a text represents. As Faudree and Pharao Hansen (2014) state, language, society and history create a nexus, which should be approached from a perspective that considers them all. The authors emphasize the mutual influence of the elements. Considering those dependencies, analysis of historical texts can reveal societal changes and political narratives, ultimately becoming fundamental in understanding the underlying social dynamic of a given time period (Szabó et al., 2020). Finally, a common NLP task is authorship attribution, which involves determining who wrote a given text when the author is unknown. As information about authors can be derived from their writing style (Silva et al., 2023), authorship attribution has become popular in fields like digital forensics or social media analysis. However, more importantly it is an essential part of literary research. The socio-linguistic characteristics of the century, language-specific attributes, genre and topics undertaken by the author, are all crucial in defining a literary style (Silva et al., 2023). Some of the frequent notions in the historical-literary domain are: identifying anonymous or disputed texts (Fung, 2003; Tuccinardi, 2017) and resolving doubted authorship (Fox & Ehmoda, 2012).

2.1.2 Approaches to Traditional Textual Analysis

Before the era of computerization, traditional textual analysis was usually done by the means of a “close read” of texts from original sources (Hills & Miani, 2025). This human reader approach includes a researcher personally reading an original source and conducting a qualitative analysis. The benefit of such a strategy is a detailed investigation done by a professional. However, this method also presents numerous limitations. Such human based studies introduce a subjective bias, they are limited by what is physically viable to read and remember, as well as such projects are costly and difficult to replicate. Fortunately, digitized databases and the development of NLP allow for a quantitative analysis, also referred to as a “distant read”. As Hills and Mani (2025) state, a large statistical analysis using computer-assisted experimentation allows researchers to analyze more documents than any human reader could. This approach is capable of evaluating multiple authors and numerous texts, all over the span of decades. Even though the scale of such analysis lacks a nuanced interpretation of a close read, it presents a number of advantages. The benefits include the large size of processed data, capturing large-scale trends, and a fast and replicable process.

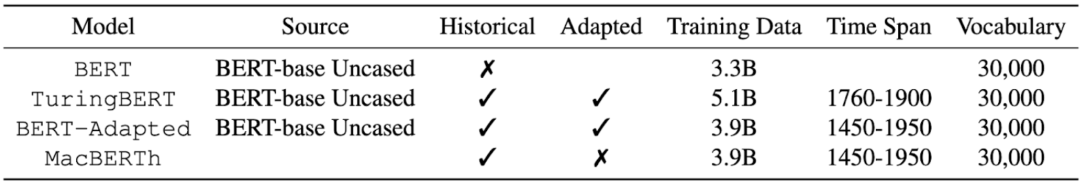
Recently, researchers have advocated to move beyond the limitations of the close and distant read, towards a combined method that would exploit aspects of both approaches. An example of such study could be Handzic and Mulavdić’s (2022) work. Their research revolves around a 19th century Bosnian novel “Zeleno busenje” (Green Turf) by Edhem Mulabdic. The aim is to identify the main characters and analyze their behavior. The process of analysis starts with annotating the characters and creating a node for each of them. In accordance with the distant approach, co-occurrences of multiple characters on the same page are computationally identified and counted, capturing a variety of types of interactions. Then, Gephi software is used to create a social network that can be divided into cohesive subgroups based on the uncovered main characters, their ties, strength and associations. This process is followed by a close read of pages where the subgroups were found, which allows for a closer investigation of the nature of the characters and their relations. As a result, Handzic and Mulavdić (2022) are able to identify a group of interconnected characters. After human reading of the passages, they learn that the characters are members of two families and liberals and conservatives, representatives of the societal structure at the time. The authors demonstrate that their proposed combined reading approach provides support in interpretation of literary work by providing context often missing in a distant read. It also directs researcher's attention to features that require deeper analysis, thereby improving efficiency and effectiveness.

2.2 NLP Applications in the Domain

The initial motivation for this thesis stemmed from personal interest in the Sherlock Holmes stories. As the project developed, the corpus (described in more detail in Section 3.2.1) took more of a central role and now serves as the historical-literary benchmark for evaluating the models. The choice of a literary text presents a number of opportunities, particularly in observing how sentiment is embedded within a narrative structure. It is worth considering how sentiment manifests itself in the story, through descriptions, character construction, or dynamics of relationships between them. However, working with historical texts also presents multiple linguistic and technical challenges. As previously mentioned, cases of diachronic changes in languages like semantic shift or changes in frequency use must be taken into account when treating historical texts. For NLP in the historical domain, the challenges include such simple properties as orthography. Firstly, English spelling has gone through major changes over the years in the process of standardization (Nevalainen, 2006). However, orthography variation is not only differences in spelling of individual words like *bok* instead of *book*, but also issues like punctuation, hyphenation, abbreviation etc. (Piotrowski, 2012). Many modern statistical methods rely on the assumption that parameters learned through training data (a tagged corpus) are applicable to other new data. As the starting point for NLP are the surface forms, spelling of words becomes a significant parameter. For example, if a model is tasked with Part-Of-Speech (POS) tagging it might rely on the suffix *-ion* to identify a noun form. However, if that suffix is not present or subject to orthographic variation in historical data, the system will not make the connection.

To address the challenges posed by historical language data, the NLP community has attempted different approaches. In general, the pattern observed across the broader NLP field also holds within the historical domain: transformer based models prove to outperform traditional ML-approaches like Naive Bayes or Support Vector Machines (Schmidt et al., 2021). In particular, different variants of BERT adapted to the domain seem to be the leading method of analyzing historical texts (Al-Laith et al., 2024; Schmidt et al., 2021; Silva et al., 2023). An attempt at evaluating such historical language models has been undertaken by Manjavacas and Fonteyn (2022). While there is a variety of model architectures available for research in the historical domain, the authors decided to base their study on BERT. The motivation for the transformer based model is that in comparison to other NLP approaches, BERT is well-established and thoroughly studied. This makes it a strong candidate for domain adaptation, as its architecture and training behavior are well understood. As well as its performance has been extensively benchmarked across a variety of NLP tasks. As such, by using BERT researchers are able to focus specifically on impact of historical pretraining, since the model's performance is well-documented on standard NLP tasks and provides a stable baseline for comparison.

Manjavacas and Fonteyn (2022) evaluate three historical models. The first one MacBERTh, originally introduced by Manjavacas and Fonteyn (2021), was developed using the same paradigm as a regular BERT, with the one difference that the pre-training phase was done entirely on historical data dating between 1450 and 1950. The remaining two models are variants of the BERT-Base Uncased model which are then adapted by further pre-training on historical English data. This means that the modern data used in the original process of pre-traing a BERT model is expanded by a collection of historical data. For the first model, TuringBERT, that additional historical data is a set of literary works published between 1760 and 1900 (Hosseini et al., 2021). The second "adapted" model, to which the authors refer to as BERT-Adapted, was further pre-trained on the same historical dataset used in developing MacBERTh. All three historical models were then compared with a BERT model, trained on present-day English, that served as a state-of-the-art benchmark. Figure 2.1 shows an overview of evaluated models in Manjavacas and Fonteyn (2022).



Model	Source	Historical	Adapted	Training Data	Time Span	Vocabulary
BERT	BERT-base Uncased	✗		3.3B		30,000
TuringBERT	BERT-base Uncased	✓	✓	5.1B	1760-1900	30,000
BERT-Adapted	BERT-base Uncased	✓	✓	3.9B	1450-1950	30,000
MacBERTh		✓	✗	3.9B	1450-1950	30,000

Figure 2.1: Overview of all the models involved in the present experiments.
Note: Reprinted from Manjavacas and Fonteyn (2022), (Table 1., p. 5)

To assess the effectiveness of alternative approaches in designing historical models, Manjavacas and Fonteyn (2022) employ the four models at a variety of tasks: tagging, Named Entity Recognition (NER), Word Sense Disambiguation, Fill-In-The-Blank, and Sentence Periodization. Each task is performed on a suitable historical dataset within the 1450-1950 period. The experiments are consistent with previous results in the field; pre-training a BERT model from scratch on a historical corpus is the most suitable form of adapting a model to the historical domain. The MacBERTh model has an advantage over the competitor models on all tasks with the exception of NER. On Named Entity Recognition TuringBERT performed slightly better. However, the authors speculate that it might be due to an overlap between the evaluation data and the data used for adapting the model (Manjavacas & Fonteyn, 2022). Finally, all three historically-adapted models showed advantage over a present-day BERT.

Manjavacas and Fonteyn's (2022) work provides some of the latest insights into adapting models for the historical domain. They show that adaptation is a viable way of creating more specialized models for the DH field. To further test their findings and expand on the topic, this thesis takes inspiration from their approach. The Sherlock Holmes corpus served as the starting point for designing this project, and its characteristics shaped the methodological choices that followed. As such, the choice of the historical model was dictated by its compatibility with said corpus. The TuringBERT model was chosen for a number of reasons. Firstly, out of the models evaluated by Manjavacas and Fonteyn (2022), TuringBERT covers the smallest time frame of 1760-1900. Since the Sherlock Corpus was originally published between 1892 and 1903, a smaller time frame of TuringBERT is more representative of the English variety at that time, and could therefore mean more accurate results in evaluation. Secondly, Manjavacas and Fonteyn (2022) showed that TuringBERT outperformed other models on the NER (present-day BERT). To further explore if the model is capable of effective performance on another task, for the purpose of this thesis TuringBERT is tested on Sentiment Analysis (SA).

2.3 Sentiment Analysis and Choice of a Benchmark

Manjavacas and Fonteyn's (2022) work proved that historically adapted models outperform their modern counterparts. However, the basis of their evaluation was limited to five NLP tasks. The experiments of this thesis contribute to filling that assessment gap. Here, TuringBERT is tested on a sentiment analysis task. SA was originally introduced as a response to the rapid development of Internet spaces that include comments, reviews and other means of expressing opinions. Sentiment analysis, also called opinion mining, is designed to automatically extract and analyze sentiment from text (Mao et al., 2024). Even though a vast majority of its applications are connected to e-commerce (like product ratings), movie reviews or social media analysis (like Tweets evaluation), SA proves to be also useful in computational literary studies. In their overview of the field, Kim and Klinger (2022) list multiple applications, such as: genre and story-type classification, emotion modeling in historical texts, or character network analysis.

Mao et al. (2024) provide a systematic literature review on sentiment analysis in NLP. The authors dedicate a section of their paper to assess the most commonly used databases for SA (shown in Table 2.1. Their analysis shows that various types of reviews are the prevalent domain.

Dataset	**Domain**
Tweets SemEval (Rosenthal et al., 2017)	Movie reviews, Tweets
Stanford Sentiment Treebank (SST) (Socher et al., 2013)	Movie reviews
Yelp datasets (Asghar et al., 2014)	Restaurants, shopping, hotels, travel reviews
Stanford large movie review (IMBD) (Maas et al., 2011)	Movie reviews

Table 2.1: Most commonly used datasets in SA.
Note: Adapted from Mao et al. (2024) (Table 6. p.10)

The homogeneity of the available databases presents a challenge in terms of choosing an appropriate benchmark for this thesis. The models selected for the experiments will be compared on their performance on a historical text and a modern text. This present-day benchmark will act as a reference point. For a regular BERT model that would mean that its performance on a historical dataset is predicted to be weaker to that on the modern benchmark. The pre-training process of such a model was done exclusively on contemporary texts, therefore the learned representations reflect the modern reality. This means that when presented with a historical text, BERT will rely on modern interpretations of words (that may be subject to change over time) and possibly misinterpret the context. What is more, because BERT relies on the WordPiece tokenizer that uses modern vocabulary, it may disrupt the process of meaningful tokenization when applied to an old text (Nayak et al., 2020); see also Section 3.1.1 for further explanation of the tokenization process. Respectively, TuringBERT is predicted to outperform the BERT model on the historical dataset due to its pre-training on a diachronic (historically diverse) corpus.

In order to facilitate a comparison between the models, the modern benchmark should complement the domain of the historical corpus as much as is feasible. Ideally, this means aligning both benchmarks by genre, so that the models are evaluated under similar conditions and face comparable challenges. Since a collection of Sherlock Holmes stories serves as the historical benchmark, the modern equivalent should, in principle, be a contemporary piece of literature. However, a search for a suitable benchmark revealed that there are no widely used databases available within the same genre. A review of commonly used benchmarks, such as the overview by Mao et al. (2024), showed that the predominant domains are reviews, which offer little genre diversity. Several alternatives were investigated.

