SAARLAND UNIVERSITY

Faculty of Mathematics and Computer Science
Department of Computer Science
Master Thesis

Leveraging Contextual Information for Spoiler Detection in Movie Reviews

submitted by
Enam Biswas
Saarbrücken
December 2024

Advisor:
Xenia Klinge
German Research Center for Artificial Intelligence
Saarland Informatics Campus
Saarbrücken, Germany

Supervisor:
Prof. Dr. Antonio Krüger
German Research Center for Artificial Intelligence
Saarland Informatics Campus
Saarbrücken, Germany

Reviewers:
Prof. Dr. Antonio Krüger
German Research Center for Artificial Intelligence
Saarland Informatics Campus
Saarbrücken, Germany

Dr. Fabrizio Nunnari
German Research Center for Artificial Intelligence
Saarland Informatics Campus
Saarbrücken, Germany

Saarland University
Faculty MI - Mathematics and Computer Science
Department of Data Science & Artificial Intelligence
Campus - Building E1.1
66123 Saarbrücken
Germany

Declarations

Statement of Authorship:
I affirm that this thesis has been composed by me independently. All sources and aids used have been duly acknowledged and listed in this thesis.

Saarbrücken, Date:

Consent to Publish:
I consent to the publication of my thesis, assuming it achieves a passing grade, by inclusion in the academic library of the Department of Data Science and Artificial Intelligence.

Saarbrücken, Date:

Acknowledgements

I would like to extend my deepest gratitude to several key individuals and groups who have contributed to the successful completion of this thesis. First and foremost, I owe a debt of gratitude to my supervisor, **Prof. Dr. Antonio Krüger**, for his invaluable feedback during the master's thesis seminars. His guidance was crucial in shaping the direction and execution of this research.

I am also profoundly appreciative of the insightful critiques and suggestions provided by **Dr. Fabrizio Nunnari**. This work has been significantly enhanced by his review and feedback, which have contributed to the thesis's overall quality by offering unique perspectives.

Special thanks are due to my advisor, **Xenia Klinge**, whose consistent and insightful feedback throughout the process was indispensable. Her dedication and support have been a cornerstone of this journey, and for that, I am profoundly thankful.

I am also grateful to the **ERS team at CISPA** for generously lending their workstations, which were essential in conducting the research necessary for this thesis.

Lastly, I cannot express enough thanks to my friends and family, who have been my constant support system. Their unwavering encouragement and belief in my abilities have been a source of strength and motivation throughout this challenging process.

Abstract

One of the particular issues that arises from the proliferation of user-generated content on online movie platforms is the unintentional exposure to spoilers. This may significantly diminish the viewing experience of media content. To dramatically expand the scope and usefulness of predictive analytics in this field, this thesis presents an enhanced model for spoiler identification in movie reviews by utilizing a comprehensive dataset of over 3 million English movie reviews until late 2020. This study investigates the usefulness of contextual information in spoiler identification using a range of deep learning methodologies, such as transformer-based models, Language Model-based methods (LLMs), and conventional statistical learning models.

Through validation on a large test set of 300,000 samples, these techniques exceed established benchmarks like the **MSVD** architecture by significant margins, reaching 85% accuracy and industry-leading average F1-score of 0.84. Furthermore, this study's analysis has been expanded to incorporate evaluations from 20 of the most well-liked films released in 2022 and 2023—a dataset the model has never seen before. Despite this, the model performed well, as shown by a weighted average precision of 0.83 and an outstanding AUC of 0.8256 over a test set with 55,000 samples. Additionally, the proposed architecture performs better than the **Llama 3.1 8B Instruct** model, whose average F1 score is only 0.71 for this task. This illustrates how, even when evaluated on an entirely fresh dataset, my model attains a closer alignment with the human viewpoint on spoilers.

Overall, the thesis introduces a state-of-the-art architecture for the task, technologically pushing the frontiers of deep learning applications in natural language processing. It also adds a novel way of dataset preprocessing by producing a clean dataset and model's decision visualization technique that will certainly be useful for further study in the field. Additionally, this study establishes a new performance standard. It emphasizes the critical role contextual information plays in spoiler detection, demonstrating the effectiveness of deep learning models in adapting to novel and different data settings.

Contents

Chapter 1
Introduction

Figure 1.1: Common Words in Spoiler Reviews

The widespread availability of digital media has fundamentally altered the manner in which content is consumed. Additionally, this presents difficulties in terms of content management and user experience, particularly with regard to the influence that spoilers in movie reviews have on the enjoyment and engagement of viewers. Using contextual information such as synopses, plots, or scripts, as well as feature engineering of relevant textual properties, this thesis addresses the difficulty of detecting spoilers in IMDb movie reviews. Specifically, the research focuses on exploiting contextual information. Enhancing the user experience by proactively recognizing spoilers without the need for explicit user tagging is the key goal behind this. This has the potential to preserve the excitement and satisfaction that are obtained from the consumption of pristine media.

As part of this project, advanced machine learning techniques and natural language

processing are investigated in order to design a spoiler detection system that is both reliable and unique. A model that is able to comprehend complex textual cues and contextual significance is what is anticipated to be successful. It is the goal of the proposed system to overcome the conventional keyword-based or surface-level text analysis methods that are typically utilized in spoiler identification systems. It will use the reviews while also including more context from the films themselves. The overarching objective is to develop a prediction model that not only reliably identifies spoilers but also makes a contribution to the more general field of sentiment analysis and content filtering in the media.

Through this introduction, the basis will be laid for a more in-depth exploration of the topic. I will first explore the reasons for tackling this issue, then I will outline the precise research questions that will lead the inquiry. Finally, I will identify the objectives that are envisioned through this research investigation.

1.1 Motivation

The growing popularity of online movie platforms has led to an increase in the significance of user-generated material, such as movie reviews. However, this increase in online debate about movies has also increased the likelihood of encountering spoilers. Spoilers can prematurely expose critical narrative information, which can reduce the viewer's enjoyment of the content and their level of engagement with it. Traditional spoiler alerts are mainly dependent on voluntary user labeling, which is not only unreliable but also frequently overlooks key textual indications. These can ruin a narrative. On top of that, spoilers can vary from person to person.

The current methods for detecting spoilers in movie reviews are limited and not effective due to their reliance on explicit spoiler tags or basic text analysis techniques. There is a need for a more advanced system that can understand the complexities of language and context associated with spoilers. This system should incorporate contextual information from the synopses, storylines, and screenplays in order to effectively identify and prevent spoilers in movie reviews.

My research focuses on using machine learning and natural language processing to automatically detect spoilers. My methodologies 4 not only aims to preserve the integrity of movie storylines but also has broader applications in content management and digital media consumption. By ensuring that spoilers are properly managed, the system aims to maintain the emotional and psychological impact of movies. Which should lead to enhanced audience satisfaction and engagement.

In addition to improving the detection of spoilers, this thesis aims to contribute to the field of computational linguistics by advancing the understanding of how spoilers can be algorithmically identified in complex text structures. The insights gained from this research may have implications for sentiment analysis, context-aware content filtering, and the adaptive use of machine learning in media platforms. Eventually, it will benefit both the entertainment industry and the academic community.

1.2 Research Questions

The study aims to answer important questions about detecting spoilers using contextual information. These questions are as follows:

1. **How effectively can machine learning models utilize contextual information such as movie synopses and scripts to detect review spoilers?**
    - I aim to find how existing technologies can benefit by integrating and interpreting diverse sources of movie-related text data for accurate spoiler identification.
2. **What are the key textual features and indicators that most significantly predict spoilers within movie reviews?**
    - I want to identify specific linguistic features and narrative indicators that correlate strongly with spoilers.
3. **Can a model trained on a dataset with embedded contextual information generalize to other movies not included in the training set?**
    - I want to explore the adaptability and scalability of the model by assessing its performance across various movie genres and new releases.
4. **What improvements in spoiler detection performance are achieved when combining textual reviews with their corresponding movie contexts, compared to using reviews alone?**
    - I want to quantify the enhancements brought about by incorporating contextual data alongside reviews.
5. **Is it the contextual information or the patterns in text? What is allowing the deep learning models to make decisions?**
    - I want to explore if the added contextual information or some hidden pattern in data drives the decision made by the spoiler detection architecture.

1.3 Research Goals and Outlines

This research aims to develop a sophisticated system to detect spoilers with clearly defined goals aligned with this objective. The objectives are as follows:

1. **Develop a novel data cleaning method:** To produce a character-level data cleaning method employing character-level inspection and substitution.
2. **Develop an advanced spoiler detection model:** To construct and train a machine learning model that effectively utilizes contextual information from movie-related information and textual properties.
3. **Identify critical textual features:** To analyze and determine the textual elements and narratives indicative of spoilers.
4. **Evaluate model generalization:** To test the model's ability to generalize its spoiler detection capabilities to unseen movies.
5. **Contribute the largest cleaned spoiler detection dataset:** To develop and provide the largest dataset of cleaned movie-related texts for spoiler detection.
6. **Introduce a novel visualization technique:** To create a new visualization method that allows the analysis of the model's decision-making process.

7. **Contribute to academic and practical fields:** To provide insights and advancements in the NLP field and content filtering that other researchers and industry professionals can utilize.

1.4 Outline

The thesis is organized into the following chapters:

- **Chapter 1: Introduction** - Introduces the problem, motivation, research questions, and study goals.
- **Chapter 2: Related Work** - Discusses existing methods and technologies in spoiler detection outlining the gaps this research aims to fill.
- **Chapter 3: Dataset** - Provides a detailed dataset analysis and data cleaning method.
- **Chapter 4: Methodology** - Describes the model development process and experimental setup used to conduct the research.
- **Chapter 5: Results** - Presents the results and findings in the research questions and goals context.
- **Chapter 6: Discussion** - Analyzes the findings' importance, evaluates the study's limitations, and suggests areas for future research.
- **Chapter 7: Conclusion** - Summarizes the research, reiterates the contributions, and proposes practical applications of the spoiler detection system.

Chapter 2
Related Work

Research on spoiler detection in movie reviews is essential due to the substantial influence spoilers can exert on the viewer's experience. Any material that gives away important plot details or the conclusion of a story can be considered a spoiler and could make readers who haven't read the story yet feel less invested in the narrative. User-generated information is sometimes rife with spoilers. For example, movie reviews on sites like IMDb may purposefully or unintentionally reveal plot twists, dramatic moments, or the resolution of the story. They can come in a variety of shapes, such as debates that put specific actions in the context of the story or direct references to significant events that provide subliminal clues leading to the identification of plot points. Leavitt and Christenfeld's study [15, 16] states that spoilers can vary from significant plot twists to subliminal clues that could influence how the audience interprets the narrative. There is significant variation among individuals in their view of what qualifies as a spoiler. While some viewers only interpret significant narrative twists or the conclusion as spoilers, others view any indication at all [6]. Because of this variation in perception, spoiler identification is an especially difficult endeavor since it necessitates knowledge of both explicit and implicit indicators in the text [4].

The influence of spoilers on the level of enjoyment derived from a narrative has been a subject of contention and scholarly investigation. Although the common belief is that spoilers impair the experience, the empirical research offer an alternate viewpoint. Spoilers have the potential to diminish the sense of anticipation and emotional involvement in a narrative. Johnson and Rosenbaum discovered that plot spoilers reduce a story's suspense and immersion, which diminishes the enjoyment of the story [12]. According to their research, spoiled stories were perceived as less emotionally impactful, intellectually stimulating, and captivating compared to unspoiled stories [12]. This perspective is supported by psychological research, which indicates that anticipation plays a significant role in enjoyment. Revealing plot details in advance can reduce the sense of excitement and result in a less pleasurable encounter. For instance, a research published by the University of Amsterdam showed that the removal of tension and surprise from stories caused by spoilers lowers their entertainment value [12].

Studies indicate that spoilers may actually increase the enjoyment of a novel, despite popular belief to the contrary. Participants in Leavitt and Christenfeld's study frequently

preferred ruined stories over unspoiled ones. They contend that being aware of the ending ahead of time enables readers to comprehend the narrative more easily, recognize its subtleties, and concentrate on the process rather than the final product [15, 16]. This view is strengthened by the notion of perceptual fluency, wherein familiar or readily processed information tends to be more pleasurable. A deeper appreciation of the story's creative components can be achieved by decreasing cognitive burden and making the narrative more predictable, as suggested by [15]. For instance, understanding a mystery story's twist may help readers better understand the author's foreshadowing and hints, which could improve their reading experience overall [18, 25]. The inconclusive results indicate that the influence of spoilers is subjective and contingent on the context.

Some people may derive greater pleasure from a story when they are not consumed with speculating about the outcome, but others may like the suspense and unpredictability that accompanies an untarnished narrative [18, 19]. Additionally, research has demonstrated that individual variances, such as personality qualities, may impact how spoilers impact enjoyment. For instance, untouched stories may appeal more to those who value deep thought and have a high desire for cognition. A more focused emotional experience can be had by persons with a high demand for affect, who may choose spoilt stories due to their ability to lessen cognitive load [12].

2.1 Why Spoiler Detection?

There are various reasons why spoiler detection is important. Finding spoilers in reviews can seriously reduce a moviegoer's experience for a lot of people. According to [4], automated spoiler detection systems can assist readers in reading reviews without worrying about important narrative details being disclosed. Automated spoiler identification can improve user experience and engagement on online platforms by preventing spoilers in user-generated content [1]. Tolerance for spoilers varies among cultures and individuals. According to [18], consumers can filter out spoilers based on their tastes, which can assist automated systems accommodate these variability. Comprehending the impact of spoilers on enjoyment can facilitate the creation of more intricate natural language processing (NLP) models and propel the field of artificial intelligence research forward. One way to do this is to enhance the models' comprehension of human language [16], together with its context and nuances. The income a film makes at the box office might be impacted by spoilers. Movies with major plot twists or suspense aspects may do worse at the box office if important facts are widely leaked before audiences have a chance to see the picture, according to a study on the economic impact of spoilers [25]. An understanding of human behavior and preferences can be gained by researching spoilers and how to spot them. Research on how individuals handle spoilers might inform psychological and social ideas on information processing and decision-making [19].

The contradictory effects that spoilers may have on the viewing experience highlight the necessity for efficient spoiler detection systems in film reviews. While some research indicates that by increasing cognitive fluency, spoilers might increase enjoyment, other studies emphasize the detrimental impacts on suspense and emotional engagement. Creating sophisticated models for spoiler identification that take contextual data into account is essential to satisfying a wide range of user preferences and improving review systems as a whole.

2.2 What is Spoiler: The Survey

Because the definition of a spoiler is subjective by nature, it is important to survey people to learn about their opinions on spoilers. Different people have different boundaries for what constitutes a spoiler. The purpose of this part is to provide pertinent citations and research findings to support the need for this survey.

First of all, spoilers are arbitrary; what one person considers to be a spoiler may not be to another. Personal tastes, prior knowledge of the narrative, and the subject's susceptibility to plot twists are some of the variables that affect this subjectivity [15, 16]. For the purpose of creating efficient spoiler detection algorithms and improving user experience on websites that use user-generated material, like Rotten Tomatoes or IMDb, it is imperative to comprehend this diversity. Surveys are useful in capturing a wide range of perspectives, which can provide vital data for developing more sophisticated and adaptable spoiler detection methods.