Ultimately, the Multimodal Emotion-Lines dataset (MELD) (Poria et al., 2018) was chosen. MELD, which will be further discussed in more detail in Section 3.2.2, consists of over 14,000 labeled utterances sourced from the tv show *Friends* and is possibly the closest domain match for the Sherlock Holmes stories. The utterances form a coherent narration following scripted scenes which resembles the literary narrative of Arthur Conan Doyle stories. In contrast, the frequent Tweet corpora usually offer short, one-sentence statements that lack the broader context. What is more, the dialogue based nature of MELD further corresponds with the narration style. Both datasets present character-oriented conventions that allow for multiple complex sentiments and their interactions. Contrarily, review based benchmarks usually offer monologic and author-centric views. Finally, the register of MELD, a semi-formal, edited, structured, and grammatically sound language, is more in line with literary prose. Tweets, on the contrary, often contain highly informal tone, abbreviations, misspellings and other metadata. Thus, despite not being an exact genre match, MELD is the most suitable option for the modern benchmark.

Chapter 3

Methodology

This thesis compares performance of two models on a sentiment analysis task. Both models are applied on a historical literary benchmark and a modern benchmark. This chapter outlines the methodology implemented for this thesis. Three sections are included. Section 3.1 describes in detail the models chosen for evaluation. Section 3.2 covers the data used in this project along their data statements and details of preprocessing. Finally, Section 3.3 explains the general approach taken.

3.1 Models

The following sections describe the chosen models, BERT and TuringBERT in more detail.

3.1.1 BERT

BERT, which stands for Bidirectional Encoder Representations from Transformers, was introduced by Devlin et al. (2019). The model is an encoder, meaning it is based on the transformer architecture designed to understand relationships between words in a language. However, it is not capable of producing any text on its own, like decoder models do.

In order to make BERT handle a variety of tasks, the authors designed the input representation as a token sequence. In other words, a single sentence or a pair of sentences is split into tokens. The tokenization is done using WordPiece embeddings (Wu et al., 2016), that includes a 30,000 token vocabulary. This means that words compiled of more frequent morphemes or letter clusters are often split into subword tokens, e.g., “scalable” → “scala”, “##ble”. The double hash marks indicate that a subword token is a part of a larger word. This method allows BERT to keep a relatively small vocabulary while still enabling the model to handle unfamiliar or compound words. Moreover, to accommodate multiple sentences in one sequence, special tokens were introduced. [CLS], a token that marks the beginning of a new sequence, and [SEP], a token that separates two sentences in one sequence. As a result, an input sequence could look as follows: Question: “Where is Eiffel Tower located?”, Answer: “Eiffel Tower is located in Paris, France” → [‘[CLS]’, ‘where’, ‘is’, ‘eiffel,’, tower’, ‘located’, ‘?’, ‘[SEP]’, ‘eiffel,’, ‘tower’, ‘is’, ‘located’, ‘in’, ‘paris, ‘,’, ‘france’, ‘[SEP]’.

For each token, its representation is a sum of three embeddings (as presented in Figure 3.1): the corresponding token embedding, a segment embedding (that differentiates between sentence A and sentence B), and a position embedding (that indicates word order). This method encodes information not only about what the token is, but also where it is situated and in which context. Each token is then transformed into a vector through multiple layers of encoders. This process allows the model to refine the representation of the token based on the surrounding context provided by other tokens in the sequence.

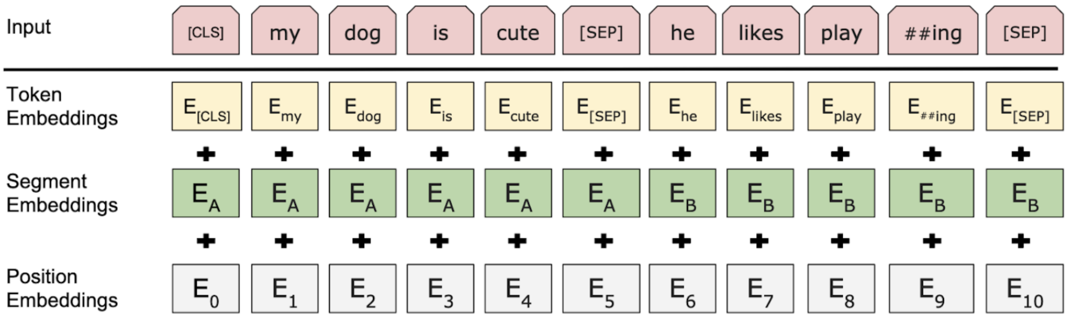

Input
[CLS]
my
dog
is
cute
[SEP]
he
likes
play
##ing
[SEP]
Token Embeddings
$E_{[CLS]}$ E_{my} E_{dog} E_{is} E_{cute} $E_{[SEP]}$ E_{he} E_{likes} E_{play} $E_{\#\#ing}$ $E_{[SEP]}$
Segment Embeddings
E_A E_A E_A E_A E_A E_A E_B E_B E_B E_B E_B
Position Embeddings
E_0 E_1 E_2 E_3 E_4 E_5 E_6 E_7 E_8 E_9 E_{10}

Figure 3.1: BERT input representation.
Note: Reprinted from Devlin et al. (2019) (Figure 2., p. 4175)

Devlin et al. (2019) introduce a two-step framework in which they implement BERT: pre-training and fine-tuning. During pre-training, the model is trained on large unlabeled data over two self-supervised tasks. The first task is *masked language modeling* MLM. The authors mask 15% of the tokens in an input sequence at random, and the model is tasked at predicting the missing token. This approach requires the model to consider both the tokens preceding and the tokens following the missing token, in order to grasp the full sequence context. The second task, *next sentence prediction* NSP, aims at understanding the relationship between two sentences. In order to train the model to grasp sentence relations, the authors train BERT on predicting whether in a pair of sentences A and B, sentence B actually follows A, or whether it is another random sentence from the corpus.

By combining the two tasks, and pre-training on a large corpus, ca. 3.3B tokens, i.e. the BookCorpus (Zhu et al., 2015) and English Wikipedia, BERT attains robust general-purpose representations. After which, it is ready for fine-tuning to a specific task. The BERT model is initialized with the pre-trained parameters, which are then adapted when it is deployed on a smaller task-specific dataset.

The BERT model in question comes in a variety of sizes, languages and other features. For the purpose of this study, the “BERT-base uncased” was chosen. It is a smaller of the two originally released models (12-layer, 768-hidden, 12-heads, 110M parameters), trained on modern English data, that does not differentiate between upper and lower cases. The original model is available at `https://github.com/google-research/bert` and will be henceforth referred to as *BERT*.

3.1.2 Turing BERT

In their paper, Hosseini et al. (2021) aimed to facilitate historicization of NLP methods by releasing various models on a 19th-century book collection. To do so, the authors designed a historical training corpus, a collection of circa 48,000 digitized Books published between 1760 and 1900. As the authors stated the specific genres included in the collection are not clearly defined, however an attempt at computing those numbers has been made by the Living with Machines Team (n.d.). About 65% of the books are nonfiction and the remaining 35% are works of fiction.

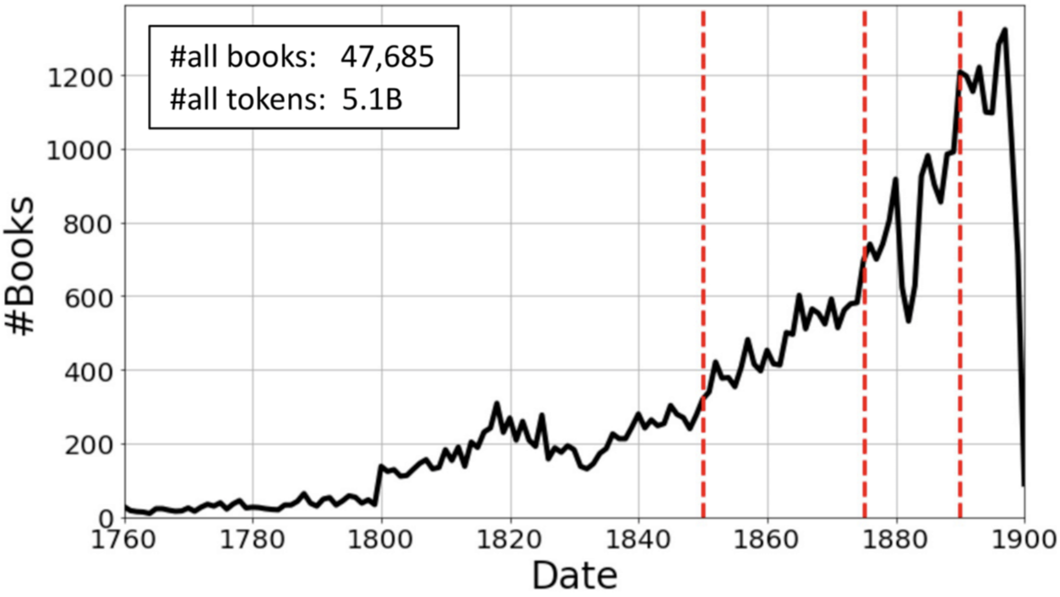


Figure 3.2: Number of books by publication date.
Note: Reprinted from Hosseini et al., 2021 (Figure 1., p. 2)

The data was minimally normalized, by removing punctuation or fixing common punctuation errors. Even though the corpus compiled mostly of English books, there was a substantial number of works in other languages. Books in languages other than English were filtered out. Finally, the books were split into sentences and tokenized, resulting in a total of 5.1B tokens. Among other models, Hosseini et al. (2021) historically adapted the BERT-base uncased model. The pre-training stage of BERT was extended to the historical corpus. Tokenized and lowercased sentences were fed to the model tasked with MLM. The resulting model, trained at the Alan Turing Institute, will be referred to as *TuringBERT* and can be found in the original repository `https://zenodo.org/records/4782245`.

3.2 Data

The following section describes the two datasets used for model evaluation. Table 3.1 presents relevant information following the data statement guidelines (Bender & Friedman, 2018). The subsequent sections provide detailed descriptions of the datasets' structure and any preprocessing performed.

Data Statement	**Sherlock Holmes corpus**	**Friends corpus**
Language Variety	British English representative of late 19th/early 20th century.	American English representative of the mid-1990s to early 2000s.
Speaker Demographic	The main character is Sherlock Holmes, a 50/60-year-old British detective. The stories are mainly narrated by Doctor Watson, a middle-aged British man.	The main characters form a group of six friends—three men and three women in their twenties.
Annotator Demographic	Sentence-level dataset: one linguistically trained native English speaker. Token-level dataset: multiple annotators proficient in English.	Three graduate students proficient in English speaking and writing.
Text Characteristics	Stories published between 1891 and 1903. Set in London at the turn of the 20th century. A collection of detective fiction stories.	A sitcom set in New York City around the early 2000s. Dialogue-based (TV script).

Table 3.1: Benchmarks' data statement comparison

3.2.1 The Historical Literary Benchmark

The historical literary benchmark (also referred to as Sherlock Holmes corpus) comprises a collection of Arthur Conan Doyle's stories. As Table 3.2 shows, there are seven stories total, written in English, and originally published between 1891 and 1903. The shortest story comprises 411 sentences and the longest 3824 sentences.

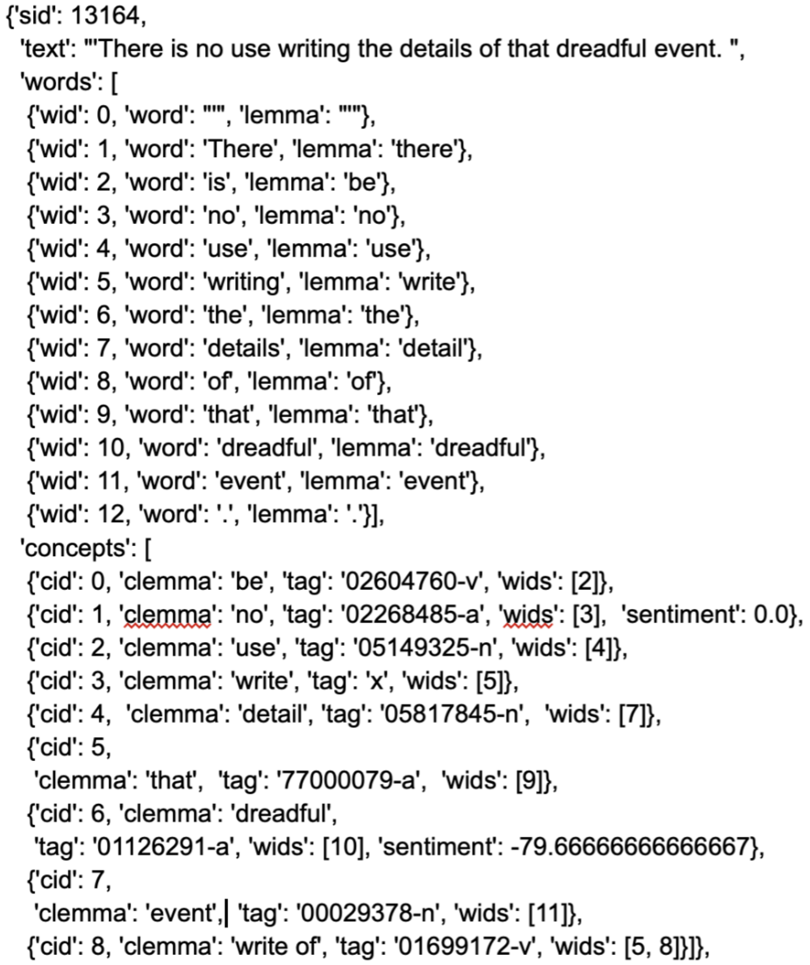
Title	**Date Published**	**Number of Sentences**
"The Red-Headed League"	1891	553
"A Scandal in Bohemia"	1891	661
"The Adventure of the Speckled Band"	1892	599
"The Adventure of the Naval Treaty"	1893	827
"The Adventure of the Final Problem"	1893	411
"The Hound of the Baskervilles"	1902	3824
"The Adventure of the Dancing Men"	1903	600

Table 3.2: Historical literary benchmark contents

Data Structure

The Sherlock Holmes corpus used in this project is a part of the NTU-multilingual corpus (NTU-MC) (Tan & Bond, 2012). The corpus, made available by Francis Bond, was originally designed to build word-disambiguation datasets (specifically an expanded version of the Princeton Wordnet 3.0 (Bond et al., 2016)). Word-disambiguation is a task of assigning a specific sense or meaning to each word. Thus, each sentence in the corpus was parsed looking for matches with synsets from the WordNet. The synsets are also informally known as "concepts".

The simplified structure of an example sentence from the corpus looks as shown in Figure 3.3. Each sentence is assigned an id "sid", which is followed by the sentence text and then two keys: "words" and "concepts". "Words" is a list of the tokenized words each with an id and its lemma. "Concepts" is a list of word meanings found in that sentence, each has its id, lemma, a tag (which refers to a concept in the Wordnet), and a span of word ids to which it refers. To further refine the NTU corpus in its word meaning annotation, the authors decided to mark sentiment (Bond et al., 2016). As different senses have different sentiment values, this was found to be quite useful in the word-disambiguation task. The sentiment annotation was originally done on concept level using a scale of -100 to 100 (from most negative to most positive). Importantly, operators such as "not" or "very" were not included. Annotators used an online environment that allowed them to choose certain points on the scale and then adjust. Hence, an example sentence like the one in Figure 3.3, consists of 11 tokens (excluding interpunction), but only seven concepts (notice that the third concept was wrongly marked as the "x" in the "tag" suggests). Out of those seven concepts only one, "dreadful", was marked for sentiment.



```
{'sid': 13164,
 'text': '"There is no use writing the details of that dreadful event. ",
 'words': [
  {'wid': 0, 'word': '"', 'lemma': '"'},
  {'wid': 1, 'word': 'There', 'lemma': 'there'},
  {'wid': 2, 'word': 'is', 'lemma': 'be'},
  {'wid': 3, 'word': 'no', 'lemma': 'no'},
  {'wid': 4, 'word': 'use', 'lemma': 'use'},
  {'wid': 5, 'word': 'writing', 'lemma': 'write'},
  {'wid': 6, 'word': 'the', 'lemma': 'the'},
  {'wid': 7, 'word': 'details', 'lemma': 'detail'},
  {'wid': 8, 'word': 'of', 'lemma': 'of'},
  {'wid': 9, 'word': 'that', 'lemma': 'that'},
  {'wid': 10, 'word': 'dreadful', 'lemma': 'dreadful'},
  {'wid': 11, 'word': 'event', 'lemma': 'event'},
  {'wid': 12, 'word': '.', 'lemma': '.'}],
 'concepts': [
  {'cid': 0, 'clemma': 'be', 'tag': '02604760-v', 'wids': [2]},
  {'cid': 1, 'clemma': 'no', 'tag': '02268485-a', 'wids': [3],  'sentiment': 0.0},
  {'cid': 2, 'clemma': 'use', 'tag': '05149325-n', 'wids': [4]},
  {'cid': 3, 'clemma': 'write', 'tag': 'x', 'wids': [5]},
  {'cid': 4,  'clemma': 'detail', 'tag': '05817845-n',  'wids': [7]},
  {'cid': 5,
   'clemma': 'that',  'tag': '77000079-a',  'wids': [9]},
  {'cid': 6, 'clemma': 'dreadful',
   'tag': '01126291-a', 'wids': [10], 'sentiment': -79.66666666666667},
  {'cid': 7,
   'clemma': 'event', 'tag': '00029378-n', 'wids': [11]},
  {'cid': 8, 'clemma': 'write of', 'tag': '01699172-v', 'wids': [5, 8]}]},
```

Figure 3.3: Token-level dataset structure

Preprocessing

For the token-level experiment of this thesis, the Sherlock Holmes corpus was split into a training and testing set. The testing set was the “The hound of the Baskervilles” story comprising 3824 sentences, or 70360 tokens. The training set was built from the remaining six stories and a total of 3651 sentences, 68241 tokens. To further simplify the dataset, it was preprocessed. Because the input sequence for both models should be a sentence, where each token should have an assigned value, the tokens and concepts were matched on the basis of word ids. Using the concepts, each token was assigned a sentiment value. If a token did not have a corresponding concept, or the concept did not have a sentiment annotation, it was assigned a default of sentiment 0. The sentiment annotation was then simplified from number values to string labels according to the following schema: values of -100 to -20 were assigned “NEGATIVE”, values between -20 and 20 were assigned “NEUTRAL”, values of 20 to 100 were assigned “POSITIVE”. The neutral interval of -20 to 20 was chosen upon data exploration which showed that it included words only slightly or not at all evaluative (like “sir” with a score of 12 or “occupy” with a score of 11.3).

Apart from the concept level annotation, the Sherlock Holmes corpus was also annotated on "chunk" level (Bond et al., 2016). Chunks refer to multiple tokens forming coherent phrases. In a similar JSON format to that of the token-level files, each sentence has a chunk that spans over all of its tokens, allowing for a sentence-, or utterance-level sentiment analysis. Unfortunately, only one of the stories was available in the sentence-level format, "The adventure of the speckled band". Because of its small size, only 599 sentences, a 10-cross fold validation was set up allowing for efficient use of a limited size data set. The story was annotated by one annotator with linguistic training, on the same -100 to 100 scale as the concept level annotation. In order to simplify the data, as well as match it to the modern benchmark, the number values of sentiment have been translated to string labels "POSITIVE", "NEUTRAL", and "NEGATIVE". This time, the sentiment labels have been assigned following a different schema: "NEGATIVE" for values from -100 to 0, "NEUTRAL" for 0, "POSITIVE" for values from 0 to 100. This choice followed a data exploration which did not show ambiguous cases within a +-10 interval.

There was also an attempt at creating sentence-level datasets from the available token-level data. However, the translation from concept to sentence proved to be unsuccessful. To compare both structures, the token-level sentiment of the "The adventure of the speckled band" story was summed for each sentence. Out of the 599 sentences, only 111 were an exact match (what is meant by an exact match are the same sentiment values, e.g., positive 73), but 99 out of those were neutral (i.e., values of 0). What is more, out of that number, 95 were true neutrals (where on the token level all tokens are 0). Given that the sentiment scores range continuously from -100 to 100, an exact match between summed token-level and sentence-level sentiment scores was unlikely. Therefore, non-exact matches were explored by allowing a margin of difference within specified intervals. Three intervals were examined: 10, 15, and 20. Each interval allowed for a difference - for example, 10 - in sentiment score between the summed token level and sentence level. Table 3.3 shows how many matches were found (excluding the neutral 95 true neutral cases).

Interval	**Matches**
10	74
15	96
20	122

Table 3.3: Sentiment matches between token-level and sentence-level sentiment scores in "The adventure of the speckled band" story

Moreover, an average sentiment difference between sum token and sentence level was calculated (excluding the exact 111 matches). For the remaining sentences within the story, the difference is on average 52.2, suggesting high disparities between sentiment represented by the two levels. Finally, after preprocessing the number values to string labels "POSITIVE", "NEUTRAL", and "NEGATIVE", the number of matches found is 387 and mismatches is 212. Due to significant differences between summed token level and sentence level sentiment, the experiment proceeded with the 10-cross fold validation setup of the available "The adventure of the speckled band" story.

3.2.2 Modern Benchmark

The Multimodal Emotion-Lines (MELD) (Poria et al., 2018) has been chosen as the modern benchmark for this thesis. The dataset evolved from the EmotionLines dataset originally introduced by Chen et al. (2018). EmotionLines contains dialogues from the tv-show *Friends*, where each dialogue contains utterances from more than two speakers. The dataset was created by crawling the scripts of the sitcom's Seasons 1 through 9. Each scene in an episode was considered a dialogue. 10,000 dialogues were randomly sampled. The chosen dialogues were annotated on the utterance level for one of Ekman's (1987) six basic emotions: anger, disgust, fear, happiness, sadness, surprise and an additional neutral. The total dataset comprised 14,503 labeled utterances. Poria et al. (2018) extended, improved and further developed the EmotionLines dataset for multimodal purposes by contributing visual and audio counterparts of the utterances. The authors reassessed the original corpus and removed a number of dialogues that spanned over multiple episodes (and as such created duplicates) or did not receive coherent labels from the annotators. The remaining utterances were re-annotated for sentiment values (“POSITIVE”, “NEUTRAL”, “NEGATIVE”), which is the reason for choosing the MELD dataset over EmotionLines. A training set of size 9,989 utterances and a testing set of 2,610 utterances were downloaded from the MELD depository (`https://affective-meld.github.io/`). The modern benchmark will be also referred to as the Friends corpus.

3.3 General Approach and Intuition

The goal of this thesis is to evaluate a historical language model, TuringBERT. The basis of the evaluation is sentiment analysis task executed on a historical literary dataset, the Sherlock Holmes corpus. In order to assess TuringBERT's capabilities, the model's performance is compared against two benchmarks: first, BERT's performance on the same corpus; and second, TuringBERT's performance on a modern benchmark.

The first experiment, sentiment analysis, is done on the token level. However, the results show that BERT did not assign any “POSITIVE” labels. The inability to successfully complete the task, combined with the absence of suitable modern token-level benchmarks, motivated the decision to conduct a second experiment, using a different approach. The second time, the sentiment analysis task was performed on a sentence level instead of the token level. TuringBERT and BERT were trained and evaluated on the Sherlock Holmes corpus. Both performed the same task on a modern benchmark, the Friends corpus. For each experiment, the respective models are trained on three different seeds to account for randomness in initialization.

The hypothesis is that TuringBERT will outperform BERT on the Sherlock Holmes corpus. Since TuringBERT was trained on a historical dataset, it is expected to have an advantage over the modern-trained BERT. Moreover, TuringBERT's unique training dataset - which combines the modern data used for BERT with a historical dataset - suggests that it should perform similarly on both modern and historical data.

Chapter 4

Results

This chapter describes the results of experiments run as a part of this thesis. The first section provides details of evaluation metrics used. The explanations are based on Jurafsky and Martin (2009) unless specified otherwise. The other two sections focus on the results of the token-level and sentence-level experiments respectively.

4.1 Evaluation Metrics

The outcomes of multi-label classification tasks, like the sentiment analysis of this thesis, can be transformed into binary classification. The results can be divided into True Positives (TP), False Positives (FP), True Negatives (TN), and False Negatives (FN). The following metrics are a common standard in tasks predicting multiple labels, as they emphasize true positives: the findings that are of importance.