Additionally, research in psychology has demonstrated that different people react differently to spoilers in terms of enjoyment. While some people believe that knowing important plot details ahead of time helps them feel less anxious and improves their comprehension of the narrative, others believe that spoilers lessen the tension and emotional connection [12, 15]. Performing a survey enables researchers to measure these preferences and investigate the underlying causes for them. Spoiler detection systems can be made more effective and tailored by using this data to customize them to meet the demands of various users.

Spoilers can also have an impact on social interactions and conversations regarding films. People who have watched a movie, for instance, could be reluctant to talk about it in public for fear of giving away spoilers. Those who haven't seen it yet might steer clear of discussions to avoid giving away any spoilers [5]. Designing technologies that enable secure and spoiler-free conversations requires an understanding of these social dynamics. Insights regarding people's social interaction skills and tactics for avoiding spoilers can be gained from surveys. Furthermore, how spoilers are viewed is greatly influenced by the environment in which they are encountered. Factors like timing, medium, and viewing habits (e.g., binge-watching vs. casual) affect the impact of spoilers [1, 9]. Researchers can create more complex algorithms that consider context when spotting spoilers by using a survey to collect thorough data on these contextual elements.

The shortcomings of the current spoiler identification techniques reinforce the necessity for surveys. The subtle and context-dependent character of spoilers is difficult for traditional text classification techniques to represent since they frequently rely on binary categorization. More sophisticated methods have demonstrated potential, such as using neural attention models or graph learning to incorporate contextual information. Grounding them in empirical facts on human views, however, can greatly increase their efficacy [1, 26]. An extensive dataset that may be utilized to train and improve these models is provided by surveys, which also give this empirical foundation. Surveys can also reveal how spoilers affect certain demographic groups emotionally. Audiences may perceive spoilers differently depending on age and culture [5]. By gathering demographic data in conjunction with responses to spoiler-related inquiries, we can discern patterns and trends that can enhance the development of spoiler detection systems that are more attuned to cultural and demographic sensitivities. This will also enable us to understand their viewpoint on spoilers.

Finally, user reliability and trust are crucial factors in spoiler detection. Customers are

more likely to believe in and utilize a system that offers precise, context-aware alerts and conforms to their own definitions of spoilers [17]. To create trust, survey people in the research process and incorporate their preferences and experiences into spoiler detection systems.

2.3 Beyond Binary Classification

Spoiler detection in autonomus systems is a multifaceted task that goes beyond straightforward binary classification. Assigning a label, such as "spoiler" or "non-spoiler," to an input is known as binary classification. However, identifying spoilers is more complex due to multiple aspects. The degree of spoilers varies depending on the situation. In certain situations, a word that might be considered spoilery could be innocuous. For example, "The butler did it" may ruin a mystery story yet be harmless in another [4, 1]. Not every spoiler is explicit. Implications might arise from subtle suggestions or incomplete details that prompt the reader to deduce crucial elements of the storyline. Such subtleties are problematic for binary classifiers [16]. The same topic may have diverse interpretations for different people. A spoiler may not always be seen in the same manner by different people. It is difficult to create a binary classifier that works for everyone due to its subjective character [6]. A possible spoiler's meaning is frequently determined by background information and the story's setting. Character deaths that conclude a major plot arc, for example, may be considered spoilers, while those that take place in a backstory or small subplot are not [4, 12].

To correctly identify spoilers, context is essential. Without it, even highly sophisticated models may miscalculate the importance of a given piece of data. It is easier to discern between important plot aspects and unimportant details when one is aware of the narrative arc. To determine if a text is a spoiler, context regarding the plot, characters, and earlier events is required [15]. Important moments in a character's journey are frequently mentioned in spoilers. Knowing the character's past and role in the tale might help determine if describing an incident is a spoiler [18]. Conventions about what qualifies as a spoiler vary throughout genres. For instance, in mystery novels, disclosing the identity of the murderer is a significant plot revelation that ruins the suspense. In contrast, one might anticipate the inevitable romantic pairing in romantic comedies. To better fit spoiler detection into genre-specific conventions, contextual knowledge is helpful [16].

Humans detect spoilers through a combination of cognitive and emotional elements. A story's element of surprise and suspense might be diminished by spoilers. It can reduce the emotional effect and engagement to know important story aspects ahead of time [12]. When people interact with stories, humans like creating narratives and figuring out puzzles. Certain people may experience decreased happiness as a result of spoilers' ability to bypass this cognitive process [15, 16]. The emotional trajectory of engaging with a narrative is pivotal. The emotional trajectory of reactions to important events can be changed by spoilers, which may lessen their intensity [18]. Individual differences in spoiler tolerance are considerable. Understanding plot aspects beforehand can improve comprehension of narrative structure and character development [6, 25].

In order for machines to efficiently identify spoilers, they need to integrate multiple sophisticated methodologies. The subtleties of human language are processed and understood using advanced NLP techniques. Parsing words, recognizing things, and comprehending semantic relationships are all necessary for this [16]. By processing vast volumes of text, models similar to transformers (BERT, GPT, etc.) are able to extract contextual

information. When compared to standard classifiers, these models are more accurate in understanding the larger narrative context and identifying potential spoilers [4, 1]. Embedding techniques that capture the contextual meaning of words and phrases in a narrative assist identify spoilers. These embeddings take into account both the context of the surrounding text and the overall narrative [16]. Accuracy can be increased by using vast datasets to train pre-trained models, then fine-tuning them for particular spoiler detection applications. These models make use of acquired language representations to enhance comprehension of the subtleties and context surrounding spoilers [4]. Human feedback is incorporated into the cycle to enhance the machine's comprehension of spoilers. In order to ensure greater accuracy and relevance, this method employs human assessments to improve and validate computer predictions [18].

2.4 Importance of Contextual Information

Contextual information is a major breakthrough in text categorization that improves overall model performance and increases the accuracy of spoiler detection. Bag-of-words (BOW) and term frequency-inverse document frequency (TF-IDF) models were frequently used in traditional text classification techniques. Although these methods had some effectiveness, they faced challenges related to the complexity of the data and a lack of comprehension of the surrounding context. These limitations hindered their capacity to accurately capture the subtle nuances of word meanings in various settings [1]. Deep learning boosted text categorization by introducing models like recurrent neural networks (RNNs) and convolutional neural networks (CNNs), which were able to capture more intricate patterns in data [1, 24].

Using graph neural networks (GNNs) to incorporate contextual information is one of the major advances in this field. The Text FCG (Fusing Contextual Information via Graph Learning) approach was presented by Wang et al. It creates a single graph for every word in a text by combining different contextual associations to identify it. This approach improves the learning effect of GNNs by adding more typed edges, which increases connectedness. When GNN and GRU are combined, the model can interact local words with global text information by improving how nodes are represented sequentially [24]. By utilizing the rich contextual information found in texts, this method shows notable gains in text categorization tasks.

Yan and Guo introduce a novel approach that utilizes contextual phrases to classify text. They employ a neural attention model for this purpose. Convolutional neural networks (CNNs) and bidirectional long short-term memory (BLSTM) networks are the foundations upon which they build various attention strategies. These models implement context information into feature representations by creating sentence and word-level attention structures. This strategy improves the model's efficiency and performance in text classification tasks by diversifying the feature information [26]. The models' ability to concentrate on the text's most pertinent passages thanks to attention processes enhances its categorization performance.

Al Sulaimani and Starkey emphasize the significance of context in text classification, particularly in the realm of brief text categorization using contextual analysis. They present a transparent text multi-class categorization system that catches the most essential words in an event's context and recognizes comparable terms in multiple contexts. Their system performs better than conventional techniques like Support Vector Machines and Naïve Bayes, especially when it comes to classifying tweets about different events [1].

This approach successfully uses contextual information to overcome the issues brought about by the briefness and ambiguity of short sentences.

Contextual embeddings from pre-trained models like BERT and GPT are being used more in NLP. Devlin et al. [8] introduced BERT, which trains on both left and right context in all layers to capture deep bidirectional representations. This approach has established new standards in numerous natural language processing (NLP) tasks, such as text categorization, by comprehending the complexities of context. BERT is especially useful for tasks like spoiler detection because of its capacity to pre-train on massive text corpora and fine-tune on specific tasks, allowing it to use vast amounts of contextual information. To further improve text categorization model capabilities, Brown et al. [3] created GPT-3, which use a transformer-based design to produce text that is both coherent and contextually relevant. Text classification tasks requiring deep contextual awareness benefit from GPT-3's complex knowledge of language, which is made possible by its autoregressive nature, which predicts the following word in a sequence.

Adding contextual data from movie scripts, summaries, and synopses can greatly improve the models' performance in the particular context of movie spoiler detection. Due to the intricacy and diversity of language employed in these media, traditional text classification techniques frequently encounter difficulties. These problems can be overcome by contextual embeddings from models such as BERT and GPT-3, which capture the complex interactions between words and their surrounding context. The narrative structure and any spoilers can be discerned from the rich contextual information included in movie scripts and plots. For example, the term "death" on its own may not suggest a spoiler, but when considered within the framework of a particular character's storyline, it becomes significantly pertinent. BERT may be customized to detect spoilers more accurately by capturing subtle elements in movie screenplays [8].

In order to offer more levels of context, scripts frequently include conversation, stage instructions, and character interactions. These scripts can be parsed and analyzed to find spoiler content using models like GPT-3, which recognize and generate human-like text. These models can determine what qualifies as a spoiler by analyzing the narrative's flow and the connections between characters [3]. Summary and synopsis summarize the film's main events and turning moments. These are very valuable for training models to detect spoilers as they frequently include essential elements of the plot. These occurrences have significance in the larger story, and contextual models can be trained to identify it. One major spoiler that could be mentioned in a summary is the discovery of a character's secret identity towards the end of the film [3].

Models like BERT and GPT-3 can learn to recognize patterns and keywords that often signify spoilers by training on a huge corpus of summaries and synopses. The models' capacity to identify spoilers is enhanced by this training, which helps them to generalize their comprehension to new literature [8]. Future research should examine hybrid approaches that mix classical and deep learning methods to improve text classification. By integrating BERT's contextual embeddings with conventional rule-based systems, a more all-encompassing solution can be achieved. BERT is capable of handling more complicated contextual understanding, whereas rule-based systems are capable of handling explicit patterns and keywords [8].

Convolutional neural networks (CNNs) are beneficial for capturing local patterns in text data, especially when it comes to comprehending context in shorter text segments, as evidenced by Kim's study on CNNs for sentence categorization [13]. Similarly, [28] Zhang and Wallace's sensitivity analysis of CNNs offered valuable information about how to best optimize these models for text classification tasks. These results imply that

by efficiently collecting both local and global context, CNNs combined with contextual embeddings from models such as BERT can improve the accuracy of spoiler detection systems.

Wadden et al. used contextual embeddings to check scientific claims, demonstrating their potential to differentiate fact from fiction [21]. This approach can be customized for identifying spoilers by training models to distinguish between crucial plot details and non-spoiler material using their contextual significance. Even with low data, models like BERT and GPT-3 can achieve high accuracy by using pre-existing information, as Gao et al. have emphasized [11]. This is because few-shot learning with pre-trained language models is crucial. In situations when labeled training data may be in short supply, this skill is critical for spoiler detection. These discoveries have useful uses in the industry and go beyond study. For example, social media networks can utilize these models in content moderation systems to automatically detect and remove spoiler content. Contextual models can classify and categorize movie reviews in aggregators and recommendation systems to advise consumers about spoilers. Streaming platforms have the ability to improve the user's experience by offering summaries and recommendations that are free of spoilers [3].

2.5 Current State of Automatic Spoiler Detection Architectures

When deep learning algorithms became available, the field of movie review spoiler detection underwent a tremendous evolution. Using complex models and massive datasets, a number of cutting-edge techniques have been developed. The study "Fine-Grained Spoiler Detection from Large-Scale Review Corpora" by Mengting Wan, Rishabh Misra, Ndapa Nakashole, and Julian McAuley advances spoiler detection. In order to identify spoilers in large review corpora at the sentence level, the researchers developed an end-to-end neural network architecture called SpoilerNet. A large dataset was compiled from Goodreads, containing more than 1.3 million book reviews accompanied by thorough spoiler comments. The dataset comprises 1,378,033 English book reviews that have been annotated specifically for spoilers at the sentence level. It was discovered through their investigation that spoilers are usually found in the later sections of reviews, are book-specific, and generally occur in clusters. Sentence encoders, word attention mechanisms, and contextual embeddings are used by SpoilerNet to improve detection accuracy and extend the hierarchical attention network (HAN) by adding item-specific information and user bias. The design integrates word embeddings with item-specific characteristics to capture the context of each review. It employs bidirectional recurrent neural networks (bi-RNN) and attention mechanisms to emphasize important words in phrases. Furthermore, it combines item and user biases to adjust predictions and models sequential dependencies among words to reflect the flow and context inside reviews. The model may be mislead by phrases that seem revelatory but may not necessarily represent spoilers, and the self-reported spoiler tags are subjective, resulting in varying data quality. To increase dataset quality and model accuracy, consider integrating crowdsourced annotations and enhancing the model's ability to distinguish between spoilery and innocuous phrases [22].

In order to tackle the issue of identifying spoilers, a separate team of researchers has devised an innovative framework known as MVSD (Multi-View Spoiler Detection). By creating three interconnected heterogeneous information networks—the movie-review subgraph, user-review subgraph, and knowledge subgraph—MVSD integrates user activity on movie

review websites with external movie knowledge. Using a unique heterogeneous graph neural network (GNN) architecture to encode and fuse the learnt representations, these networks describe a variety of input sources and their multi-view properties. Spoiler detection is made more reliable and accurate with this all-encompassing technique, which goes beyond simply examining the language of reviews. In order to improve the identification process, MVSD combines a large-scale network-based spoiler detection dataset (LCS)—much larger and more complete than earlier datasets—with an enormous movie knowledge base (UKM) that includes entries of recent films. The framework utilizes several perspectives of data, such as semantic, meta, and knowledge views, to enhance the representation of nodes in the subgraphs. MVSD is not without its limitations, despite its advances. Currently, each subgraph is modeled using the Relational Graph Convolutional Network (R-GCN). More sophisticated heterogeneous graph algorithms, such as SimpleHGN and HGT, might, nevertheless, enhance performance. The LCS dataset additionally depends on user-reported spoiler annotations from IMDB, which can be inaccurate. Using weak supervision learning techniques to increase annotation reliability and verifying these labels with expert input are the next steps toward improvement [23].

Another noteworthy method is Yiping Yang and Xiaohui Cui's BERT-Enhanced Text Graph Neural Network for Classification. This study combines BERT and Graph Neural Networks (GNN) to utilize both semantic and structural information for text categorization tasks. The BEGNN model that is being suggested creates distinct text graphs for every document and merges them with features that are retrieved from BERT. In order to extract semantic features, BERT converts each page into a network based on word co-occurrence, which captures structural information. The integration and interaction of BERT and GNN features through a co-attention module improves the model's comprehension of the text's structure and context. Nevertheless, the merged model necessitates substantial computer resources for both training and inference because to its high intensity. Additionally, because the model depends on both semantic and structural elements, performance could change for different kinds of text. Possible future enhancements could involve the development of more streamlined training techniques to decrease computing burden and conducting tests on the model across other areas to ensure wider applicability. The citation for this information is referenced as [27].

VGCN-BERT, TextING, and TextGCN are some other noteworthy spoiler detection models. Capturing intricate links between text parts, VGCN-BERT improves BERT by utilizing graph convolutional network characteristics that are taken from the whole dataset. Its efficacy on a variety of datasets may be diminished, nevertheless, as it does not fully capture the distinctive structural elements of particular texts. TextING creates a graph for each document to represent words in their unique context. The quality and consistency of the initial word embeddings, which can range greatly between texts and datasets, determine how effective the method is. While TextGCN creates a heterogeneous graph for the whole corpus, it might not be able to capture specific document structures. Instead, it captures the associations between words and documents.

Notably, spoiler detection has showed potential in the light of recent advances in contextual language models. Devlin et al. proposed the BERT model, which uses bidirectional transformers to comprehend the context of words within sentences and improves performance dramatically on a variety of NLP tasks, including spoiler identification [7]. The OpenAI GPT-3 model, trained on different internet literature, can produce and analyze sophisticated language, making it useful for spotting implicit spoilers [4]. These models need significant resources for inference and fine-tuning, nevertheless, due to their high computational cost.

Chapter 3
Dataset

This chapter provides a comprehensive analysis of the IMDb dataset used in this study, which includes user-generated movie reviews. The primary objective of the research is to explore the dataset's general characteristics, with particular attention to the distribution of reviews, reviewer activity, the popularity of movies and TV shows, rating trends, and other relevant factors. The chapter then goes on to discuss the data cleanup procedure, which is an essential step in implementation.

3.1 Dataset Analysis and Insights

The dataset includes 2,920,858 reviews in total, all of which are uniquely recognized by their `review_id`, which guarantees the uniqueness of each entry. 989,008 unique reviewers contributed these reviews, demonstrating the variety of user activity on the IMDb platform. The average number of reviews published by a reviewer is 2.95, suggesting a high frequency of participation that is essential to comprehending long-term user behavior.

The dataset covers reviews from July 27, 1998 to January 8, 2021, a broad temporal period. This two-decade span offers a longitudinal viewpoint on user reviews, allowing patterns to be analyzed over time. 42,576 distinct films and TV series are also included in the collection, as indicated by the `tconst` identifier. Because of the wide range of titles covered, a thorough examination of different genres, ratings, and other attributes is possible, giving the dataset a well-rounded content base.

3.1.1 Reviewer and Review Activity

The dataset exhibits a heterogeneous reviewer activity, with an average of 2.95 reviews per reviewer. There is a fraction of users who are very active, even though the majority offer a few reviews. The leading five critics are:

- **SnoopyStyle**: 8,287 reviews
- **MartinHafer**: 6,417 reviews

- **bkoganbing**: 6,023 reviews
- **Leofwine_draca**: 5,831 reviews
- **claudio_carvalho**: 3,923 reviews

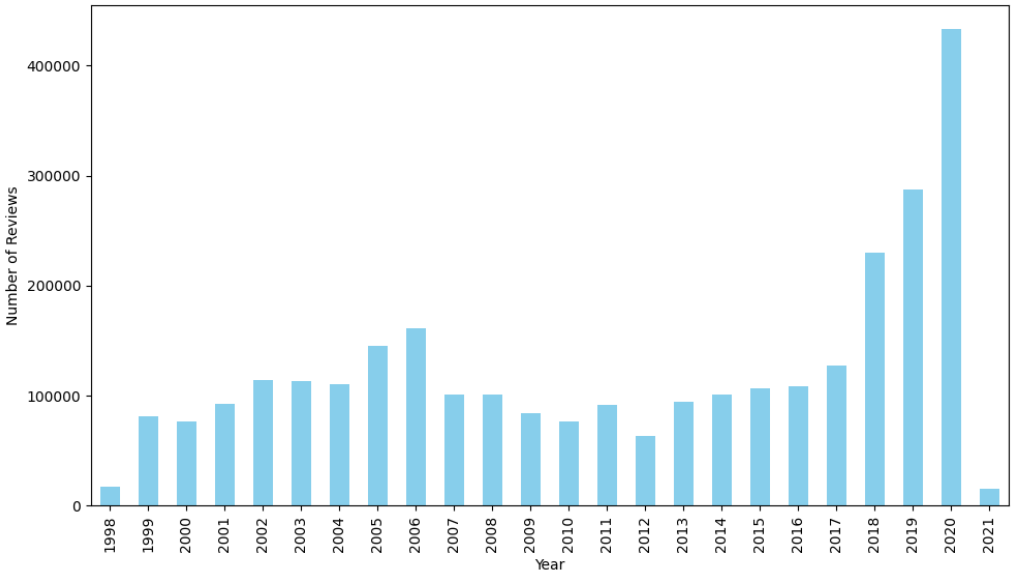

Figure 3.1: Number of Reviews Per Year

These reviewers are particularly influential within the dataset, as their prolific reviewing habits provide a plethora of information that could shape overall ratings and trends. Conversely, the average number of reviews per movie/show is 68.60, which indicates a consistent level of engagement with various titles. Figure 3.1 provides a visual representation of the review distribution per year. It is worth noting that the most highly regarded films and television programs are as follows:

- **Avengers: Endgame (tt4154796)**: 8,703 reviews
- **The Shawshank Redemption (tt0111161)**: 8,084 reviews
- **The Mandalorian (tt8110330)**: 7,679 reviews
- **Avengers: Infinity War (tt4154664)**: 7,150 reviews
- **Tenet (tt7126948)**: 6,796 reviews

The high review counts of these titles indicate that they are among the most popular and culturally significant works. The global impact and the pervasive discussion they have generated are indicated by the concentration of reviews around specific titles.

3.1.2 Ratings and Helpfulness

The average rating across all reviews in the dataset is 6.71, indicating a generally favorable perception among users, as the mean rating is above the midpoint of 5 on a 10-point scale. To find any biases or trends in user ratings, additional analysis of the rating distribution will be done in the following sections. However, figure 3.2 provides an overview of rating distribution. From this, we can see that most of the movies are highly rated by users.

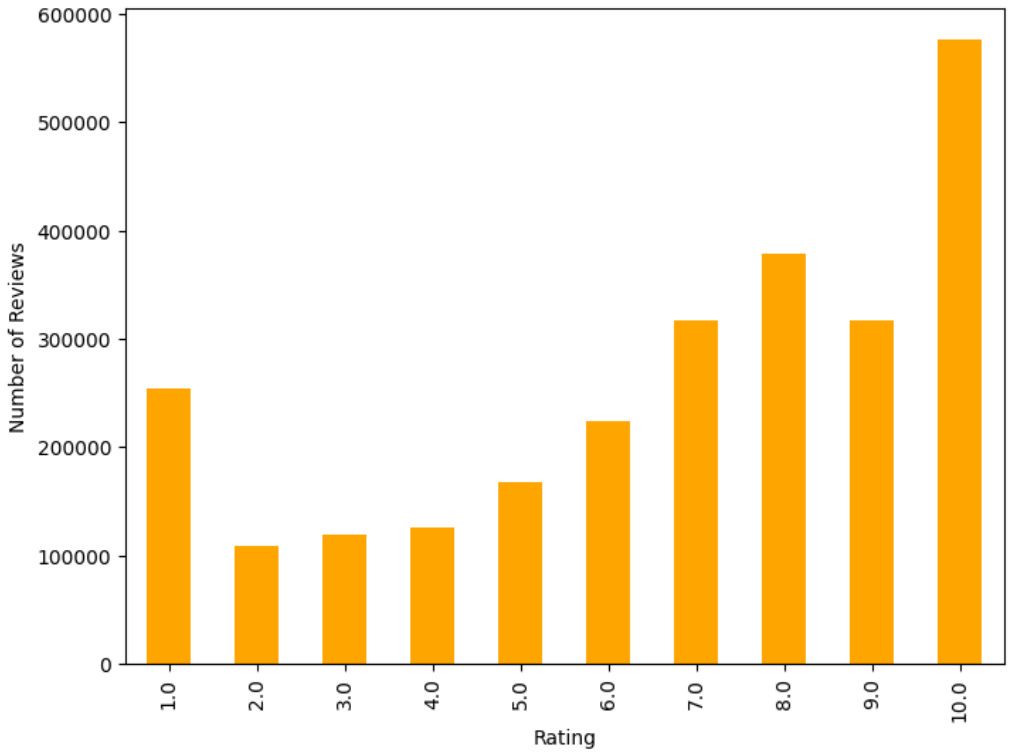

Figure 3.2: Distribution of Ratings

The average review helpfulness score, which is determined by counting the number of `helpful_votes`, is 7.31. This shows that, on average, seven users thought a particular review to be beneficial. Analyzing user interaction with reviews and determining which reviews appeal to the audience the most requires an understanding of the helpfulness vote distribution.

3.1.3 Genres, Language, and Spoiler Tags

The dataset includes reviews from a wide range of genres, with the most common being:

- **Drama**: 1,498,954 reviews
- **Action**: 939,756 reviews
- **Comedy**: 783,141 reviews

- **Adventure**: 694,364 reviews
- **Crime**: 520,504 reviews

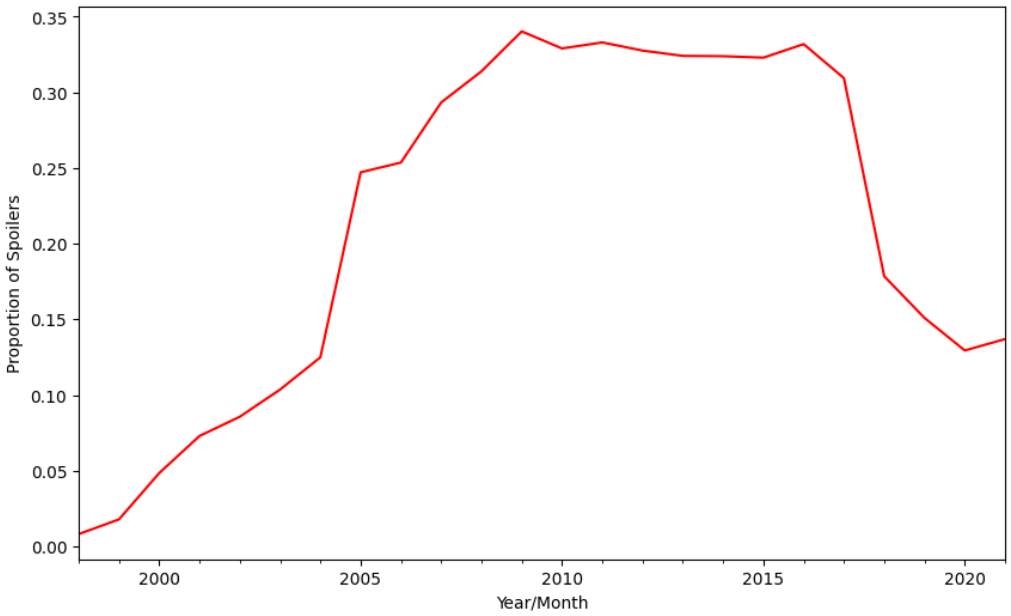

Figure 3.3: Spoiler Trends Over Time

The genre that gets reviewed the most is drama, which is followed by action and humor. This distribution is a reflection of larger trends in television and film material, where these genres are more popular in terms of both production value and audience interest. The average ratings by genre can also be analyzed using this dataset. As an illustration, the genre "Action" has an average rating of 5.32, although genres like "Action, Adventure, and Animation" typically have higher average ratings (e.g., 7.59). Understanding audience expectations and satisfaction with various material kinds can be gained by examining ratings that are particular to a given genre.

The vast majority of reviews (2,930,525) are written in English (`en`) in terms of language. Given that IMDb's major audience is in English, the dataset's English language dominance is predicted. It also implies that there aren't enough reviews in languages other than English, which may be a topic for platform improvement in the future. Moreover, spoilers are flagged in about 20.67% of the reviews. This large percentage suggests that many people provide intricate story details that can ruin the viewing experience for other viewers. Figure 3.3 provides a visual representation of spoiler tag over time. Examining spoiler patterns over time may provide information about how the use of spoilers has changed, maybe as a result of platform regulations or changes in the way people talk about material.

3.1.4 Review Summary and Detail Length

The dataset reveals interesting statistics concerning review lengths:

- **Average Summary Length**: 5.64 words
- **Average Detail Length**: 210.91 words

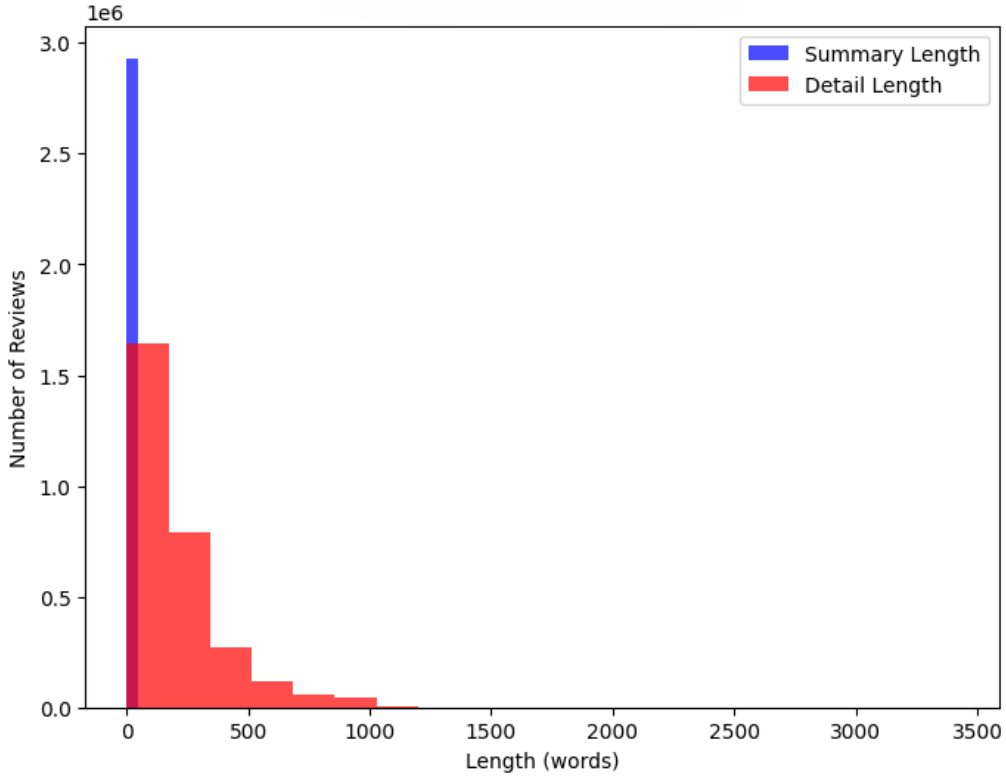

Figure 3.4: Distribution of Review Lengths

These numbers demonstrate that while users usually write comprehensive assessments with an average length of 211 words, their review summaries are typically shorter—often containing only a few lines. The difference in length between the summary and the detail portion implies that individuals use the summary to condense the main points of their review, saving the detailed section for a more in-depth examination or perspective. Figure 3.4 provides more insight about this statistics.