Precision measures the percentage of the items that the system found, that were in fact positive. In other words, how many of the identified labels are labeled correctly. It can be calculated by

$$P = \frac{TP}{TP + FP}$$

Recall measures the percentage of correctly identified items out of all the ones present in the data. Simply put, out of all possible correct labels, how many did the system find? It can be calculated as

$$P = \frac{TP}{TP + FN}$$

To optimize the results further, the **F1-score** can be calculated. It is the harmonic mean of precision and recall, defined as

$$P = \frac{2PR}{P + R}$$

In cases of multi-label classification tasks, especially ones with a varying number of labels, it is important to consider how to average the scores across the different classes. In the sentiment analysis tasks presented here, the categories are imbalanced. In the token-level experiment for the historical benchmark, the ”NEUTRAL” class dominates the dataset, with 67,799 instances out of 70,360 (about 96% of the tokens). In the same corpus, in the sentence-level experiments, the ”NEGATIVE” class occurs most often:

277 out of 599 sentences in the corpus are negative. Due to such significant representations of labels in the data, precision, recall and F1-score will be macro averaged. This method means computing the metric for each class and then averaging over the classes. The approach is favorable when dealing with imbalanced datasets, as it assigns equal weight to each class, regardless of its frequency.

Another measure implemented in this thesis is the **Kappa coefficient** κ, also known as Cohen's kappa, a metric that measures the agreement between two raters of the same task. As Grandini et al. (2020) explain, Cohen's kappa proves to be a useful tool when assessing the performance of multiple models undertaking the same task. In the case of this thesis, two models are compared, BERT and TuringBERT. Both are trained three times on different seeds in order to account for randomness in initialization. The Kappa coefficient helps to assess the stability of a model across the multiple seeds. It can be defined as

$$\kappa = \frac{P(A) - P(E)}{1 - P(E)}$$

where P(A) is the observed agreement between the raters and P(E) is the hypothetical agreement between them that could occur by chance. When there is no agreement other than the one that could happen by chance, κ is 0. If there is total agreement, κ is 1. It is often mentioned that a score above 0.8 proves good reliability.

Finally, **standard deviation** is used as another measure of model stability. Calculating the standard deviation of the results, like the F1-score, provides a mean score across the seeds and a reading on the model's reliability. The following formula is used to compute it:

$$\sigma = \sqrt{\frac{\sum (x_i - \mu)^2}{N}}$$

where x_1 stands for each individual data point in the set, μ is the mean, and N is the total number of data points.

4.2 Token-Level Experiments

The initial approach to evaluating the BERT and TuringBERT models is based on a token-level sentiment analysis. Both models are applied using three different seeds, and the full classification report for each run is recorded in Appendix A. TuringBERT provides satisfactory results; the recorded macro averaged F1-scores are 0.669, 0.672, and 0.524. The mean result is therefore 0.670 with a standard deviation of about 0.004. What is more, the kappa scores have been calculated between the models. The kappa values shown in Table 4.5 are all above 0.8, indicating good reliability. The kappa scores, combined with a low standard deviation, illustrate that the models provide similar results and are in overall agreement.

Seeds	**Kappa**
23 and 4	0.8859
23 and 95	0.8696
4 and 95	0.8594

Table 4.1: Comparison of Kappa scores for different seed pairs of the TuringBERT model for token level.

While TuringBERT provides promising results, BERT fails at the task. It does not assign any "POSITIVE" labels. Table 4.2 shows a classification report for one of the seeds of the BERT model. While the categories "NEGATIVE" and "NEUTRAL" are picked up by the model, the "POSITIVE" label receives zero precision and recall. This means that the model does not assign any positive labels to the test data set. It does so consistently across the three seeds.

	Precision	Recall	F1-Score	Support
NEGATIVE	0.3996	0.8634	0.5464	1508
NEUTRAL	0.9830	0.9729	0.9779	67799
POSITIVE	0.0000	0.0000	0.0000	1053

Table 4.2: BERT classification report for seed 4 on token-level sentiment analysis

The irregularity of no "POSITIVE" labels, prevents a comparison between BERT and TuringBERT. Thus, in the hope of continuing with the experiments, the BERT model is examined for errors. Firstly, the same model was tested on a selection of sentences from the training dataset. Three example sentences that included positive-labeled tokens were manually chosen. The model failed to mark the tokens as "POSITIVE". The inability to perform the task on training data that was already seen by the model implies a fundamental issue with the setup.

A possible explanation could be underfitting, as the relatively small number of "POSITIVE" labels in the training data might not be enough for the model to generalize. To further explore the potential problem, the model was trained once more on an adjusted training dataset. This time, only sentences with strong positive sentiment were chosen to develop BERT. The model was trained on filtered examples that included four or more positive labels. This means that the model was only exposed to sentences that included positive tokens. No sentences that did not include a positive token were used. This strategy aimed to tackle the underfitting problem. BERT was then tested on 10% of the original test dataset to expedite the process. This new training approach did not prove to be successful. The model failed to assign any positive labels.

The final approach to exploring the failure was to adjust the labeling system. The labels were simplified to "POSITIVE" and "NON-POSITIVE". The model was trained on the newly labeled corpus, and further tested on a random sample of that same dataset. It once again failed by not assigning any positive labels.

4.3 Sentence-Level Experiments

After TuringBERT's failure at the token-level experiment, a new approach was taken. The models were used to perform sentiment analysis on the sentence level. Both models were applied using three different seeds, the full classification report for each run is recorded in Appendix B and Appendix C, for the historical-literary benchmark and the modern benchmark respectively.

4.3.1 Results: Modern Benchmark

BERT

On the modern benchmark both models provide satisfactory results. For the BERT model the recorded macro averaged F1-scores are 0.676, 0.687, and 0.676. The mean result is therefore 0.680 with a standard deviation of a 0.005. Moreover, the kappa scores were calculated for the models. The values are shown in Table 4.3. All the scores are above 0.8, indicating good reliability.

Seeds	**Kappa**
23 and 4	0.874
23 and 95	0.874
4 and 95	0.871

Table 4.3: Comparison of Kappa scores for different seed pairs of the BERT model for the Friends corpus on sentence level.

The kappa scores, combined with a low standard deviation illustrate that the different seeds of the BERT model provide similar results. Since the models trained on different seeds are in overall agreement, the best-performing model was selected for a more detailed discussion and error analysis in Chapter 5. Table 4.4 shows the classification report for the highest-scoring BERT model.

	Precision	Recall	F1-Score	Support
NEGATIVE	0.6610	0.6483	0.6545	833
NEUTRAL	0.7731	0.7866	0.7798	1256
POSITIVE	0.6291	0.6219	0.6255	521
Accuracy			0.7096	2610
Macro Avg	0.6877	0.6856	0.6866	2610
Weighted Avg	0.7086	0.7096	0.7090	2610

Table 4.4: Classification report for BERT seed 4 on sentence-level experiment on the Friends corpus.

The "NEUTRAL" class dominates the Friends corpus with 1256 instances. As could be expected, it also has the highest F1-score of all classes, i.e., 0.780. The other two labels are less common; "NEGATIVE" occurs 833 times in the dataset, and "POSITIVE" occurs 521 times. However, both classes score similar F1-scores of 0.655 and 0.623 respectively.

TuringBERT

In comparison to BERT, TuringBERT performs slightly worse. The recorded macro averaged F1-scores are 0.671, 0.667, and 0.666. The mean result is therefore 0.666 with a standard deviation of a 0.004. What is more, Table 4.5 shows the calculated kappa scores for the models. Similarly to BERT, all values are above 0.8, indicating good reliability.

Seeds	**Kappa**
23 and 4	0.858
23 and 95	0.844
4 and 95	0.834

Table 4.5: Comparison of Kappa scores for different seed pairs of the TuringBERT model for the Sherlock corpus on sentence level.

The kappa scores, combined with a low standard deviation suggest that the TuringBERT model, similarly to BERT, provides similar results no matter the different training seed. Again, the best-performing model was selected for a detailed discussion. Table 4.6 shows the classification report for the highest-scoring Turing BERT model.

	Precision	Recall	F1-Score	Support
NEGATIVE	0.6469	0.6026	0.6240	833
NEUTRAL	0.7676	0.7970	0.7820	1256
POSITIVE	0.6019	0.6123	0.6070	521
Accuracy			0.6981	2610
Macro Avg	0.6721	0.6706	0.6710	2610
Weighted Avg	0.6960	0.6981	0.6967	2610

Table 4.6: Classification report for TuringBERT seed 23 on sentence-level experiment on the Friends corpus.

The trends among the classes are the same as in the case of BERT, i.e., "NEUTRAL" dominates with the F1-score of 0.782, followed by "NEGATIVE" with 0.624 and "POSITIVE" with an F1-score of 0.607. While the model generally performs slightly worse (the macro F1-score of 0.671 compared to BERT's 0.687), it does score higher on the "NEUTRAL". The other less frequent classes score lower.

4.3.2 Results: Historical-Literary Benchmark

BERT

For BERT the recorded macro averaged F1-scores are 0.493, 0.535, and 0.413. The mean result is thus 0.480 with a standard deviation of a 0.051. These preliminary results shows that BERT performs worse on the historical-literary benchmark than it does on a modern one. What is more, the kappa scores were calculated. The values are shown in Table 4.7. The kappa scores range from 0.263 to 0.382. These values are drastically lower to those of the modern benchmark. What is more, the low scores indicate little agreement between the outputs of the model. This means that the models often assigned different labels and made different types of errors, even though they attain similar F1-scores.

Despite the low kappa scores, the best-performing model was still used for deeper analysis to maintain methodological consistency. Table 4.8 shows the classification report for the highest-scoring model. The "NEGATIVE" class dominates the dataset with 277 instances. As could be expected it also scores the highest on F1 with 0.715.

Seeds	**Kappa**
23 and 4	0.382
23 and 95	0.342
4 and 95	0.263

Table 4.7: Comparison of Kappa scores for different seed pairs of the BERT model for the Sherlock corpus on sentence level.

The other two classes, comparable in size, achieve similar F1-scores: 0.458 for the ”NEUTRAL” label and 0.433 for ”POSITIVE”.

	Precision	Recall	F1-Score	Support
NEGATIVE	0.6606	0.7798	0.7152	277
NEUTRAL	0.4921	0.4276	0.4576	145
POSITIVE	0.4795	0.3955	0.4334	177
Accuracy			0.5810	599
Macro Avg	0.5440	0.5343	0.5354	599
Weighted Avg	0.5663	0.5810	0.5696	599

Table 4.8: Classification report for BERT seed 4 on sentence-level experiment on the Sherlock Holmes corpus.

TuringBERT

The TuringBERT model achieved macro averaged F1-scores of 0.426, 0.385, and 0.443. The mean result is therefore 0.418 with a standard deviation of a 0.024. The average F1-score is comparable to that of BERT, but slightly lower. The standard deviation is smaller than that of BERT. What is more, the kappa scores were calculated. Table 4.9 shows the values. Similarly to BERT, the kappa scores are significantly lower for the historical-literary benchmark in comparison to those for the modern benchmark. The values vary between 0.208 and 0.379, which are comparable to BERT.

Seeds	**Kappa**
23 and 4	0.320
23 and 95	0.379
4 and 95	0.208

Table 4.9: Comparison of Kappa scores for different seed pairs of the TuringBERT model for the Sherlock corpus on sentence level.

The best-performing model’s classification is shown in Table 4.10. The ”NEGATIVE” class achieves the highest F1-score of 0.684. This is slightly lower than BERT’s 0.715. The ”POSITIVE” class scored 0.423, similarly to BERT. Surprisingly, the ”NEUTRAL” class scored significantly lower than BERT, at 0.221. It does so consistently across the three different seeds (see Tables B.1-3 in Appendix B).

	Precision	Recall	F1-score	Support
NEGATIVE	0.5396	0.9350	0.6843	277
NEUTRAL	0.5556	0.1379	0.2210	145
POSITIVE	0.6627	0.3107	0.4231	177
Accuracy			0.5576	599
Macro avg	0.5859	0.4612	0.4428	599
Weighted avg	0.5798	0.5576	0.4949	599

Table 4.10: Classification report for TuringBERT seed 95 on sentence-level experiment on the Sherlock Holmes corpus.

Chapter 5

Error Analysis

The evaluated models were run using three different random seeds. Following the convention, the best-performing seeds are the basis of the error analysis. The analysis is composed of quantitative and qualitative part. In the quantitative section the confusion matrices which illustrate alignment between gold labels and predicted labels are examined. The analysis is structured by the benchmark and frequency distribution of the classes in the test set, starting with the most frequent class and progressing to the least frequent one. The qualitative part aims to identify and summarize the underlying reasons for classification errors.

5.1 Quantitative analysis

5.1.1 Quantitative analysis: TuringBERT

On the Historical-Literary Benchmark

The first point of the results exploration is the significantly lower F1-scores of the "NEUTRAL" class of TuringBERT and the model's overall performance. When tested on the Sherlock Holmes corpus, the model performs worse in comparison to other classes (a score of 0.221 compared to 0.423 of the "POSITIVE" class). Figure 5.1 shows the confusion matrix for that model.