3.1.5 Implications

Numerous important insights that can guide both academic research and real-world applications are revealed by the intensive dataset analysis. Notable aspects include the spread of reviews among genres, the activity level of specific reviewers, and the frequency of spoiler tags. These elements draw attention to how intricate user interaction is on IMDb and how several ways users contribute to the site exist.

Reviews tend to cluster around particular well-known titles, which highlights the cultural relevance of these works and reflects their influence on audiences around the world. Furthermore, content producers and marketers may find interest in the insightful information

that the genre-specific rating analysis offers on how various kinds of material are viewed. The majority of reviews are written in English, indicating a potential development area in non-English regions where user experience might be improved by more locally relevant content and reviews. The results pertaining to spoiler tags also highlight a potential area for IMDb to improve its user interface in order to better govern spoiler content—possibly by providing users with more precise controls.

3.2 Review Statistics

This section provides an in-depth statistical analysis of the reviews within the IMDb dataset. We can gain valuable insights into user behavior, content reception, and the interactions between these factors by examining the distribution of ratings, review lengths, spoiler tags, and helpfulness of reviews.

3.2.1 Distribution of Ratings

The dataset exhibits a diverse distribution of ratings across the spectrum from 1 to 10. The frequency of each rating is as follows:

- **Rating 1**: 254,782 reviews
- **Rating 2**: 108,165 reviews
- **Rating 3**: 119,671 reviews
- **Rating 4**: 126,039 reviews
- **Rating 5**: 168,048 reviews
- **Rating 6**: 224,224 reviews
- **Rating 7**: 316,978 reviews
- **Rating 8**: 379,193 reviews
- **Rating 9**: 317,192 reviews
- **Rating 10**: 576,582 reviews

This distribution suggests a tendency toward higher ratings, with a mode of 10 being the most frequently occurring rating. This skewness may be indicative of a selection bias in user participation, or it may reflect a propensity for users to rate content they notably enjoyed more frequently. The average rating of 6.71 across all evaluations indicates that users have a generally favorable opinion of the content. The ratings' standard deviation is 2.94, which suggests a robust range of opinions and moderate variability. The observation that users most frequently provide the highest possible score, which is 10, is further reinforced by the mode of 10. This likely reflects their engagement with content that they are impassioned about. Refer to figure 3.2.

Nevertheless, it is crucial to acknowledge that 11.59% of the total reviews, or 339,651, lack a rating. The absence of data may be attributable to prior reviews that did not require ratings or instances in which users chose not to rate the content. This missing data should be taken into account in analyses that rely on comprehensive rating information, as it has the potential to introduce bias or restrict the interpretability of specific findings.

3.2.2 Review Length Analysis

Users typically provide concise overviews or titles for their reviews, as demonstrated by the average length of review summaries at 5.64 words. In contrast, the detailed reviews have an average length of 210.91 words, suggesting that users are more expressive in this section, providing comprehensive feedback or analyses of the content.

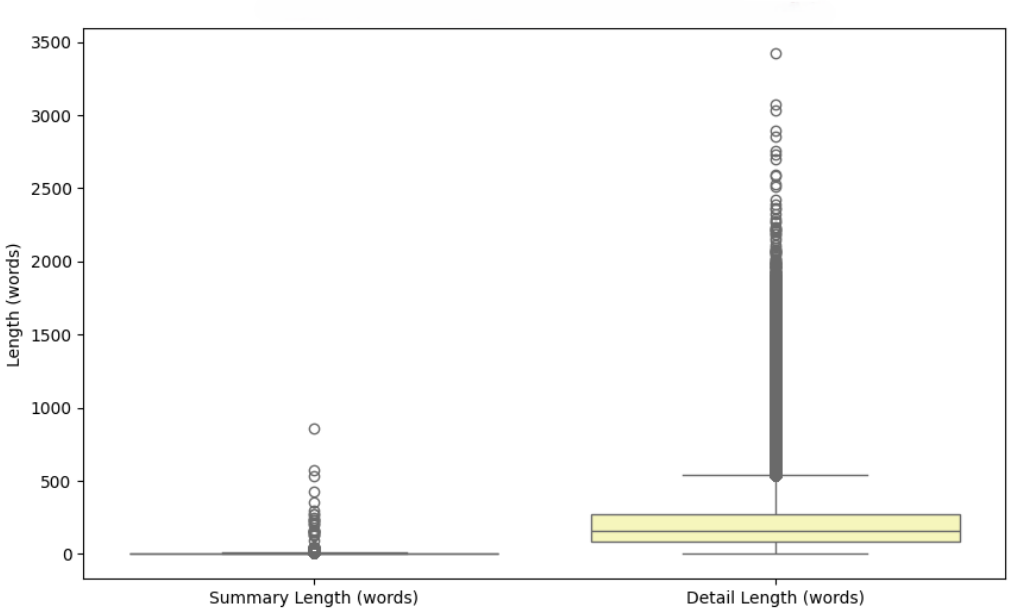

Figure 3.5: Distribution of Review Lengths (Summaries vs. Details)

The distribution of review lengths is as follows (see figure 3.5):

- **Summary Length**: Mean = 5.64 words, Standard Deviation = 3.87 words, Minimum = 1 word, Maximum = 856 words
- **Detail Length**: Mean = 210.91 words, Standard Deviation = 191.57 words, Minimum = 1 word, Maximum = 3,422 words

These distributions indicate that the duration of detailed reviews varies significantly, despite the fact that the majority of users maintain brief summaries. The maximum observed lengths indicate that certain users provide exceptionally comprehensive analyses, offering in-depth perspectives on the content. This discrepancy between the duration of the summary and the detail section implies that users utilize the summary to summarize the core of their review, while the detailed section facilitates a more in-depth discussion.

3.2.3 Spoiler Tag Analysis

A substantial number of reviews contain information that could reveal crucial plot points or outcomes, as approximately 20.67% of them are tagged as spoilers. The frequency of spoilers differs among various ratings, with a higher proportion of spoiler-tagged reviews

in lower ratings (1-5). For instance, spoilers are designated in 21.68% of reviews with a rating of 1, while only 15.29% of reviews with a rating of 10 are marked as spoilers (see figure 3.6).

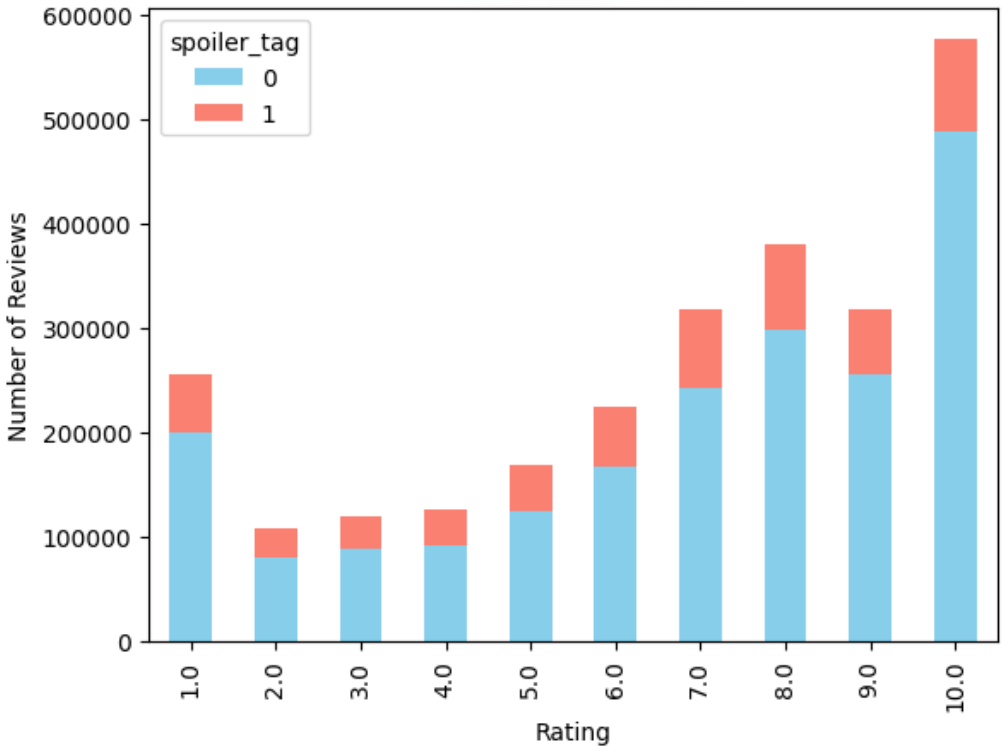

Figure 3.6: Spoilers by Rating

This trend indicates that users who give low ratings to content are more inclined to include spoilers, which may be a result of frustration or a desire to critique specific narrative elements. In contrast, users who found the content enjoyable may be more inclined to preserve the viewing experience for others, as evidenced by the correlation between higher ratings and fewer disclosures.

The percentage of spoilers is also contingent upon the duration of the review. Spoilers are less likely to be included in reviews with shorter summaries and details, while lengthier reviews are more likely to contain them. The average spoiler proportion for summary length is 23.14%, while for detail length, it is 44.34%. This correlation between spoilers and review length implies that users are more inclined to discuss narrative elements in detail as they provide more comprehensive feedback, thereby increasing the likelihood of revealing spoilers.

3.2.4 Review Helpfulness Analysis

According to the `helpful_votes` metric, the average helpfulness score is 7.31, which suggests that approximately seven users found a specific review to be helpful. The

helpfulness scores exhibit a wide range of variability, with a standard deviation of 33.42 votes, a minimum of 0 votes, and a maximum of 12,652 votes (more details in figure 3.7). This variation suggests that, despite the fact that the majority of evaluations receive a modest number of helpful votes, a small number of them are particularly influential.

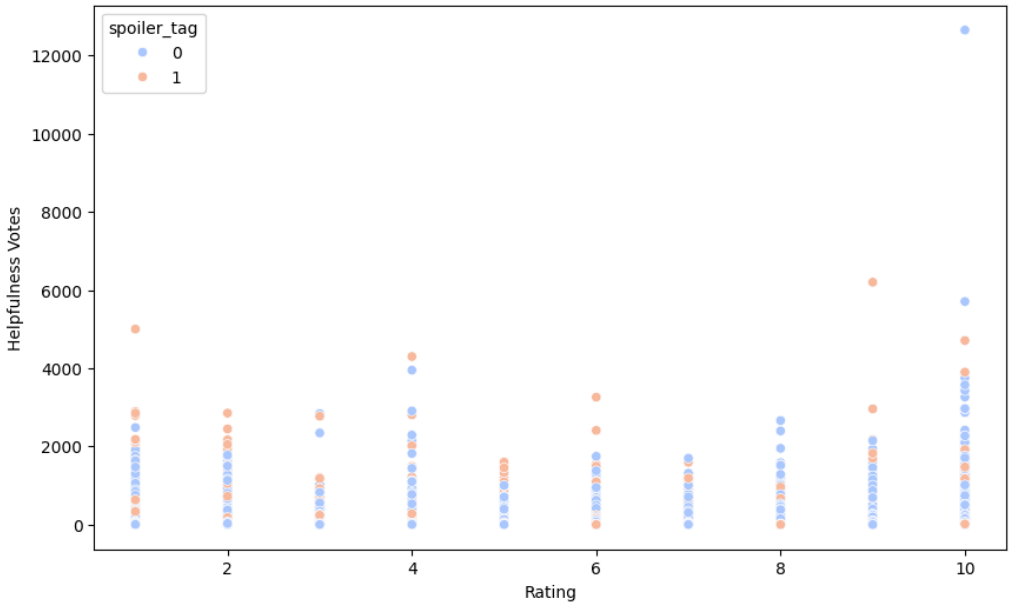

Figure 3.7: Helpfulness vs. Rating (Colored by Spoiler Tag)

The relationship between helpfulness and rating is marginally negative, with a value of -0.0399. This implies that higher ratings do not necessarily indicate greater helpfulness. This could suggest that the content and substance of the review are more important factors in determining its usefulness than the rating itself. In the same vein, the correlation between helpfulness and the presence of a spoiler tag is exceedingly feeble, at 0.0039, suggesting that spoilers do not have a substantial impact on the perceived helpfulness of a review. This discovery implies that users derive value from evaluations regardless of spoilers, potentially employing spoiler tags as a precautionary measure rather than a deterrent.

3.2.5 Implications

There are numerous intriguing patterns in user behavior and content reception that have been uncovered through this analysis of review statistics. The substantial proportion of spoiler-tagged reviews, the bias toward higher ratings, and the variability in helpfulness scores all contribute to a nuanced understanding of how users interact with content on IMDb.

According to the negative correlation between helpfulness and rating, reviews are more highly regarded for their content than their rating, which suggests that users value detailed and considerate analyses. In the same vein, the absence of a robust correlation between helpfulness and spoilers implies that users are not discouraged from obtaining value from

a review, provided that the review is well-crafted.

These discoveries have significant implications for the design of review systems on platforms such as IMDb, particularly in the areas of spoiler content management and the encouragement of more detailed reviews. Platforms can improve the overall user experience and better meet the requirements of their audience by comprehending these user behaviors.

3.3 Reviewer Statistics

This section provides a detailed analysis of the behavior and characteristics of reviewers within the IMDb dataset. We can gain a deeper understanding of user engagement on the platform by examining the top reviewers, the distribution of reviews per reviewer, and their rating consistency. Additionally, this section explores how reviewer activity has evolved over time which offers insights into trends and patterns in review submission.

3.3.1 Top Reviewers and Overall Engagement

The dataset reveals a small group of highly active reviewers who have contributed a substantial number of reviews, significantly influencing the platform's discourse. The top five reviewers (see the trend on figure 3.8), based on the number of reviews submitted:

- **SnoopyStyle**: 8,287 reviews
- **MartinHafer**: 6,417 reviews
- **bkoganbing**: 6,023 reviews
- **Leofwine_draca**: 5,831 reviews
- **claudio_carvalho**: 3,923 reviews

The reviews of these reviewers are particularly influential not only because of the volume of their contributions but also because they are likely to reach a wide audience. The existence of such prolific evaluators indicates a devoted user base that consistently interacts with the platform by offering feedback on a diverse array of content.

Approximately 2.95 reviews have been submitted by each reviewer in the dataset on average. The general user engagement is represented by this figure, which indicates that the majority of users contribute a comparatively small number of reviews. Nevertheless, the average is considerably influenced by a small number of highly active reviewers, underscoring the concentration of activity among a small group of users.

3.3.2 Distribution of Reviews per Reviewer and Rating Consistency

The distribution of reviews per reviewer exhibits significant variability:

- **Mean**: 2.96 reviews
- **Standard Deviation**: 29.70 reviews

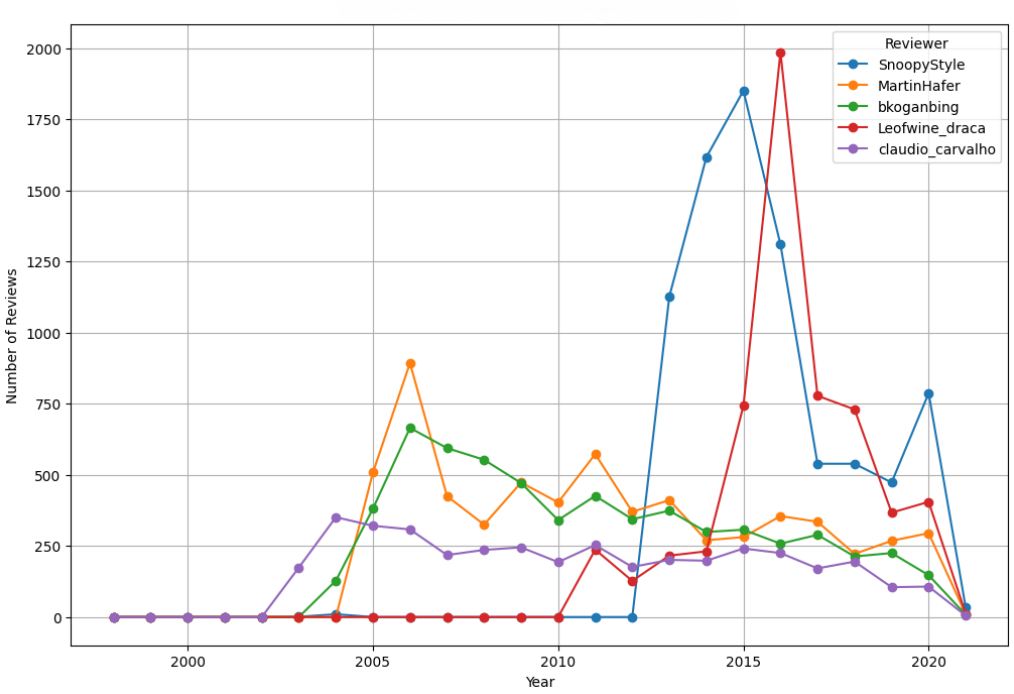

Figure 3.8: Reviewer Activity Over Time (Top 5 Reviewers)

- **Minimum**: 1 review
- **Maximum**: 8,287 reviews

Most reviewers contribute only one or two reviews, as indicated by the 25th, 50th, and 75th percentiles, all at or below two reviews. This implies that most users use the platform on a sporadic basis. Nevertheless, the high standard deviation and the maximum value indicate a significant long tail of highly active reviewers who contribute considerably more reviews. This distribution is indicative of user-generated content platforms, in which a small number of users are responsible for a substantial portion of the content.

Regarding rating consistency, the standard deviation of ratings across reviewers is used as a measure:

- **Mean Standard Deviation**: 2.07
- **Standard Deviation**: 1.68
- **Minimum**: 0.00
- **Maximum**: 6.36

A mean standard deviation of 2.07 suggests that, on average, reviewers exhibit moderate variability in their ratings. Some reviewers are highly consistent, with a standard deviation close to zero, indicating similar ratings across different movies/shows. In contrast, other reviewers display a higher standard deviation, reflecting a broader range of preferences or a more critical engagement with content.

3.3.3 Reviewer Activity Over Time

The evolution of reviewer activity over time shows several notable trends:

- **1998-2000**: Reviewer activity was minimal, with few users contributing reviews. The mean number of reviews per reviewer was very low, and overall engagement on the platform was limited.
- **2001-2005**: A gradual increase in activity is observed, with more reviewers contributing each year. By 2005, the mean number of reviews per reviewer had increased, and the maximum number of reviews by a single reviewer in a year reached 754.
- **2006-2012**: Continued growth in reviewer activity, with peak engagement occurring in some years. This period marks a diversification in reviewer engagement, with a steady increase in the number of reviews per reviewer.
- **2013-2017**: Reviewer activity fluctuated, with some years showing increased engagement and others a slight decline. However, the overall trend remained positive, with significant contributions from top reviewers.
- **2018-2020**: A sharp increase in reviewer activity is observed, particularly in 2020, which saw the highest mean reviews per reviewer. The maximum number of reviews by a single reviewer in 2020 was 1,225, likely reflecting increased home viewing during the COVID-19 pandemic.
- **2021**: A sharp decline in reviewer activity is noted, possibly due to the dataset capturing only partial data for the year or changes in user behavior following the 2020 peak.

According to these trends, user engagement has generally increased over time, with substantial growth in recent years. In particular, the significant increase in activity in 2020 may be attributed to the COVID-19 pandemic's impact and broader trends in media consumption, which resulted in an increase in the amount of time spent at home and participation in online content.

3.3.4 Implications

Reviewer statistics analysis demonstrates significant patterns in user engagement on IMDb. The substantial influence that a small number of highly active users have in influencing the discourse surrounding movies and shows is emphasized by the concentration of reviews. Especially when these evaluators are perceived as authoritative voices by the community, their consistency in rating, or lack thereof, can impact the overall perception of content.

The proliferation of streaming services and global trends in media consumption are consistent with the increase in reviewer activity over time, particularly during the late 2010s. For content creators and platform developers who are interested in improving their engagement with their audience, comprehending these trends can offer valuable insights.

3.4 Movie/Show Statistics

This section delves into the attributes of the most frequently reviewed films and television programs in the IMDb dataset. By examining the frequency of reviews, average ratings,

spoiler proportions, and helpfulness scores, we can obtain a better understanding of how specific titles engage audiences and how they are perceived in terms of quality and content disclosure.

3.4.1 Most Reviewed Movies/Shows and Engagement

The dataset emphasizes numerous films and television programs that have attracted substantial attention from critics, thereby demonstrating their cultural influence and popularity. The titles that have received the most reviews are as follows:

- **Avengers: Endgame (tt4154796)**: 8,703 reviews
- **The Shawshank Redemption (tt0111161)**: 8,084 reviews
- **The Mandalorian (tt8110330)**: 7,679 reviews
- **Avengers: Infinity War (tt4154664)**: 7,150 reviews
- **Tenet (tt7126948)**: 6,796 reviews

Some of the most popular and extensively discussed works within the dataset are represented by these titles, which have garnered substantial viewer engagement. The diversified interests of IMDb users are reflected in the prominence of critically acclaimed television series such as "The Mandalorian" and blockbuster films like "Avengers: Endgame." The substantial quantity of reviews for these titles suggests that audiences have a strong desire to discuss and critique these works, in addition to their widespread viewing.

3.4.2 Average Rating and Rating Distribution

The dataset reveals that certain titles have achieved exceptionally high average ratings, with some even receiving a perfect score. The top five titles with an average rating of **10.0** are:

- **Orfeu Negro (1959) (tt2108517)**
- **Z (1969) (tt0131494)**
- **Aghet - Ein Völkermord (2010) (tt12945742)**
- **Wings (1983) (tt0092218)**
- **Sherlock Holmes (1984) (tt5540196)**

These perfect ratings suggest that these titles have been highly appreciated by a small, possibly niche, audience. It is important to note, however, that a perfect average rating often reflects a limited number of reviews, which can skew the perception of the title's overall reception. Such high ratings may also be influenced by the specific audience demographic that engages with these titles, indicating a strong affinity among particular user groups.

3.4.3 Spoiler Proportion in Reviews

After conducting an analysis of spoiler proportions, it has been determined that specific titles have a spoiler proportion of **100%**. This implies that each review for these titles contains spoiler content. Reviewers are likely to engage in a detailed discussion of these elements, as the content of these titles is likely to be heavily reliant on plot twists or critical developments. The significance of spoiler tags for these titles is underscored by the high spoiler proportions, which guarantee that potential viewers are amply informed prior to reading the reviews. This trend underscores the importance of platforms carefully managing spoiler-sensitive content to safeguard the viewing experience.

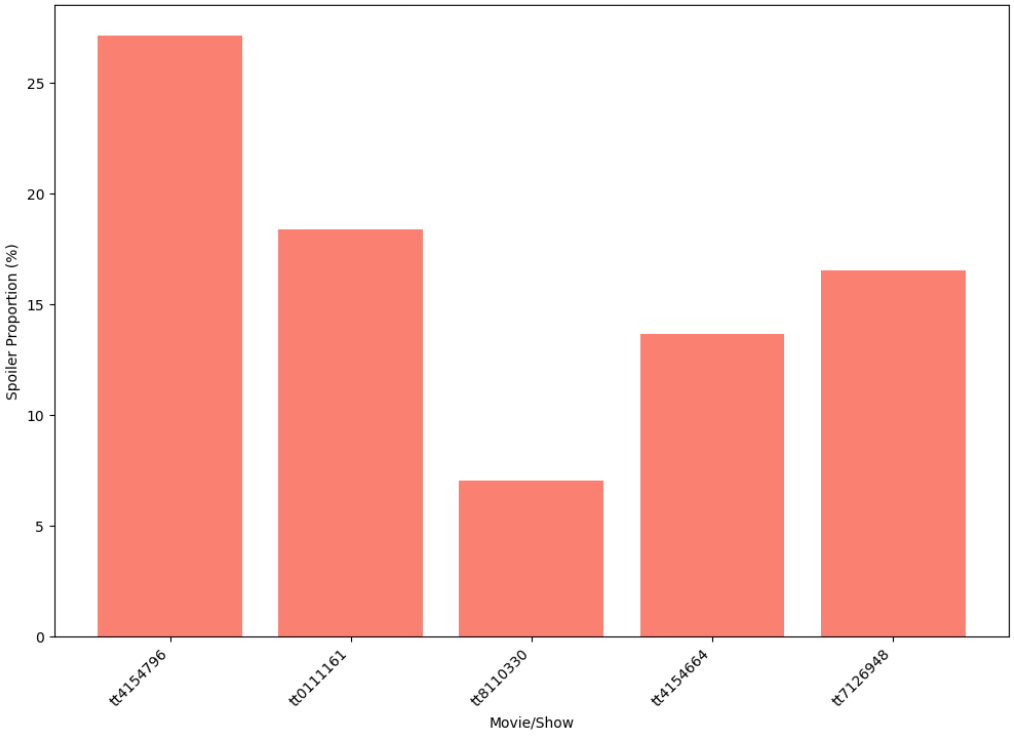

Figure 3.9: Top Movies by Spoiler Proportion

The five titles with the highest spoiler proportions (see figure 3.9) are as follows:

- **The Metamorphosis of Mr. Samsa (1977) (tt1519457)**
- **Angel Guts: Red Classroom (1979) (tt0082153)**
- **The Big Turnip (2002) (tt2411024)**
- **Banana Joe (1982) (tt0082107)**
- **The Rapture (1991) (tt0135157)**

3.4.4 Helpfulness of Reviews by Title

The analysis of helpfulness scores reveals that certain titles have reviews that are particularly valued by the community. Reviews with high helpfulness scores often include detailed analyses, well-articulated opinions, or crucial information that assists other users in making informed viewing decisions. The high helpfulness of these reviews may reflect their quality and the depth of engagement from both the reviewers and the readers.

The top five titles with the highest average helpfulness scores are:

- **A Teacher (2013) (tt7662752)**: 731.44 helpful votes (on average)
- **The Night Of (2016) (tt4693464)**: 315.24 helpful votes
- **The Man in the High Castle (2015) (tt5537140)**: 207.00 helpful votes
- **A Scanner Darkly (2006) (tt0994917)**: 162.24 helpful votes
- **Love/Hate (2010) (tt3576728)**: 160.28 helpful votes

3.4.5 Implications

The analysis of movie/show statistics reveals several key trends in the manner in which specific titles engage audiences and encourage interaction on IMDb. The cultural significance and widespread appeal of blockbuster films and critically acclaimed series are underscored by their prominence among the most reviewed titles. Indicative of the diversity of tastes within the IMDb community, the presence of perfect ratings, particularly for lesser-known titles, implies that specific works resonate profoundly with specific user groups.

Additionally, the imperative for effective spoiler management on review platforms is underscored by the high spoiler proportions of specific titles, particularly those that heavily rely on narrative developments. Lastly, the evaluation of helpfulness scores reveals that specific reviews are highly regarded for their clarity and profundity, which offers valuable advice to other users.

These insights can be invaluable for content creators, marketers, and platform developers who are seeking to enhance user experience and comprehend audience engagement. The results indicate that certain titles generate substantial discourse and attract widespread attention, while others resonate deeply with smaller, more niche audiences. Each piece of user-generated content on IMDb contributes uniquely to the overall landscape.

3.5 Review Words Analysis

Review textual features offer important information on user interaction, the quality of comments, and how reviews could affect readers' perceptions. This part explores the word count statistics of reviews, looking at different aspects like average lengths, distributions, and connections with other review qualities to give a more comprehensive picture of how consumers interact with the content through writing.

3.5.1 Word Count Statistics

Average Word Count per Review

The dataset reveals distinct differences in the length of review summaries and detailed reviews:

- **Average Summary Word Count**: 5.64 words
- **Average Detail Word Count**: 210.91 words

These graphs show that full reviews provide more in-depth input, with an average length of 211 words, but review summaries are more succinct, usually summarizing the review in about six words. This pattern implies that reviewers want to deliver attention-grabbing, brief summaries, with more in-depth analysis or comments in the detailed part.

Distribution of Word Counts

The distribution of word counts reveals variability in how users engage with reviews:

- **Summary Word Count**:
  - Mean = 5.64 words
  - Standard Deviation = 3.87 words
  - Minimum = 1 word
  - 25th Percentile = 3 words
  - Median (50th Percentile) = 5 words
  - 75th Percentile = 7 words
  - Maximum = 856 words
- **Detail Word Count**:
  - Mean = 210.91 words
  - Standard Deviation = 191.57 words
  - Minimum = 1 word
  - 25th Percentile = 86 words
  - Median (50th Percentile) = 154 words
  - 75th Percentile = 269 words
  - Maximum = 3,422 words

The majority of summaries are brief, with 75% containing seven words or fewer. There have been some exceptionally lengthy summaries, with a maximum length of 856 words, nevertheless. Detailed reviews, on the other hand, vary greatly, from very short remarks (1 word) to lengthy criticisms that reach 3,400 words. The wide range of reviewing styles, from succinct observations to in-depth analysis, is indicated by the significant standard deviation of extensive reviews.

3.5.2 Word Count by Movie/Show

The analysis of word counts by movie or show reveals differences in how certain titles inspire longer or shorter discussions. For instance, the following sample of titles illustrates this variability:

IMDb ID	Average Summary Word Count	Average Detail Word Count
tt0000574	10.38	431.00
tt0002461	4.50	209.00
tt0003131	2.50	247.00
tt0003419	6.59	203.29
tt0003442	9.00	30.00

Table 3.1: Word Count by Movie/Show

In both summaries and thorough evaluations, **tt0000574** stands out as having a substantially higher average word count, indicating that it stimulates more in-depth discussions. In contrast, **tt0003442** has abnormally short detailed evaluations but larger summary word counts, suggesting a potential preference for condensed commentary on this title.

3.5.3 Word Count by Reviewer

Analyzing word counts by reviewer highlights variations in individual reviewing styles. A sample of five reviewers shows diverse approaches:

Reviewer	Average Summary Word Count	Average Detail Word Count
!@N	3.29	111.29
”Ace” Rothstein	4.00	132.00
”Bandit”	10.50	98.50
”EO”	5.33	366.33
”Garfield”	6.00	77.00

Table 3.2: Word Count by Reviewer

For instance, **”EO”** tends to write longer, more detailed reviews with an average word count exceeding 366 words, while **”Bandit”** provides longer summaries but shorter detailed reviews. Such differences in reviewing styles can influence how readers engage with and perceive content.