The "NEGATIVE" class was the most frequently occurring label in the dataset, with 277 instances. TuringBERT made 239 classification errors, with 221 FP and 18 FN instances. The model most frequently confused the "NEGATIVE" label with "POSITIVE" (ten times) and "NEUTRAL" (eight times), as illustrated by the examples below.

- "I cannot as yet see any connection." - misclassified as "POSITIVE"
- "Dr. Roylott then abandoned his attempts to establish himself in practice in London and took us to live with him in the old ancestral house at Stoke Moran." - misclassified as "POSITIVE"
- "'Indeed, Doctor,' said Holmes blandly." - misclassified as "NEUTRAL"
- "'How about poison?'" - misclassified as "NEUTRAL"

The second most frequent class was "POSITIVE", with 177 instances in the test set. TuringBERT made 150 classification errors, with 28 FP and 122 FN instances.

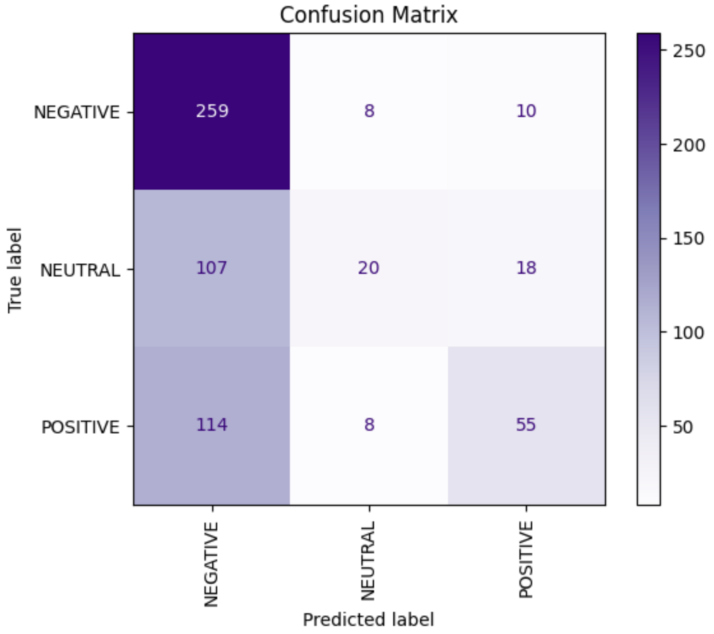


Figure 5.1: Confusion matrix of seed 95 TuringBERT model on the Sherlock Holmes corpus.

The model most frequently confused the "POSITIVE" label with "NEGATIVE" (114 times) and then "NEUTRAL" (eight times), as illustrated by the examples below.

- "'Good-morning, madam,' said Holmes cheerily." - misclassified as "NEUTRAL"
- "By no means." - misclassified as "NEUTRAL"
- "It was a perfect day, with a bright sun and a few fleecy clouds in the heavens." - misclassified as "NEGATIVE"
- "'He seems a very amiable person,' said Holmes, laughing." - misclassified as "NEGATIVE"

The last class was "NEUTRAL", with 145 instances in the test set. The model made 141 classification errors, with 16 FP and 125 FN instances. TuringBERT most frequently confused the "NEUTRAL" label with "NEGATIVE" (107 times), and then "POSITIVE" (18 times), as illustrated by the examples below.

- "'Do I make myself plain?'" - misclassified as "NEGATIVE"
- "Holmes had brought up a long thin cane, and this he placed upon the bed beside him." - misclassified as "NEGATIVE"
- "'I have heard of you before.'" - misclassified as "POSITIVE"
- "'Ah!'" - misclassified as "POSITIVE"

On the Modern Benchmark

For comparison against a modern benchmark, Figure 5.2 shows the confusion matrix for TuringBERT.

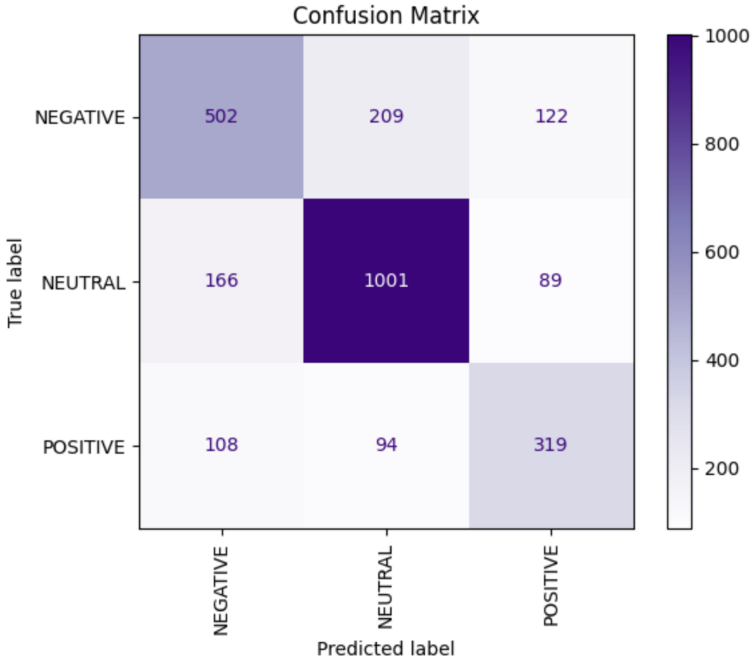


Figure 5.2: Confusion matrix of seed 23 TuringBERT model on the Friends corpus.

The "NEUTRAL" class was the most frequently occurring label in the dataset, with 1256 instances. TuringBERT made 558 classification errors, with 303 FP and 255 FN instances. The model most frequently confused the "NEUTRAL" label with "NEGATIVE" (166 times) and "POSITIVE" (89 times), as illustrated by the examples below.

- "'And it's not fake, it's totally brutal.'" - misclassified as "NEGATIVE"
- "'It's kind of embarrassing.'" - misclassified as "NEGATIVE"
- "'Yeah. I really like his glasses.'" - misclassified as "POSITIVE"
- "'Oh, thank you that's very helpful, I'm glad you came over.'" - misclassified as "POSITIVE"

The second most frequent class was "NEGATIVE", with 833 instances in the test set. TuringBERT made 605 classification errors, with 274 FP and 331 FN instances. The model most frequently confused the "NEGATIVE" label with "NEUTRAL" (209 times) and then "POSITIVE" (122 times), as illustrated by the examples below.

- "'Okay, fine, whatever. Welcome to the building.'" - misclassified as "NEUTRAL"
- "'The three losers.'" - misclassified as "NEUTRAL"
- "'Oh, Bob, he was nothing compared to you. I had to bite my lip to keep from screaming your name.'" - misclassified as "POSITIVE"

- ”’I didn’t think I could ever love again.’” - misclassified as ”POSITIVE”

The last class was ”POSITIVE”, with 521 instances in the test set. The model made 413 classification errors, with 211 FP and 202 FN instances. TuringBERT most frequently confused the ”NEUTRAL” label with ”NEGATIVE” (108 times), and then ”NEUTRAL” (94 times), as illustrated by the examples below.

- ”’It kicked! I think the baby kicked!’” - misclassified as ”NEGATIVE”
- ”’I can’t belive it, I’m gonna be an aunt. I’m gonna have like a nephew.’” - misclassified as ”NEGATIVE”
- ”’Why do all you’re coffee mugs have numbers on the bottom?’” - misclassified as ”NEUTRAL”
- ”’Love is the best medicine.’” - misclassified as ”NEUTRAL”

5.1.2 Quantitative analysis: BERT

On the Historical-Literary Benchmark

BERT outperformed TuringBERT on both benchmarks. This section will further analyze the confusion matrices. Figure 5.3 shows the confusion matrix for BERT tested on the Sherlock Holmes corpus.

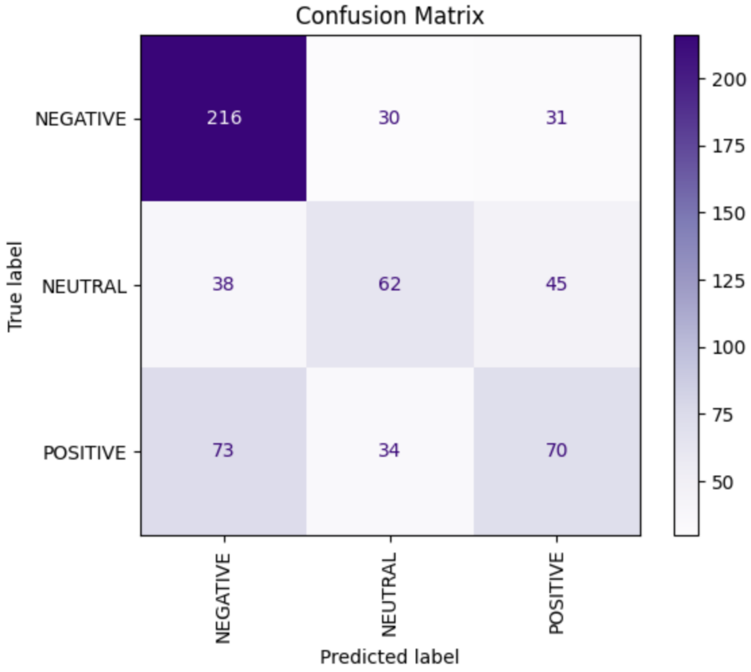


Figure 5.3: Confusion matrix of seed 4 BERT model on the Sherlock Holmes corpus.

The most frequent label was ”NEGATIVE”. BERT made 172 classification errors, with 111 FP and 61 FN instances. The model confused the ”NEGATIVE” label with ”POSITIVE” (31 times) and ”NEUTRAL” (30 times), as illustrated by the examples below.

- "'You have been cruelly used,' said Holmes." - misclassified as "POSITIVE"
- "Dark enough and sinister enough." - misclassified as "POSITIVE"
- "'Why, it's a dummy,' said he." - misclassified as "NEUTRAL"
- "'Certainly not.'" - misclassified as "NEUTRAL"

The second most frequent class was "POSITIVE". BERT made 239 classification errors, with 221 FP and 107 FN instances. The model confused the "POSITIVE" label with "NEGATIVE" (73 times) and "NEUTRAL" (34 times), as illustrated by the examples below.

- "'Well, it is of no great consequence, at any rate.'" - misclassified as "NEGATIVE"
- "My companion noiselessly closed the shutters, moved the lamp onto the table, and cast his eyes round the room." - misclassified as "NEGATIVE"
- "'This is my intimate friend and associate, Dr. Watson, before whom you can speak as freely as before myself.'" - misclassified as "NEUTRAL"
- "'Ah, yes, of course!'" - misclassified as "NEUTRAL"

The last class was "NEUTRAL". BERT made 147 classification errors, with 64 FP and 83 FN instances. The model confused the "NEUTRAL" label with "NEGATIVE" (38 times) and "POSITIVE" (45 times), as illustrated by the examples below.

- "'Do I make myself plain?'" - misclassified as "NEGATIVE"
- "'That is the baboon'." - misclassified as "NEGATIVE"
- "Holmes nodded his head." - misclassified as "POSITIVE"
- "'Now, would you have the kindness to go into your room and bar your shutters?'" - misclassified as "POSITIVE"

On the Modern Benchmark

Figure 5.4 shows the confusion matrix for BERT.

The "NEUTRAL" class was the most frequently occurring label in the dataset. BERT made 558 classification errors, with 290 FP and 268 FN instances. The model confused the "NEUTRAL" label with "NEGATIVE" (182 times) and "POSITIVE" (86 times), as illustrated by the examples below.

- "'It was just so awkward and bumpy.'" - misclassified as "NEGATIVE"
- "'Actually it's, it's quite, y'know, typical behaviour when you have this kind of dysfunctional group dynamic.'" - misclassified as "NEGATIVE"
- "'Great. He's doing great. Don't you worry about Chandler.'" - misclassified as "POSITIVE"
- "'I'm sorry, but I just wrote the best dance song for your wedding. Check this out.'" - misclassified as "POSITIVE"

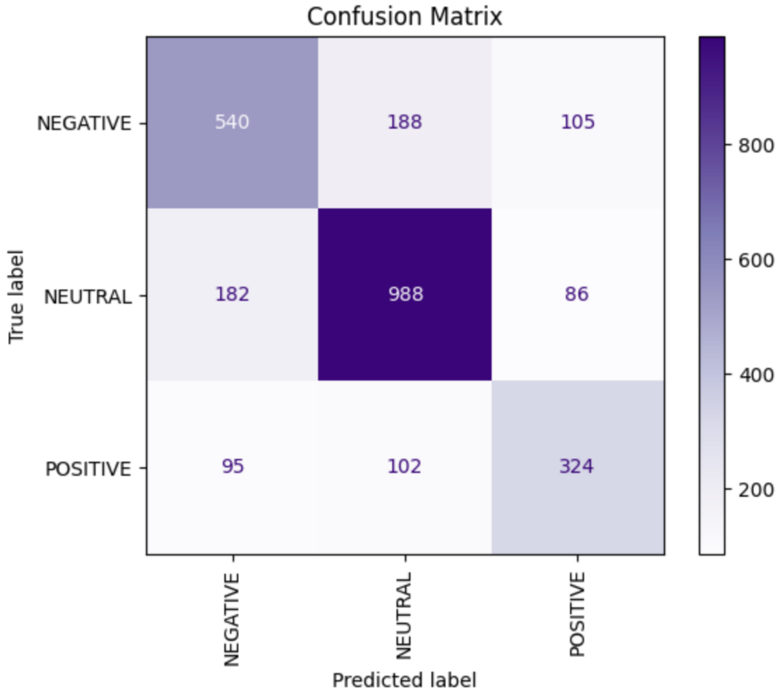


Figure 5.4: Confusion matrix of seed 23 BERT model on the Friends corpus.

The second most frequent class was ”NEGATIVE”. BERT made 570 classification errors, with 277 FP and 293 FN instances. The model confused the ”NEGATIVE” label with ”NEUTRAL” (188 times) and ”POSITIVE” (105 times), as illustrated by the examples below.

- ”’You had to do it, didn’t you? You couldn’t just leave it alone.’” - misclassified as ”NEUTRAL”

- ”’Four percent. Okay. I tip more than that when there’s a bug in my food.’” - misclassified as ”NEUTRAL”

- ”’Rachel! I’m never gonna think it’s okay for you to cheat on your husband!’” - misclassified as ”POSITIVE”

- ”’Y’know I don’t understand why they didn’t cry. It was a beautiful speech.’” - misclassified as ”POSITIVE”

The last class was ”POSITIVE”. BERT made 388 classification errors, with 191 FP and 197 FN instances. The model confused the ”POSITIVE” label with ”NEGATIVE” (95 times) and ”NEUTRAL” (102 times), as illustrated by the examples below.

- ”’That’s laughter.’” - misclassified as ”NEUTRAL”

- ”’They published my paper.’” - misclassified as ”NEUTRAL”

- ”’That is so silly.’” - misclassified as ”NEGATIVE”

- ”’No, but I want...I want the pinecones!’” - misclassified as ”NEGATIVE”

5.2 Qualitative Analysis

5.2.1 Qualitative Analysis: Historical-Literary Benchmark

The mistakes done by the models are often the same. Table 5.1 shows the number of unique mistakes each model made on the Sherlock Holmes corpus, as well as the number of shared mistakes. Out of the misclassifications of the "POSITIVE" class, 86 of them were shared by both models. This means that they made the same mistakes. The number of shared mistake is lower for the "NEUTRAL" class (74 mistakes), and it is the smallest for the "NEGATIVE" class (11 mistakes).

	"POSITIVE"	**"NEGATIVE"**	**"NEUTRAL"**
shared	86	11	74
BERT	21	50	9
TuringBERT	36	7	51

Table 5.1: Number of unique and shared mistakes made by the models on the Sherlock Holmes corpus (misclassifications per class).

Within their respective unique mistakes, both models often misclassify similar types of sentences. For the "POSITIVE" class this means overlooking phrases easily interpretable as positive due to vocabulary or exclamation marks, as illustrated by the examples below.

- "His smile broadened." - misclassified by TuringBERT as "NEGATIVE"
- "It was a perfect day, with a bright sun and a few fleecy clouds in the heavens." - misclassified by TuringBERT as "NEGATIVE"
- "'My dear Holmes!'" - misclassified by BERT as "NEGATIVE"
- "'Excellent.'" - misclassified by BERT as "NEGATIVE"

The mentions of smiling, highly positive vocabulary like "excellent" or "dear", exclamation marks, and vivid descriptions should have been clear indicators of positive sentiment that the models should have picked up.

Similarly for the "NEUTRAL" class, both models exhibit a substantial overlap in their misclassifications. Here, however, BERT performs better, with only nine unique mistakes compared to 51 made by TuringBERT. Often the model classifies sentences that do not contain any clear indicators of positive or negative sentiment, as the examples below show. In those cases, TuringBERT often incorrectly assigns the "NEGATIVE" label.

- "'What do you make of that, Watson?'" - misclassified by TuringBERT as "NEGATIVE"
- "He took up a small saucer of milk which stood on the top of it." - misclassified by TuringBERT as "NEGATIVE"

The models share the least mistakes within the "NEGATIVE" class (only 11 mistakes). Here TuringBERT performs better, as it only makes seven additional errors, while BERT makes 50 unique mistakes. While some of the sentences are arguably ambiguous and do not show signs of clear negative sentiment, TuringBERT manages to classify them into the "NEGATIVE" class with better accuracy. Additionally, BERT again misses clear signs of negative sentiment like vocabulary choice "dark" or "sinister", as illustrated by the examples below.

- "'You have been cruelly used,' said Holmes." - misclassified by BERT as "POSITIVE"
- "Dark enough and sinister enough." - misclassified by BERT as "POSITIVE"
- "We are only just in time to prevent some subtle and horrible crime." - misclassified by BERT as "POSITIVE"
- "We had no feeling of security unless our doors were locked." - misclassified by TuringBERT as "POSITIVE"

Overall, the "NEGATIVE" class seems to be the least problematic (68 mistakes made by both models). The other two classes are comparable difficult, with 143 mistakes made in the "POSITIVE" class and 134 mistakes in the "NEUTRAL" category. However, models seem to struggle differently. TuringBERT makes more mistakes in the neutral category. In examples like "'Ah, but I sleep more heavily than you.'", the model assigns the "NEGATIVE", possibly relying on the word *heavily*, not realizing that it does not carry the same meaning in the context of sleeping, as it would when for example describing food or people. Similarly, in the example of "'What's in here?' he asked, tapping the safe.", TuringBERT makes the same mistake. A possible explanation is that it relies in the word *tapping*, which could indicate stress or hurry, however it does not in this slightly ambiguous example.

In the case of the "NEGATIVE" category, BERT seems to be struggling more than TuringBERT. It may be due to perhaps slightly outdated expressions like *to have a say*, like in the example of "I will go when I have said my say," which BERT misclassified as neutral. Another case of a misclassification is "'It is a nice household,' he murmured.", which BERT labels as positive probably due to the word *nice*. It does not, however, pick up on the word *murmured*, which in this context suggests sarcasm or unclear intentions. Finally, BERT seems to overlook sentiment in questions. In the example of "But have you told me all?" BERT assigns the neutral label. Nevertheless, the question is part of an interrogation and implies that the speaker is aware of som undisclosed information.

5.2.2 Qualitative Analysis: Modern Benchmark

Similarly, to the historical-literary benchmark, there are a lot of shared mistakes between the models. Table 5.2 shows the number of unique mistakes each model made on the Friends corpus, as well as the number of shared mistakes.

The models seem to struggle the least with the positive category, collectively they make 257 errors. BERT seems to often misclassify exclamations like "Ohh!" or "Ahhh!" as negative instead of positive. Interestingly, it also does not interpret endearment terms like *baby* in "That's right baby." or modifiers like *million* in "Thanks a million.", both of which indicate a more positive sentiment. TuringBERT, on the other hand, struggles with context dependent meaning. In the example "It kicked! I think the baby kicked!" the verb *kicked* does not cause any physical harm to the mother, and is therefore a

	"POSITIVE"	**"NEGATIVE"**	**"NEUTRAL"**
shared	184	253	287
BERT	35	58	77
TuringBERT	38	98	64

Table 5.2: Number of unique and shared mistakes made by the models on the Friends corpus (misclassifications per class).

sign that evokes positive emotions in people. Similarly, in "Good! A verbal contract is binding in the state of New York!" the word *binding*, that could mean physically restricting, here is a law related term. The possible explanation is that the model relies on the wrong meanings for those words and in result assigns the wrong sentiment.

The models also perform similarly in the number of mistakes made in the neutral category. Most of the examples on which the models make mistakes are ambiguous and could indicate sentiment other than neutral. For example BERT misclassifies sentences like "Oh no wait, oh no, the elastic on my underwear busted" with the leading indicator *busted* or "Actually it's, it's quite, y'know, typical behavior when you have this kind of dysfunctional group dynamic" with the word *dysfunctional*. Both sentences, like many other examples, could be argued to present non neutral sentiment. TuringBERT makes similar mistakes. In example "Yeah. I really like his glasses," it assigns positive sentiment indicated by the phrase *really like*. However, TuringBERT again misses pragmatic meaning changes. For instance, in "Hey, dragon! Here's your tips from Monday and Tuesday," it assigns "NEGATIVE" label, likely due to the noun *dragon*. Normally, *dragon* could be an indicator of danger and thus negative sentiment, however in this example it a figurative expression that can be understood within a broader comedic context.

Finally, the mistakes made in the negative category are due mainly to ambiguous sentences like "Why?" or "Oh God!", that on their own could be easily understood as neutral. However, BERT does not pick up on the exclamation mark in "Uh, it's 2:30 in the morning!" and the context that likely suggests that the speaker is not simply stating what time it is, but also expresses their irritation. TuringBERT repeatedly makes context dependent and pragmatic mistakes. In the example "Like me tiny doctor!" the adjective *tiny* is not a modifier that serves endearment but likely is used to humiliate the other person. In the example of "Janice, what umm, what are you doing here?" the hesitation marker *umm* could indicate the negative feelings the speaker has towards Janice, however the model assigns the "NEUTRAL" label instead of "NEGATIVE".

Chapter 6

Discussion

This chapter delves into the implications of the results presented in the previous chapters (Chapters 4 and 5).

6.1 Discussion of Results

6.1.1 Discussion: Token-Level Experiments

This section focuses specifically on the findings in Section 4.2. While The TuringBERT model was able to perform the sentiment analysis task with satisfactory results on the token level, the BERT model failed. At the first attempt, BERT did not assign any ”POSITIVE” labels. An investigation of the data showed that underfitting of this one label was not the problem. The other approaches presented in Section 4.2 also proved unsuccessful at addressing the missing ”POSITIVE” label.

As the final step, the training data was examined in the hope of identifying a possible cause. The analysis of the Sherlock Holmes corpus showed that the sentiment annotations were often ambiguous and inconsistent. As explained in Section 3.2.1 the dataset was originally annotated on the concept level. This means that only tokens that constitute word meanings can be annotated or sentiment. It could be the case that mistakes in labeling the concepts were made by the annotators. A search for concepts with different sentiment values showed ambiguous cases of annotation. Table 6.1 shows some of the sentiment values and tokens to which they were assigned. While most tokens are assigned expected values, like ”brilliant” with a score of 95 or ”diabolical” with a score of -95, there are some words that seem to be assigned a wrong value, like ”thunder” with a sentiment -47 which potentially might have been a neutral word. Moreover, several words, like ”sir”, ”keen”, or ”abandon”, are assigned multiple values.

A closer look into the data showed further inconsistency in which concepts are annotated and which are not. The following two examples are excerpts from ”The Hound of the Baskervilles”:

- ”Since we have been so unfortunate as to miss him and have no notion of his errand, this accidental souvenir becomes of importance.”
- ”’Has anything escaped me?’ I asked, with some self-importance.”

Both sentences use ”miss” and ”escape” in the sense of omitting something. However, only in the second example the word is marked for sentiment with a score od

Sentiment	Concepts
95	best, brilliant, excellent
70	sir
64	admirable, beautiful, affection
60	rich, keen
38	keen
34	achievement, admirable, accomplish
12	sir
-20	abandon, barren
-34	abandon, absent
-47	thunder
-64	agony, agitated, abhor
-95	diabolical, foul, evil

Table 6.1: Concepts and their sentiment values (from "The Hound of the Baskervilles").

-34. The listed ambiguities and inconsistencies could be a cause of the model's failure to assign positive labels. Nevertheless, the total of "POSITIVE" labels in the training test dataset is comparable to that of the "NEGATIVE" label (respectively 1053 and 1508). This would suggest that the model should struggle with both categories, yet it only fails to assign the positive one.

Given that further work in the token-level direction was unlikely to produce meaningful results within a reasonable time frame. To avoid spending additional effort on a token-level setup that consistently failed to detect positive sentiment, the decision was made to shift focus to sentence-level experiments.