3.5.4 Word Count by Rating

Analyzing the correlation between word counts and review ratings reveals patterns in the ways that users express their opinions:

For mid-range ratings (4-7), detailed appraisals are longer; for ratings 6 and 7, they peak at about 244 words. It’s interesting to note that evaluations with extreme scores (1 and 10) tend to be shorter, which implies that strongly held opinions, whether favorable or unfavorable, may be communicated more succinctly because of the depth of emotion behind them.

Rating	Average Summary Word Count	Average Detail Word Count
1	5.21	149.62
2	5.32	186.69
3	5.47	206.94
4	5.67	225.82
5	5.82	232.15
6	6.06	244.59
7	6.06	244.39
8	5.87	234.87
9	5.66	216.26
10	5.34	173.16

Table 3.3: Word Count by Rating

3.5.5 Word Count by Spoiler Tag

Reviews containing spoilers tend to be longer than non-spoiler reviews:

Spoiler Tag	Average Summary Word Count	Average Detail Word Count
0 (No)	5.53	184.58
1 (Yes)	6.07	311.97

Table 3.4: Word Count by Spoiler Tag

Spoiler reviews are far lengthier than non-spoiler reviews; detailed reviews average about 312 words, whereas non-spoiler reviews average 185 words. This discrepancy implies that reviewers who provide plot details go into greater detail and analysis, which calls for longer stories.

3.5.6 Implications

Analysis of the word count of reviews offers insightful information on user interaction and the depth of criticism. The differing lengths of summaries and full reviews illustrate the dual function of reviews, which is to draw readers in quickly while providing more in-depth analysis in portions that are longer.

The correlation between word counts and ratings also implies that while strong positive or negative views may be communicated more succinctly, intermediate ideas frequently spark more in-depth conversations. The significant word count disparity between reviews that contain spoilers and those that do not highlights the difficulty of discussing content that contains plot twists and the influence of spoilers on review structure.

These results have applications for content aggregators such as IMDb. Comprehending user behavior with respect to word counts can help design the platform in a way that best suits various audience preferences. For example, it can optimize how reviews are presented based on length, relevancy, and spoiler content.

3.6 Textual Complexity Analysis

One of the most important factors in determining how readable and interesting the content is for various audiences is the textual complexity of reviews. I concentrated on two important measures in my analysis: **Lexical Diversity** and **Readability Scores**. These measures were selected because they can shed light on the review's level of comprehensibility and lexical richness.

3.6.1 Readability Scores

A text's **Readability** is a measurement of how simple it is to read and comprehend. The **Flesch Reading Ease** and **Flesch-Kincaid Grade Level** ratings are well-known readability analysis techniques that I used for the purpose of my study.

- **Flesch Reading Ease**: Text is rated on a 100-point scale by this score, with higher values denoting easier readability. For the majority of information intended for a wide readership, a score of 60 to 70 is deemed appropriate [10].
- **Flesch-Kincaid Grade Level**: The result given here denotes the grade level of the text that is necessary for comprehension in a U.S. school. A score of 8.0, for instance, indicates that a student in the eighth grade should be able to comprehend the material. In order to make sure that the content is appropriate for the intended audience's reading level, this metric is frequently employed in educational contexts [14].

Summary of Findings:

- **Mean Flesch Reading Ease**: 71.55
- **Mean Flesch-Kincaid Grade Level**: 7.84

The findings indicate that the reviews are generally written at a level that is comprehensible to a wide range of readers, with the average reader requiring a reading ability equivalent to that of a seventh or eighth grade individual. There is some variety in readability, as indicated by the standard deviations for these metrics; some reviews are noticeably easier or harder to read than others.

Scientific Reasoning: The Flesch Reading Ease and Flesch-Kincaid Grade Level were selected due to their widespread recognition in readability studies and their reliability as indices of text difficulty. For websites like IMDb that serve a varied user base, these techniques offer practical insights on how well the reviews are suited to the wider public.

3.6.2 Lexical Diversity

A text's **Lexical Diversity** is the variety of terms that are employed in it. It is a crucial indicator of word richness and can show how complex and varied the reviews' language is. I determined the lexical variety for this analysis by dividing the total number of words in each review by the number of unique words.

This shows that 72% of the words in a review are unique on average (more details on figure 3.10). A high score for lexical diversity indicates that reviewers are utilizing a wide range of terms, which can improve the content's expressiveness and depth.

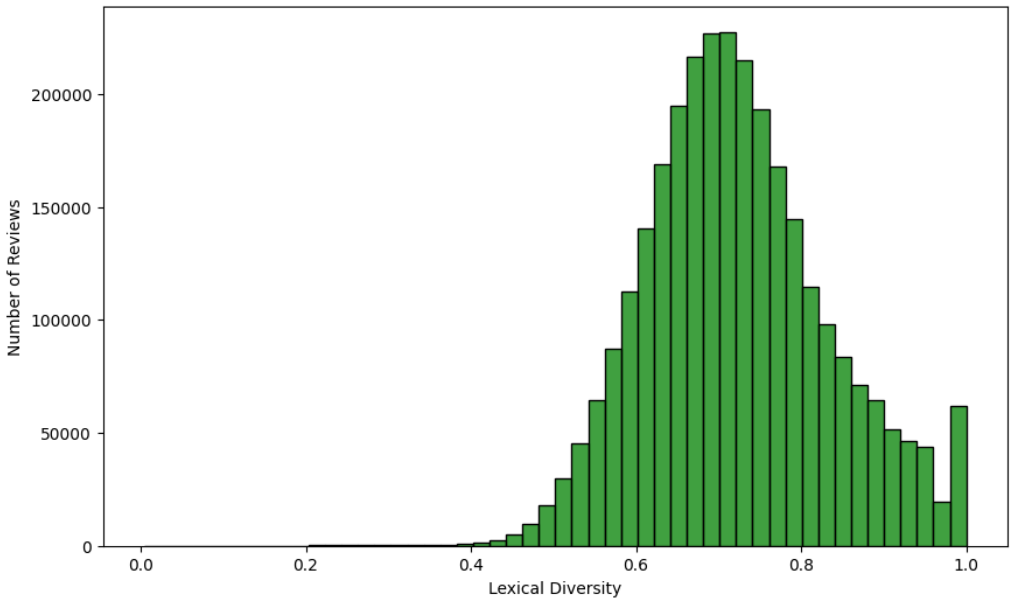

Figure 3.10: Distribution of Lexical Diversity in Reviews

3.7 Spoiler Proportion Analysis

This section offers a thorough examination of the proportion of spoilers in reviews, looking at the frequency of spoilers in different shows, films, genres, ratings, and other categories. The analysis provides insights into patterns and behaviors on the IMDb platform by illuminating the ways in which specific content genres or reviewers are more likely to provide spoilers.

3.7.1 Spoiler Proportion by Movie/Show

The study revealed that numerous films and television programs had a spoiler proportion of 100%, suggesting that all evaluations for these titles contained spoilers. Given the high percentage of spoilers, it is likely that these books have important story points or twists that reviewers will feel forced to share.

- **Top Movies/Shows by Spoiler Proportion:**
  - **Title with ID tt1519457**: 100% spoilers
  - **Title with ID tt0082153**: 100% spoilers
  - **Title with ID tt2411024**: 100% spoilers
  - **Title with ID tt0082107**: 100% spoilers
  - **Title with ID tt0135157**: 100% spoilers
  - **Title with ID tt0068892**: 100% spoilers
  - **Title with ID tt0363494**: 100% spoilers

  - **Title with ID tt0046711**: 100% spoilers
  - **Title with ID tt0033195**: 100% spoilers
  - **Title with ID tt2066857**: 100% spoilers

The narrative structure of these titles may facilitate spoiler-heavy discussions or be significantly reliant on plot developments.

3.7.2 Spoiler Proportion by Genre

Certain genres are more prone to spoilers, as indicated by their higher spoiler proportions. The analysis highlights the following genres with the highest spoiler proportions:

- **Top Genres by Spoiler Proportion:**
  - **News**: 29.44%
  - **Film-Noir**: 27.80%
  - **Sci-Fi**: 24.29%
  - **Adult**: 24.26%
  - **Horror**: 24.25%
  - **Western**: 23.02%
  - **Mystery**: 22.99%
  - **Thriller**: 22.56%
  - **Adventure**: 22.06%
  - **Fantasy**: 21.51%

Genres like **News** and **Film-Noir** show the highest propensity for spoilers, possibly due to the narrative complexities or the focus on critical events and mysteries that reviewers are inclined to discuss in detail.

3.7.3 Spoiler Proportion by Rating

The correlation between ratings and spoiler proportions reveals interesting trends on figure 3.11.

There are more spoilers in reviews with lower scores, especially those between 2.0 and 5.0. According to this trend, people who give content low ratings might be more likely to discuss plot specifics or critique particular parts of it, which could result in a higher rate of spoilers.

3.7.4 Spoiler Proportion by Review Length

The analysis of spoiler proportions in relation to review length shows the following:

- **Spoiler Proportion by Review Length:**
  - **Mean Spoiler Proportion**: 44.34%

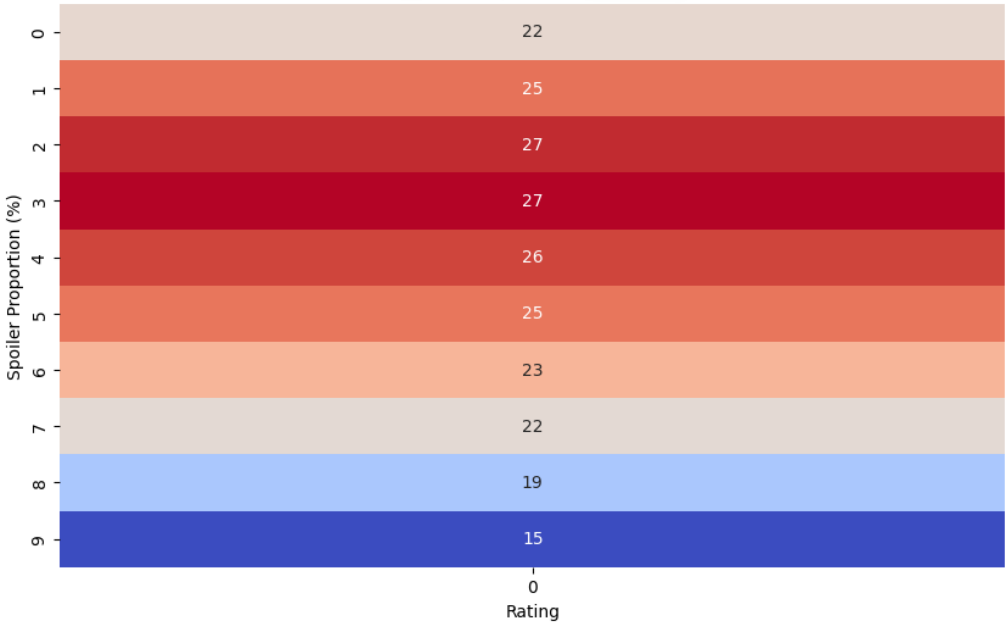

Figure 3.11: Spoiler Proportion by Rating

  - **Minimum**: 0%
  - **Maximum**: 100%

As demonstrated by the mean spoiler proportion of 44.34%, reviews that are longer in duration are more likely to contain disclosures. This research implies that reviewers are more inclined to delve into plot details in their reviews, which frequently leads to spoilers, since they write more in-depth and thorough assessments.

3.7.5 Spoiler Trends Over Time

The temporal analysis of spoiler trends indicates a significant increase in the proportion of spoilers over the years, particularly during the mid-2000s:

- **Spoiler Trends Over Time:**
  - **1998**: 0.81% spoilers
  - **2005**: 24.72% spoilers
  - **2009**: 34.03% spoilers
  - **2017**: 30.93% spoilers
  - **2020**: 12.94% spoilers
  - **2021**: 13.70% spoilers

The proportion of spoilers reached its maximum around 2009 and remained elevated until 2017, following which there was a discernible decrease. This pattern may be the result of shifting material types being examined, platform policies, or user behavior changes.

3.7.6 Spoiler Proportion by Reviewer

Certain reviewers are more prone to including spoilers in their reviews:

- **Top Spoiler-Prone Reviewers:**
  - **Reviewer: thetris**: 100% spoilers
  - **Reviewer: jdsantos-95954**: 100% spoilers
  - **Reviewer: jdseago**: 100% spoilers
  - **Reviewer: jdschreiber-21034**: 100% spoilers
  - **Reviewer: chethankeerthidg**: 100% spoilers
  - **Reviewer: WonderBlondie**: 100% spoilers
  - **Reviewer: jdscherf**: 100% spoilers
  - **Reviewer: nataliamaekivi**: 100% spoilers
  - **Reviewer: tallinen-93319**: 100% spoilers
  - **Reviewer: jdsarahsl**: 100% spoilers

All of these reviewers reviews include spoilers, which suggests that they either really like talking about story details or a lack of concern for spoiler-sensitive readers.

3.7.7 Spoiler Proportion by Review Helpfulness

The relationship between spoilers and review helpfulness shows a slight difference:

- **Helpfulness of Spoiler vs. Non-Spoiler Reviews:**
  - **Non-Spoiler Reviews**: 7.24 average helpful votes
  - **Spoiler Reviews**: 7.56 average helpful votes

Interestingly, spoiler-tagged reviews tend to receive slightly higher helpfulness ratings compared to non-spoiler reviews. This may indicate that in-depth reviews, which often contain spoilers, are perceived as more informative or valuable by the reader.

3.8 Sentiment Analysis and Trends

In this section, I conduct a comprehensive examination of the sentiment conveyed in IMDb reviews, analyzing the ways in which sentiment varies across various ratings, spoiler tags, time periods, and specific movies. The analysis provides information about the attitudes of viewers toward material and how these perceptions have changed over time. These sentiment labels are derived from C.6.6. Refer to figure 3.12 for detailed outline of the corrections.

Figure 3.12: Sentiment Analysis (Distribution & Trends)

3.8.1 Sentiment by Rating

The sentiment distribution by rating shows a clear correlation between sentiment and the rating given by reviewers:

- **Negative Sentiment**: Decreases as ratings increase. For instance, **94.17%** of 1-star ratings are associated with negative sentiment, which gradually decreases to **14.73%** for 10-star ratings.
- **Positive Sentiment**: Increases as ratings increase. Only **4.80%** of 1-star ratings are positive, but this rises sharply to **84.24%** for 10-star ratings.
- **Neutral Sentiment**: Remains relatively constant across ratings, though slightly higher in the middle range of ratings (e.g., 6-star ratings).

This trend is anticipated, as higher ratings generally indicate a favorable evaluation of the content, whereas lower ratings are more indicative of dissatisfaction or negative experiences. Because most reviews tend to offer a distinct perspective rather than a neutral attitude, neutral feeling is even less common.

3.8.2 Sentiment by Spoiler Tag

The sentiment distribution by spoiler tags reveals that:

- **Spoiler Reviews**: Have a higher proportion of negative sentiment (**55.15%**) compared to non-spoiler reviews (**40.99%**). Positive sentiment is lower in spoiler reviews (**43.42%**) compared to non-spoiler reviews (**57.57%**).
- **Neutral Sentiment**: Remains nearly identical between spoiler and non-spoiler reviews.