6.1.2 Discussion: Sentence-Level Experiments

This section focuses specifically on the findings in Section 4.3. Table 6.2 shows the mean macro averaged F1-scores of the models evaluated on both benchmarks. Contrary to the initial hypothesis, BERT outperforms TuringBERT on both benchmarks. This result is particularly unexpected in the case of the historical-literary benchmark, where TuringBERT was anticipated to hold an advantage due to its training on a combination of modern and historical data.

	BERT	TuringBERT
Friends corpus	0.680	0.666
Sherlock Holmes corpus	0.480	0.418

Table 6.2: F1-scores comparison of BERT and TuringBERT

Both models achieve significantly higher scores on the Friends corpus compare to the Sherlock Holmes corpus. The decline in performance of the models is proportional: a difference of 0.200 in case of BERT and 0.248 in case of TuringBERT. This suggests that the historical-literary benchmark presents greater challenges, regardless of which model was used. This interpretation is reinforced by kappa coefficient results: while for the Friends corpus the kappa scores exceeded 0.8, indicating strong agreement, the scores for the Sherlock Holmes corpus drop to around 0.3, reflecting limited agreement and highlighting the dataset's complexity. A drop in performance of this magnitude, combined with low kappa scores, suggests that the linguistic intricacy is not the sole reason the models struggle. A likely contributing factor is the substantial disparity in training set size, with the Friends corpus containing almost 10,000 examples compared to only 599 in the Sherlock Holmes corpus. This significant imbalance likely restricted the models' capacity to learn robust patterns, resulting in consistently lower scores. The pronounced imbalance in training dataset sizes complicates the assessment of the second part of the hypothesis - that TuringBERT would perform comparably on both modern and historical data.

Finally, the classification reports showed that TuringBERT consistently achieves lower performance on the "NEUTRAL" class when evaluated on the Sherlock Holmes corpus. As noted in Section 4.3.2, the neutral class is comparable in size to the positive class (145 and 177 examples, respectively). However, TuringBERT struggles to assign the "NEUTRAL" label, achieving an F1-score of 0.221 compared to 0.423 for "POSITIVE". This difference in performance might be due to the small size of the training set. A dataset that is not large enough could prevent the model from generalizing well, especially for the least common label. Alternatively, the lower performance on the "NEUTRAL" class might stem from the linguistic complexity of the historical-literary corpus. This will be further explored in Section 5 (Error Analysis).

6.1.3 Discussion: Error Analysis

Error analysis presented in Chapter 5 provided some valuable insights into the results. Firstly, it showed that both models share a significant number of errors made across the benchmarks. This suggests that lower scores on the historical-literary benchmark stem from linguistic complexity of the corpus. As both models made a lot of the same mistakes it is difficult to draw clear conclusions on their ability to process historical texts.

An often finding was that in both datasets the example sentences were to short, ambiguous in sentiment or lacked context for the models to assign the right label. It is important to note that both corpora are coherent pieces of narration and were annotated as such. The Sherlock Holmes dataset is a short story and the Friends corpus is a collection of dialogue lines from the show. This means that the sentiment label often does not correspond to the isolated sentence, but only makes sense when it a part of a broader narration. Separating the examples from context likely hindered meaningful processing and resulted in the mentioned errors.

It should be noted that within the unique mistakes the models made, they still exhibited similar patterns. While the exact sentences misclassified differed, the models frequently made errors on similar types of examples. Aside from the shared tendencies of overlooking interpunction or clearly positive and negative vocabulary, the models presented subtle patterns of their own. On one hand, TuringBERT seems to struggle with context dependent meaning. The model seems to depend on the most common meaning of words that in specific contexts might carry another meaning and therefore a different sentiment. This further relates to pragmatics and notions of figurativeness or sarcasm, which sometimes pose a challenge for TuringBERT. BERT, on the other hand, has happened to misinterpret outdated phrases and overlook modifiers that attribute positive sentiment. Overall, however, the number of shared mistakes, combined with the small number of unique mistakes, does not suffice to draw clear conclusions, especially in the case of the larger Friends dataset.

6.2 Limitations

This research was subject to several constraints that influenced its scope and outcomes. One primary limitation was time, which restricted the depth of the conducted experiments. In particular, it limited the exploration of failures observed in the token-level experiments, which could have provided additional insights into the BERT's misclassification patterns.

A significant limitation arose from the imbalanced training datasets. The Sherlock Holmes corpus contains only 599 examples, compared to nearly 10,000 in the Friends corpus. This substantial difference likely reduced the models' limited ability to generalize, resulting in significantly lower overall scores on the historical-literary corpus. The imbalanced sizes of the training sets make it difficult to draw firm conclusions about the models' performance on the historical data. While the performance decline on the Sherlock Holmes dataset may reflect its linguistic complexity, it is difficult to separate these effects from those of the limited dataset size.

Finally, the evaluation of the models was done based on the sentiment analysis task, which while informative, may not fully capture all aspects of understanding historical texts. SA focuses on expressed emotions and opinions, but it does not reflect every language aspect needed for comprehensive historical analysis. Moreover, the findings are specific to the selected corpora, and model performance on other historical or literary texts may differ, particularly if these texts vary in genre, style, time period, or annotation quality. The corpora used in this study are not directly comparable: the Sherlock Holmes dataset consists of a collection of stories, whereas the Friends corpus is composed of dialogue lines. This further complicates direct comparison of models' performance, as the corpora do not represent the exact same genre.

6.3 Future Work

Future research directions could extend the scope of this study in several directions, given the previously outlined limitations. The primary step could mean expanding the training datasets, particularly for the historical-literary corpus to improve generalization. Making sure that both benchmarks use the same size of the training dataset could allow a more reliable evaluation for the models. Other stories in the Sherlock Holmes collection could be annotated on the sentence level to expand the corpus.

Secondly, while this thesis focuses on sentiment analysis, future work could implement a variety of NLP tasks that would capture different linguistic aspects. Complementary tasks could include semantic role labeling, entity recognition, or syntactic parsing. A wider range of NLP tasks would ensure a more comprehensive understanding of the model.

Finally, broader generalization should be examined by evaluating models on a wider range of historical and literary corpora that vary in genre, style, and time period. Since the current study compares a collection of short stories with a dialogue-based corpus, future research should investigate datasets that are more directly comparable. Comparing models on texts of the same domain and genre would enable a clearer assessment of model capabilities.

Chapter 7

Conclusion

This study aimed to evaluate TuringBERT, a historical language model, at the sentiment analysis task against it modern counterpart BERT. The models were tested on a historical-literary dataset and modern benchmark. The approach included a series of experiments and quantitative and qualitative analysis of the results.

The initial token-level experiments did not produce meaningful results, as the BERT model failed to assign any "POSITIVE" labels. Several attempts at adapting the data and exploring the problem were made, however did not prove successful. Given that further work in that direction was unlikely to to produce satisfying results, the decision was made to approach the task at sentence level.

The sentence-level experiments provided inconclusive and surprising results. While TuringBERT was hypothesized to outperform BERT on the historical-literary benchmark, it was not the case. The average F1-scores were comparable between the models, however BERT scored higher. The Sherlock Holmes corpus proved to be a similar challenge to both models, as the decline in their performance was essentially equivalent. A closer analysis showed that, in comparison to the Friends dataset, the historical-literary benchmark was not significantly more demanding in linguistic complexity. This suggested that the drop in performance may be related to something else. The Sherlock Holmes dataset is significantly smaller than the Friends corpus. Due to the difference in training set size, it is difficult to draw clear conclusions about the models' performance.

Nevertheless, the slightly lower scores of TuringBERT on both benchmarks suggest that the training of the model did have an impact on the classification patterns it made. As error analysis showed, there are slight differences in interpretation of vocabulary, interpunction and pragmatic meaning between the models. Thus, while this study did not manage to firmly answer its research question, it does present promising findings that should be explored. Further research of the historical models should include similar training set sizes and more comparable in genre benchmarks.

Appendix A

Token-level experiment results; BERT and TuringBERT models

	Precision	Recall	F1-Score	Support
NEGATIVE	0.3996	0.8634	0.5464	1508
NEUTRAL	0.9830	0.9729	0.9779	67799
POSITIVE	0.0000	0.0000	0.0000	1053
Accuracy			0.9560	70360
Macro Avg	0.4609	0.6121	0.5081	70360
Weighted Avg	0.9558	0.9560	0.9540	70360

Table A.1: Classification report for BERT seed 4 on token-level experiment

	Precision	Recall	F1-score	Support
NEGATIVE	0.4194	0.8694	0.5658	1508
NEUTRAL	0.9829	0.9747	0.9788	67799
POSITIVE	0.0000	0.0000	0.0000	1053
Accuracy			0.9578	70360
Macro Avg	0.4674	0.6147	0.5149	70360
Weighted Avg	0.9561	0.9578	0.9553	70360

Table A.2: Classification report for BERT seed 23 on token-level experiment

	Precision	Recall	F1-score	Support
NEGATIVE	0.4172	0.8534	0.5604	1508
NEUTRAL	0.9826	0.9750	0.9788	67799
POSITIVE	0.0000	0.0000	0.0000	1053
Accuracy			0.9578	70360
Macro Avg	0.4666	0.6095	0.5131	70360
Weighted Avg	0.9558	0.9578	0.9552	70360

Table A.3: Classification report for BERT seed 95 on token-level experiment

	Precision	Recall	F1-Score	Support
NEGATIVE	0.4246	0.8389	0.5639	1508
NEUTRAL	0.9932	0.9559	0.9742	67799
POSITIVE	0.3636	0.7341	0.4863	1053
Accuracy			0.9501	70360
Macro avg	0.5938	0.8430	0.6748	70360
Weighted avg	0.9716	0.9501	0.9581	70360

Table A.4: Classification report for TuringBERT seed 4 on token-level experiment

	Precision	Recall	F1-Score	Support
NEGATIVE	0.4123	0.8402	0.5532	1508
NEUTRAL	0.9937	0.9530	0.9729	67799
POSITIVE	0.3513	0.7559	0.4797	1053
Accuracy			0.9476	70360
Macro Avg	0.5858	0.8497	0.6686	70360
Weighted Avg	0.9716	0.9476	0.9565	70360

Table A.5: Classification report for TuringBERT seed 23 on token-level experiment

	Precision	Recall	F1-Score	Support
NEGATIVE	0.3954	0.8448	0.5387	1508
NEUTRAL	0.9930	0.9540	0.9731	67799
POSITIVE	0.3706	0.7056	0.4859	1053
Accuracy			0.9479	70360
Macro avg	0.5863	0.8348	0.6659	70360
Weighted avg	0.9709	0.9479	0.9565	70360

Table A.6: Classification report for TuringBERT seed 95 on token-level experiment

Appendix B

Sentence-level experiment results on the Sherlock Holmes corpus

	Precision	Recall	F1-Score	Support
NEGATIVE	0.6606	0.7798	0.7152	277
NEUTRAL	0.4921	0.4276	0.4576	145
POSITIVE	0.4795	0.3955	0.4334	177
Accuracy			0.5810	599
Macro Avg	0.5440	0.5343	0.5354	599
Weighted Avg	0.5663	0.5810	0.5696	599

Table B.1: Classification report for BERT seed 4 on sentence-level experiment on the Sherlock Holmes corpus

	Precision	Recall	F1-Score	Support
NEGATIVE	0.5542	0.8484	0.6705	277
NEUTRAL	0.6140	0.2414	0.3465	145
POSITIVE	0.5763	0.3842	0.4610	177
Accuracy			0.5643	599
Macro Avg	0.5815	0.4913	0.4927	599
Weighted Avg	0.5752	0.5643	0.5302	599

Table B.2: Classification report for BERT seed 23 on sentence-level experiment on the Sherlock Holmes corpus.

	Precision	Recall	F1-Score	Support
NEGATIVE	0.5431	0.9097	0.6802	277
NEUTRAL	0.5500	0.0759	0.1333	145
POSITIVE	0.5391	0.3503	0.4247	177
Accuracy			0.5426	599
Macro Avg	0.5441	0.4453	0.4127	599
Weighted Avg	0.5436	0.5426	0.4723	599

Table B.3: Classification report for BERT seed 95 on sentence-level experiment on the Sherlock Holmes corpus.

	Precision	Recall	F1-Score	Support
NEGATIVE	0.5202	0.9278	0.6667	277
NEUTRAL	0.5862	0.1172	0.1954	145
POSITIVE	0.4868	0.2090	0.2925	177
Accuracy			0.5192	599
Macro Avg	0.5311	0.4180	0.3849	599
Weighted Avg	0.5263	0.5192	0.4420	599

Table B.4: Classification report for TuringBERT seed 4 on sentence-level experiment on the Sherlock Holmes corpus.