The increased negative sentiment in spoiler reviews may indicate that reviewers are more critical of the content when they include disclosures. This might be the case due to the fact that spoilers frequently result from a close examination of the material, which may involve criticizing important plot points or disappointing endings.

3.8.3 Sentiment Trends Over Time

The sentiment trends over time show some significant shifts:

- **1998 to Early 2000s**: Positive sentiment was initially high (e.g., **66.50%** in 1998) but gradually declined, reaching a lower point around 2009 (**49.50%**).
- **Mid 2000s to 2010s**: The period saw a steady increase in negative sentiment, peaking around 2008-2009, which coincides with significant cultural and economic changes globally.
- **2010s to 2021**: Positive sentiment began to stabilize and slightly recover, while negative sentiment saw a slight decline.

The periods between the mid-2000s and the early 2010s were characterized by intense criticism and even disillusionment; yet, as the industry and audience expectations changed, the trends may be an indication of more general cultural and socioeconomic changes.

3.8.4 Sentiment Trends for Specific Movies

The Shawshank Redemption (tt0111161)

- **1998 to Early 2000s**: Positive sentiment remains overwhelmingly high, exceeding **90%** in some years, particularly in the early years of IMDb when the movie was often cited as a fan favorite.
- **Mid 2000s to 2010s**: Positive sentiment fluctuates but remains strong, generally staying above **80%**. Negative sentiment slightly increases but never dominates.
- **2018 to 2021**: Positive sentiment dips slightly to around **80%**, with a corresponding minor increase in negative sentiment, potentially reflecting newer audiences revisiting the movie with different expectations.

The Shawshank Redemption maintains its status as a beloved classic, consistently generating high positive sentiment. The slight increase in negative sentiment over time could indicate a generational shift in perceptions or simply the law of large numbers as more people review the film.

The Dark Knight (tt0468569)

- **2008 to Early 2010s**: Sentiment is mixed in the early years, with **negative sentiment** peaking around **2009** (**44.35%**), likely due to initial hype followed by some critical backlash.
- **Mid 2010s to 2021**: Positive sentiment recovers significantly, with the movie achieving a strong positive reception, particularly in the years following its release, with **positive sentiment** exceeding **85%** in several years.
- **2018 to 2021**: There's a slight decrease in positive sentiment, but it remains above **79%**, indicating the film's lasting appeal.

Mixed reactions were initially observed toward *The Dark Knight*, which may have been attributed to its dark tone and the lofty expectations established by its predecessor. Nevertheless, it has maintained its status as a film that is both revered and critically acclaimed, with a prevailing positive sentiment in recent years.

The sentiment analysis offers valuable insights into the ways in which various factors, including time, ratings, and spoilers, affect the sentiment conveyed in reviews. The correlation between positive sentiment and higher ratings is robust, as anticipated; however, spoiler-tagged reviews tend to be more negative. Throughout time, sentiment trends are indicative of broader cultural changes. Certain films, such as *The Shawshank Redemption* and *The Dark Knight*, exhibit stable or improving sentiment, underscoring their enduring allure and influence.

3.9 Correlation Between Sentiment, Rating, Review Length, and Spoiler Tags

It is essential to recognize the dynamics of sentiment, rating, review length, and spoiler tags in order to understand how users interact with content on platforms such as IMDb. This report investigates the relationships between these variables, with a particular emphasis on the extent to which they contribute to the probability of a review containing disclosures.

3.9.1 Correlation Analysis and Key Findings

1. Sentiment and Rating for Spoiler Reviews (Figure 3.13)

A moderate to strong positive relationship is indicated by the correlation coefficient of **0.5298** between sentiment and rating in spoiler evaluations. The sentiment becomes more optimistic as the rating rises. This implies that the rating provided by evaluators is consistent with their overall sentiment, even when spoilers are included. Regardless of the spoiler content, positive sentiment and higher ratings frequently correlate.

2. Review Length and Spoiler Tags (Figure 3.13)

The correlation coefficient between review length and spoiler tags is **0.2693**, indicating a weak but noticeable positive relationship. This suggests that longer reviews are slightly more likely to contain spoilers. While the correlation is not strong, it implies that as reviews become more detailed, the likelihood of spoilers increases. However, review length alone is not a definitive predictor of spoilers.

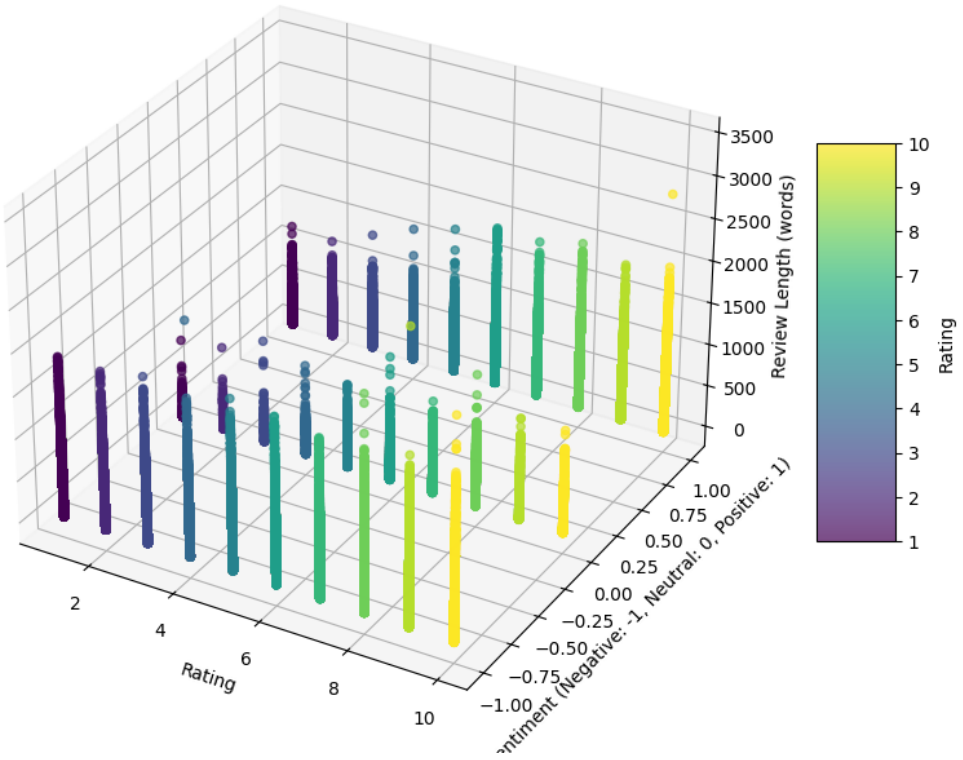

Figure 3.13: Sentiment vs. Rating vs. Review Length for Spoiler Reviews

3. Correlation Matrix for Spoiler Reviews (Figure 3.14):

Key findings from the correlation matrix:

- **Sentiment vs. Rating**: A moderate positive correlation (**0.5298**) suggests that higher ratings are associated with positive sentiment, even in spoiler reviews.
- **Sentiment vs. Review Length**: A weak negative correlation (**-0.0977**) indicates that longer reviews are slightly more likely to express neutral or negative sentiment.
- **Rating vs. Review Length**: A weak positive correlation (**0.0707**) suggests that higher-rated reviews tend to be slightly longer, but the relationship is minimal.

3.9.2 Implications

The analyses reveal that while there is a significant relationship between sentiment and rating, this does not strongly dictate whether a review will contain spoilers or how long the review will be. Positive sentiment and high ratings do not necessarily lead to non-spoiler reviews; these reviews may still contain spoilers but express overall favorable opinions.

Although the correlation between spoilers and review length is present, it is insufficient to serve as the solitary predictor. Spoilers are somewhat more likely to be included in longer assessments; however, this is not a definitive rule.

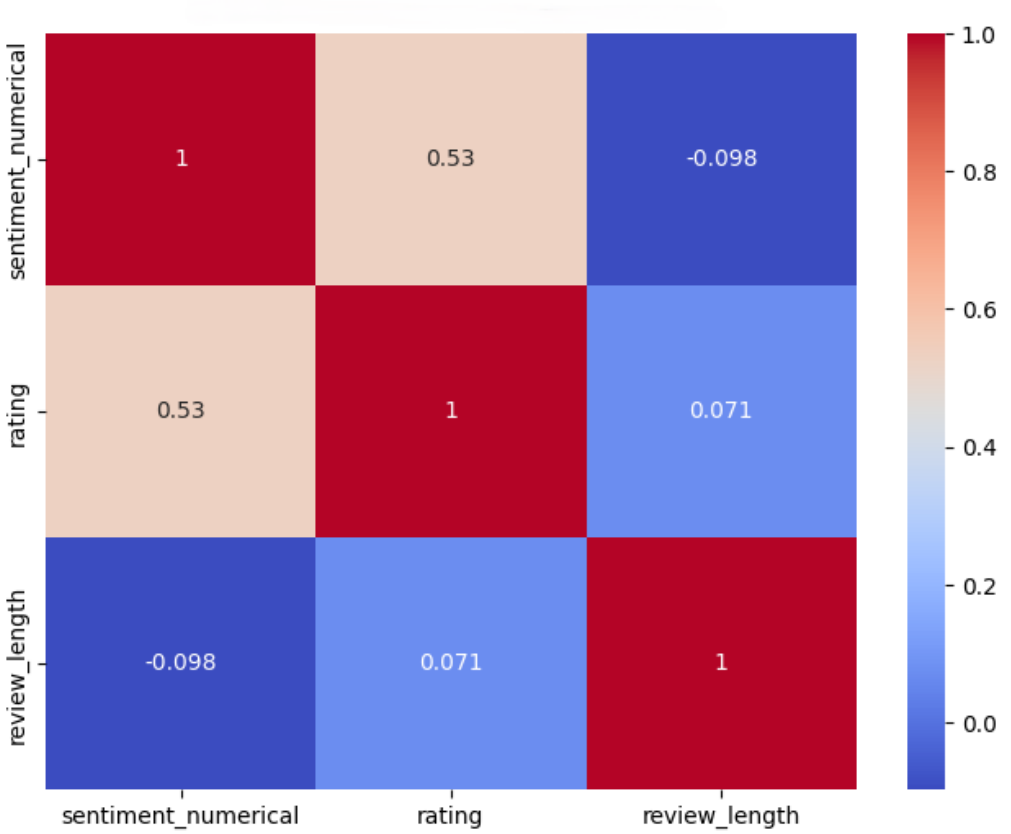

Figure 3.14: Correlation Matrix for Spoiler Reviews

3.10 Data Cleaning

Upon preliminary analysis of the raw dataset, I have identified roughly 6,996,247 distinct words when segmenting the text by whitespace. Notably, 6,211,101 of these distinct terms have punctuation marks. these discovery indicates that a significant proportion of the words may have noise in the form of punctuation, which could adversely affect the performance of any NLP model trained on these data. I have employed a character-wise cleaning procedure to get a high-quality text dataset. Manually checking and cleaning each distinct word is impractical. The existence of 6,211,101 distinct words with punctuation signifies that extensive preprocessing is required to mitigate noise and enhance the dataset's quality.

3.10.1 Character Database: Preliminary Overview

It is essential to understand the composition of the text data at a granular level before proceeding with the data cleanup process. After extracting the distinct characters from the entire dataset, there are 1,935 unique characters. To enhance comprehension of character distribution (figure 3.15). The visualization focuses on the most 50 frequent

characters to provide insights into the text's composition.

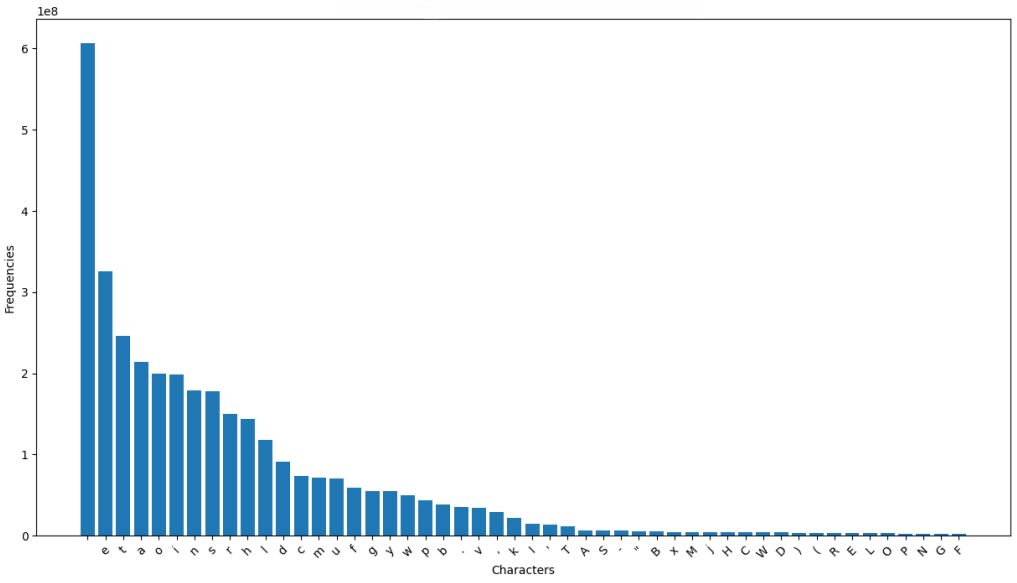

Figure 3.15: Top 50 Most Frequent Characters in Review Details

The graph indicates that the space character (‘ ‘) is the most prevalent character, which is typical for text data indicating that the text primarily consists of words separated by spaces. The most common alphabetical characters include ‘e‘, ‘t‘, ‘a‘, ‘o‘, and ‘n‘, consistent with common English text distributions. Characters such as commas, periods, and apostrophes are also prevalent. Additionally, the presence of various special characters (e.g., ‘@‘, ‘#‘, ‘¡, etc.) and accented letters (e.g., ‘é‘, ‘ü‘, etc.) suggests that the text data may include user-generated content with diverse formatting styles and possibly multilingual elements. Nonetheless, the dataset includes emojis and symbols, indicating that the reviews may contain emotional expressions or visual enhancements typically found in informal text. Thus comes the character categorization.

3.10.2 Character Categorization

To get a proper understanding of the characters, I have categorized them. Since, category based inspection alongside their occurrences would help handpicking replacements of unusual characters. Characters were divided into groups based on the pseudo-algorithm 1.