	Precision	Recall	F1-Score	Support
NEGATIVE	0.5376	0.9025	0.6739	277
NEUTRAL	0.4872	0.1310	0.2065	145
POSITIVE	0.5684	0.3051	0.3971	177
Accuracy			0.5392	599
Macro avg	0.5311	0.4462	0.4258	599
Weighted avg	0.5345	0.5392	0.4789	599

Table B.5: Classification report for TuringBERT seed 23 on sentence-level experiment on the Sherlock Holmes corpus.

	Precision	Recall	F1-score	Support
NEGATIVE	0.5396	0.9350	0.6843	277
NEUTRAL	0.5556	0.1379	0.2210	145
POSITIVE	0.6627	0.3107	0.4231	177
Accuracy			0.5576	599
Macro avg	0.5859	0.4612	0.4428	599
Weighted avg	0.5798	0.5576	0.4949	599

Table B.6: Classification report for TuringBERT seed 95 on sentence-level experiment on the Sherlock Holmes corpus.

Appendix C

Sentence-level experiment results on the Friends corpus

	Precision	Recall	F1-Score	Support
NEGATIVE	0.6610	0.6483	0.6545	833
NEUTRAL	0.7731	0.7866	0.7798	1256
POSITIVE	0.6291	0.6219	0.6255	521
Accuracy			0.7096	2610
Macro Avg	0.6877	0.6856	0.6866	2610
Weighted Avg	0.7086	0.7096	0.7090	2610

Table C.1: Classification report for BERT seed 4 on sentence-level experiment on the Friends corpus.

	Precision	Recall	F1-Score	Support
NEGATIVE	0.6484	0.6267	0.6374	833
NEUTRAL	0.7658	0.7811	0.7734	1256
POSITIVE	0.6164	0.6200	0.6182	521
Accuracy			0.6996	2610
Macro Avg	0.6769	0.6759	0.6763	2610
Weighted Avg	0.6985	0.6996	0.6990	2610

Table C.2: Classification report for BERT seed 23 on sentence-level experiment on the Friends corpus.

	Precision	Recall	F1-score	Support
NEGATIVE	0.6567	0.6315	0.6438	833
NEUTRAL	0.7617	0.7890	0.7751	1256
POSITIVE	0.6181	0.6027	0.6103	521
Accuracy			0.7015	2610
Macro avg	0.6788	0.6744	0.6764	2610
Weighted avg	0.6995	0.7015	0.7003	2610

Table C.3: Classification report for BERT seed 95 on sentence-level experiment on the Friends corpus.

	Precision	Recall	F1-Score	Support
NEGATIVE	0.6333	0.6158	0.6245	833
NEUTRAL	0.7687	0.7858	0.7772	1256
POSITIVE	0.6008	0.5950	0.5979	521
Accuracy			0.6935	2610
Macro avg	0.6676	0.6656	0.6665	2610
Weighted avg	0.6920	0.6935	0.6926	2610

Table C.4: Classification report for TuringBERT seed 4 on sentence-level experiment on the Friends corpus.

	Precision	Recall	F1-Score	Support
NEGATIVE	0.6469	0.6026	0.6240	833
NEUTRAL	0.7676	0.7970	0.7820	1256
POSITIVE	0.6019	0.6123	0.6070	521
Accuracy			0.6981	2610
Macro Avg	0.6721	0.6706	0.6710	2610
Weighted Avg	0.6960	0.6981	0.6967	2610

Table C.5: Classification report for TuringBERT seed 23 on sentence-level experiment on the Friends corpus.

	Precision	Recall	F1-Score	Support
NEGATIVE	0.6386	0.6002	0.6188	833
NEUTRAL	0.7544	0.8121	0.7822	1256
POSITIVE	0.6126	0.5585	0.5843	521
Accuracy			0.6939	2610
Macro Avg	0.6685	0.6570	0.6618	2610
Weighted Avg	0.6892	0.6939	0.6906	2610

Table C.6: Classification report for TuringBERT seed 95 on sentence-level experiment on the Friends corpus.

References

[1] Al-Laith, A., Conroy, A., Bjerring-Hansen, J., & Hershcovich, D. (2024). Development and evaluation of pre-trained language models for historical Danish and Norwegian literary texts. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)* (pp. 4811–4819). ELRA and ICCL. `https://aclanthology.org/2024.lrec-main.431/`

[2] Asghar, M. Z., Khan, A., Ahmad, S., & Kundi, F. M. (2014). A review of feature extraction in sentiment analysis. *Journal of Basic and Applied Scientific Research, 4*(3), 181–186.

[3] Bender, E. M., & Friedman, B. (2018). Data statements for natural language processing: Toward mitigating system bias and enabling better science. *Transactions of the Association for Computational Linguistics, 6*, 587–604. `https://doi.org/10.1162/tacl_a_00041`

[4] Bond, F., Ohkuma, T., Costa, L. M. D., Miura, Y., Chen, R., Kuribayashi, T., & Wang, W. (2016). A multilingual sentiment corpus for Chinese, English and Japanese.

[5] Chen, S. Y., Hsu, C. C., Kuo, C. C., & Ku, L. W. (2018). EmotionLines: An emotion corpus of multi-party conversations. *arXiv.* `https://arxiv.org/abs/1802.08379`

[6] Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv.* `https://doi.org/10.48550/arXiv.1810.04805`

[7] Ekman, P., Friesen, W. V., O'Sullivan, M., Chan, A., Diacoyanni-Tarlatzis, I., Heider, K., Krause, R., LeCompte, W. A., Pitcairn, T., Ricci-Bitti, P. E., & others. (1987). Universals and cultural differences in the judgments of facial expressions of emotion. *Journal of Personality and Social Psychology, 53*(4), 712–717.

[8] Faudree, P., & Pharao Hansen, M. (2014). Language, society, and history: Towards a unified approach? In N. J. Enfield, P. Kockelman, & J. Sidnell (Eds.), *The Cambridge handbook of linguistic anthropology* (pp. 223–245). Cambridge University Press. `https://doi.org/10.1017/CBO9781139342872.011`

[9] Feltgen, Q., Fagard, B., & Nadal, J.-P. (2017). Frequency patterns of semantic change: Corpus-based evidence of a near-critical dynamics in language change. *Royal Society Open Science, 4*(11), 170830. `https://doi.org/10.1098/rsos.`

170830

[10] Fox, N. P., & Ehmoda, O. (2012). Statistical stylometrics and the Marlowe–Shakespeare authorship debate.

[11] Fung, G. (2003). The disputed Federalist Papers: SVM feature selection via concave minimization. In *Proceedings of the Richard Tapia Celebration of Diversity in Computing Conference 2003* (pp. 42–46). ACM.

[12] Grandini, M., Bagli, E., & Visani, G. (2020). Metrics for multi-class classification: An overview. *arXiv*. `https://arxiv.org/abs/2008.05756`

[13] Handzic, M., & Mulavdić, V. (2022). Combining close and distant reading: A plausible way forward? Paper presented at the Distant Reading COST Action Closing Conference.

[14] Hills, T., & Miani, A. (2025). A short primer on historical natural language processing. In G. Progrebna & T. Hills (Eds.), *Handbook of behavioural data science*. Cambridge University Press.

[15] Hosseini, K., Beelen, K., Colavizza, G., & Ardanuy, M. C. (2021). Neural language models for nineteenth-century English. *Journal of Open Humanities Data, 7*(0), 22. `https://doi.org/10.5334/johd.48`

[16] Kim, E., & Klinger, R. (2022). A survey on sentiment and emotion analysis for computational literary studies. *arXiv*. `https://doi.org/10.17175/2019_008`

[17] Maas, A. L., Daly, R. E., Pham, P. T., Huang, D., Ng, A. Y., & Potts, C. (2011). Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies* (pp. 142–150). Association for Computational Linguistics. `https://aclanthology.org/P11-1015/`

[18] Manjavacas, E., & Fonteyn, L. (2021). MacBERTh: Development and evaluation of a historically pre-trained language model for English (1450–1950). In *Proceedings of the Workshop on Natural Language Processing for Digital Humanities* (pp. 23–36). NIT Silchar, India: NLP Association of India (NLPAI).

[19] Manjavacas, E., & Fonteyn, L. (2022). Adapting vs. pre-training language models for historical languages. *Journal of Data Mining & Digital Humanities, NLP4DH*. `https://doi.org/10.46298/jdmdh.9152`

[20] Mao, Y., Liu, Q., & Zhang, Y. (2024). Sentiment analysis methods, applications, and challenges: A systematic literature review. *Journal of King Saud University - Computer and Information Sciences, 36*(4), 102048. `https://doi.org/10.1016/j.jksuci.2024.102048`

[21] Nayak, A., Timmapathini, H., Ponnalagu, K., & Gopalan Venkoparao, V. (2020). Domain adaptation challenges of BERT in tokenization and sub-word representations of out-of-vocabulary words. In A. Rogers, J. Sedoc, & A. Rumshisky (Eds.), *Proceedings of the First Workshop on Insights from Negative Results in*

NLP (pp. 1–5). Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.insights-1.1

[22] Nevalainen, T. (2006). Towards a standard language. In *An Introduction to Early Modern English* (pp. 29–42). Edinburgh University Press. http://www.jstor.org/stable/10.3366/j.ctt1g09z3p

[23] Living with Machines Team. (n.d.). Introduction to genre classification. In *Living with Machines: Genre classification.* Retrieved June 13, 2025, from https://living-with-machines.github.io/genre-classification/intro.html

[24] Periti, F., Picascia, S., Montanelli, S., Ferrara, A., & Tahmasebi, N. (2025). Studying word meaning evolution through incremental semantic shift detection. *Language Resources & Evaluation, 59*, 1363–1399. https://doi.org/10.1007/s10579-024-09769-1

[25] Piotrowski, M. (2012). *Natural language processing for historical texts.* Morgan & Claypool Publishers.

[26] Poria, S., Hazarika, D., Majumder, N., Naik, G., Mihalcea, R., & Cambria, E. (2018). MELD: A multimodal multi-party dataset for emotion recognition in conversation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL).*

[27] Rosenthal, S., Farra, N., & Nakov, P. (2017). SemEval-2017 Task 4: Sentiment analysis in Twitter. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)* (pp. 502–518). Association for Computational Linguistics. https://aclanthology.org/S17-2093/

[28] Schmidt, T., Dennerlein, K., & Wolff, C. (2021). Emotion classification in German plays with transformer-based language models pretrained on historical and contemporary language. In *Proceedings of the 5th Joint SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature* (pp. 67–79). Association for Computational Linguistics. https://aclanthology.org/2021.latechclfl-1.8/

[29] Silva, K., Can, B., Blain, F., Sarwar, R., Ugolini, L., & Mitkov, R. (2023). Authorship attribution of late 19th century novels using GAN-BERT. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 4: Student Research Workshop)* (pp. 310–320). Association for Computational Linguistics. https://aclanthology.org/2023.acl-srw.44/

[30] Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C. D., Ng, A., & Potts, C. (2013). Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing* (pp. 1631–1642). Association for Computational Linguistics. https://aclanthology.org/D13-1170/

[31] Sun, K., & Wang, R. (2022). The Evolutionary Pattern of Language in English Fiction Over the Last Two Centuries: Insights From Linguistic Concreteness and

Imageability. https://doi.org/10.31234/osf.io/9vywb_v1.

[32] Szabó, M. K., Ring, O., Nagy, B., Kiss, L., Koltai, J., Berend, G., Vidács, L., Gulyás, A., & Kmetty, Z. (2020). Exploring the dynamic changes of key concepts of the Hungarian socialist era with natural language processing methods. *Historical Methods: A Journal of Quantitative and Interdisciplinary History, 54*(1), 1–13. https://doi.org/10.1080/01615440.2020.1823289

[33] Tan, L., & Bond, F. (2012). Building and annotating the linguistically diverse NTU-MC (NTU-multilingual corpus). *International Journal of Asian Language Processing, 22*(4), 161–174.

[34] Wu, Y., Schuster, M., Chen, Z., Le, Q. V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., & others. (2016). Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*. https://arxiv.org/abs/1609.08144

[35] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. In *Advances in Neural Information Processing Systems* (Vol. 30, pp. 6000–6010). Curran Associates, Inc.

[36] Zhou, P., Shi, W., Tian, J., Qi, Z., Li, B., Hao, H., & Xu, B. (2016). Attention-based bidirectional long short-term memory networks for relation classification. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL)* (pp. 207–212). Association for Computational Linguistics. https://aclanthology.org/P16-2034/