The figure 3.16 offers a detailed analysis of the various character types included in the dataset. It emphasizes that English characters, including both uppercase and lowercase letters, constitute a substantial part of the text. A significant quantity of non-English characters indicates the existence of several languages or specialized alphabetic symbols in the evaluations. Special characters, including symbols and financial signs, are featured, along with a significant quantity of emojis that highlight the informal and emotive quality of the content. Punctuation marks are present, although in lesser quantities than letters and non-English characters. The presence of numeric characters is limited, suggesting that the reviews mostly emphasize textual material with minimal numerical elements. Finally, there exists a category termed ”Others,” which includes characteristics that do

Algorithm 1 Categorize Characters

Input: Set of characters $\mathcal{C}$
Output: Sets $\mathcal{E}$, $\mathcal{N}$, $\mathcal{S}$, $\mathcal{M}$, $\mathcal{P}$, $\mathcal{D}$, $\mathcal{O}$
for each character c in $\mathcal{C}$ **do**
    if c is an emoji **then** ▷ Check if character is an emoji
        Assign c to $\mathcal{M}$ ▷ Add to emoji set
    else if c is an English letter **then** ▷ Check if character is an English alphabet letter
        Assign c to $\mathcal{E}$ ▷ Add to English set
    else if c is a digit **then** ▷ Check if character is a numeric digit
        Assign c to $\mathcal{D}$ ▷ Add to numbers set
    else if c is punctuation **then** ▷ Check if character is a punctuation mark
        Assign c to $\mathcal{P}$ ▷ Add to punctuation set
    else if c is a special character **then** ▷ Check if character is a special symbol
        Assign c to $\mathcal{S}$ ▷ Add to special characters set
    else if c is a non-English letter **then** ▷ Check if character is a non-English alphabet letter
        Assign c to $\mathcal{N}$ ▷ Add to non-English set
    else ▷ For any other character
        Assign c to $\mathcal{O}$ ▷ Add to others set
    end if
end for
Return $\mathcal{E}$, $\mathcal{N}$, $\mathcal{S}$, $\mathcal{M}$, $\mathcal{P}$, $\mathcal{D}$, $\mathcal{O}$

not conform to any of the established classifications, presumably due to their distinctive or infrequent presence in the dataset.

3.10.3 Character Replacement Pipeline

The following pseudocode 2 summarizes the text cleaning pipeline implemented to process and standardize the movie reviews. This pipeline systematically applies a series of replacement functions to ensure that the text is clean, consistent, and ready for analysis.

Algorithm 2 Text Cleaning Pipeline for Movie Reviews

Input: List of raw reviews $\mathcal{R}$
Output: List of partially cleaned reviews $\mathcal{C}$
Initialize $\mathcal{C} \leftarrow [\,]$ ▷ Initialize empty list for cleaned reviews
for each review r in $\mathcal{R}$ **do**
    $r \leftarrow$ ReplaceCharsNonEnglish(r, replacements_non_english)
    $r \leftarrow$ ReplaceSpecialChars(r, replacements_special_chars)
    $r \leftarrow$ ReplaceNumbers(r, number_mapping)
    $r \leftarrow$ ReplaceCommonChars(r, common_chars_mapping)
    $r \leftarrow$ ReplaceWeirdCharacter(r, weird_characters)
    Append r to $\mathcal{C}$
end for
Return $\mathcal{C}$

The sequence of operations in the text cleaning pipeline is meticulously designed to ensure that each replacement step is executed without disrupting subsequent processes. The first

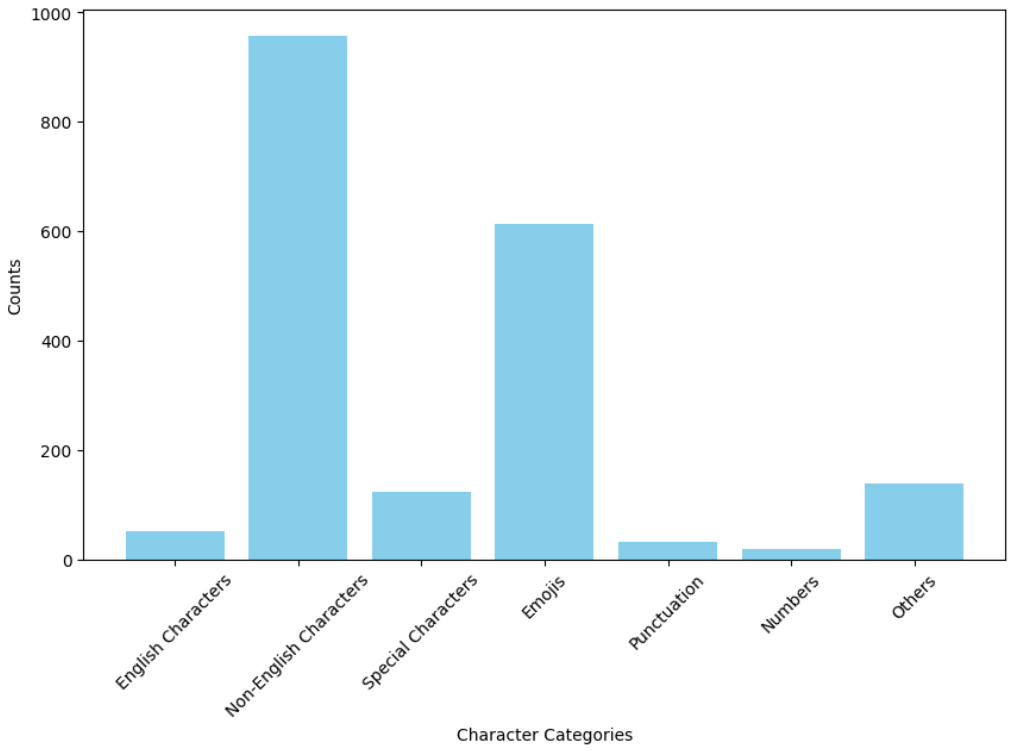

Figure 3.16: Counts of Character Categories in Dataset

step, *Non-English Character Replacement* (`ReplaceCharsNonEnglish`), is prioritized to standardize the text by converting foreign characters to their closest English equivalents or removing them, thus ensuring a uniform language format from the outset. Subsequently, *Special Character Replacement* (`ReplaceSpecialChars`) is performed to manage symbols like currency signs and mathematical operators, particularly addressing any special characters embedded within or alongside non-English text. Next, *Number Replacement* (`ReplaceNumbers`) is conducted to process numbers in a consistent and meaningful way, especially when they appear adjacent to special characters. The fourth step, *Common Character Replacement* (`ReplaceCommonChars`), emphasizes the standardization of fundamental textual components, such as punctuation marks and spaces, after the earlier stages have addressed non-English characters, special characters, and numbers. Finally, *Weird Character Removal* (`ReplaceWeirdCharacter`) is executed as a concluding step to eliminate any remaining noise or anomalies, ensuring that the text is left in its cleanest and most standardized form. This carefully serialized order ensures that each operation builds upon the previous one, resulting in a thoroughly cleaned dataset. The following sub-sections will provide more useful understanding on what how characters were handled for replacements –

Preservation of English Characters

The dataset contains the following set of English characters:

$$\{A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U, V, W, X, Y, Z, \}$$
$$\{a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z\}$$

These characters remained in their original state without any alteration or substitution. This decision is based on the fact that these characters belong to the standard ASCII set, which is universally acknowledged and utilized in English text processing. Preserving these characters guarantees that the reviews maintain their original linguistic structure and significance.

Non-English Characters

Non-English characters were identified as those that do not belong to the standard English alphabet. These characters could arise from various languages or special symbols used in the text data. The handling of non-English characters was carried out in two steps: replacement and removal. A subset of non-English characters with more than 300 occurrences in the dataset was replaced with their nearest English equivalents. The replacement was guided by the following mapping:

The table 3.5 clearly outlines how each frequently occurring non-English character was substituted with an English equivalent. The replacements were chosen to maintain both the phonetic and visual similarity to the original characters, ensuring minimal disruption to the text's readability and meaning. Reasoning behind the replacements are -

- **Phonetic Similarity**: The chosen replacements map non-English characters to English letters that are phonetically similar, ensuring that the readability and meaning of the text are preserved. For example, `é` is replaced with `e`, as they sound similar in most contexts.
- **Visual Similarity**: Some characters, like `ä` and `a`, are visually similar, making the transition less noticeable and preserving the aesthetic structure of the words.
- **Common Usage**: The replacements were prioritized for characters with significant occurrences (more than 300), as these have a noticeable impact on the dataset. By addressing these frequent characters, the integrity of the text data is maintained while reducing noise.

The non-English characters that did not meet the threshold of 300 occurrences were replaced with a whitespace character (`' '`). This decision was made to eliminate infrequent and potentially noisy characters that do not contribute significantly to the overall analysis. Reasoning behind the removals are -

- **Data Uniformity**: Replacing low-frequency characters with whitespace ensures uniformity across the dataset, reducing the risk of skewed results in subsequent text processing tasks.
- **Noise Reduction**: Characters with very low occurrences are likely to be errors, typos, or irrelevant symbols. Removing them prevents these outliers from affecting the quality of the analysis.

Algorithm 3 shows an overview of the non-english character replacements process -

Non-English Character	Replacement Character
â	a
ä	a
ó	o
û	u
Å	A
ô	o
à	a
ê	e
½	1/2
ë	e
ç	c
è	e
Á	A
á	a
ö	o
æ	ae
í	i
ü	u
é	e
ñ	n
ı	i
ï	i
ã	a
ú	u
ø	o
å	a
î	i
É	E

Table 3.5: Mapping of Non-English Characters to English Equivalents

Special Characters and Their Treatment

Special characters are symbols that are generally absent from conventional text data but may emerge in user-generated content, particularly in online reviews where individuals may express sentiments, ideas, or visual components through non-alphanumeric symbols. The treatment of these characters is essential for preserving the text's integrity and readability, while ensuring that the processed data is appropriate for analysis.

The replacements for special characters were guided by several key principles aimed at preserving the semantic meaning and ensuring clarity in the text.

Firstly, semantic preservation was prioritized. Characters such as '¢', '£', '€', and '*rupee*', which represent currency symbols, were replaced with their corresponding names ("cent", "pound", "euro", "rupee") to ensure the financial context is preserved in the text. Similarly, mathematical symbols like '×', '÷', and '√' were replaced with their English equivalents ("multiplied by", "divided by", and "square root of"), thereby maintaining the mathematical context, which could be crucial in reviews discussing numerical data or mathematical concepts.

Algorithm 3 Handle Non-English Characters

Input: Set of characters $\mathcal{C}$, Dictionary of replacements $\mathcal{R}$, Threshold $T = 300$
Output: Modified set of characters $\mathcal{C}'$
for each character c in $\mathcal{C}$ **do**
    if c is in $\mathcal{R}$ **then** ▷ Check if character has a replacement
        Replace c with $\mathcal{R}[c]$
    else if Frequency of $c > T$ **then** ▷ Check if character frequency is significant
        Keep c as it is
    else
        Replace c with whitespace ▷ Replace low-frequency characters
    end if
end for
Return Modified character set $\mathcal{C}'$

Secondly, readability enhancement was addressed by replacing special characters that convey direction or movement, such as '↑', '→', and '↓', with descriptive text ("up arrow", "right arrow", and "down arrow"). This ensures the directional meaning is retained while enhancing the text's readability for further processing. Symbols like 'check marks' and 'flower marks' were replaced with their descriptive equivalents ("check" and "flower") to retain the symbolic meaning without relying on visual cues that might not be captured in text analysis.

In addition, visual symbols were converted to descriptive text to ensure that the sentiment or rating expressed by the user is retained in a form that can be analyzed by text-processing algorithms. For example, stars and music notes, often used in reviews to express ratings or emotions, were replaced with words like "star" and "music note". Similarly, some characters were replaced with their descriptive equivalents to capture the emotional tone conveyed by these symbols.

Moreover, non-informative symbols which do not add semantic value to the text, were either replaced with whitespace or removed entirely to avoid unnecessary clutter in the dataset. This step helps in reducing noise in the text, making it cleaner for analysis. Flags codes were also replaced with whitespace, as they often represent country codes or emojis that do not contribute meaningfully to the textual content in a review context.

Finally, handling ambiguities was crucial. Some characters often appeared as a placeholder for an unknown or unrecognized character, was replaced with empty spaces. This replacement acknowledges the uncertainty or potential error in the text without introducing ambiguity into the analysis.

Table 3.6 summarizes the special character replacements, providing the reasoning behind each decision to ensure that the text remains semantically meaningful and suitable for analysis.

Other ASCII Character Replacement

In the context of text processing, ASCII characters refer to a variety of non-standard, control, and special characters that can disrupt analysis. These characters, often non-printable or used for formatting, are typically unnecessary in the text analysis pipeline and thus require careful handling. The categories of ASCII characters considered in this report include control characters, Latin-1 supplement control characters, whitespace and formatting characters, private use area characters, and special characters.

Special Character	Replacement Text	Reasoning
¢	cent	Preserves financial context
£	pound	Preserves financial context
€	euro	Preserves financial context
¥	yen	Preserves financial context
√	square root of	Preserves mathematical context
×	multiplied by	Preserves mathematical context
÷	divided by	Preserves mathematical context
↑	up arrow	Enhances readability
→	right arrow	Enhances readability
check	check	Converts visual symbol to descriptive text
heart	heart	Converts visual symbol to descriptive text
star	star	Captures rating or emphasis
♪	music note	Converts musical symbols to descriptive text
non-informative symbols	(Whitespace)	Removes non-informative symbols
?	question_mark	Acknowledges text ambiguity
dashed symbols	-	Simplifies visual line characters
summation of	summation of	Preserves mathematical context
∞	infinity	Preserves mathematical context
!=	not equal	Preserves mathematical context
intersection	intersection	Preserves mathematical context
>=	greater than or equal slant	Preserves mathematical context
sun_with_face	sun_with_face	Captures symbolic visual context
~~£~~, , №, , ~~₦~~,	currency/number/measurement name	Preserves financial context
A to *Z*	(Whitespace)	Removes country flags with no textual contribution

Table 3.6: Summary of Special Character Replacements

Control Characters (U+0000 to U+001F, U+007F) **Description:** These are non-printable characters used for text control in data streams.
Action: Remove
Reason: They do not contribute to the content and can disrupt text processing.
Examples:

- `\x00` (U+0000): Null character
- `\x03` (U+0003): End of Text
- `\x08` (U+0008): Backspace
- `\x1b` (U+001B): Escape
- `\x7f` (U+007F): Delete

Latin-1 Supplement Control Characters (U+0080 to U+009F) **Description:** These are additional control characters in the Latin-1 Supplement block.

Action: Remove
Reason: Similar to the basic control characters, these are generally non-printable and can cause issues.
Examples:

- `\x80` (U+0080) to `\x9f` (U+009F): Various control characters

Whitespace and Formatting Characters Non-Breaking Space (U+00A0)
Action: Replace with a regular space
Reason: Ensures consistency in word separation.
Example: `\xa0` (U+00A0)

Soft Hyphen (U+00AD)
Action: Remove
Reason: It does not contribute to textual meaning and can disrupt word boundaries.
Example: `\xad` (U+00AD)

Zero Width Characters (U+200B to U+200D)
Action: Remove
Reason: These control formatting and are usually unnecessary for content analysis.
Examples:

- `\u200b` (U+200B): Zero Width Space
- `\u200c` (U+200C): Zero Width Non-Joiner
- `\u200d` (U+200D): Zero Width Joiner

Directional Formatting Characters (U+200E to U+200F, U+202A to U+202E)
Action: Remove
Reason: These control text direction and are not needed unless handling bidirectional text.
Examples:

- `\u200e` (U+200E): Left-to-Right Mark
- `\u200f` (U+200F): Right-to-Left Mark
- `\u202a` (U+202A): Left-to-Right Embedding

Invisible Formatting Characters (U+2062 to U+206F)
Action: Remove
Reason: These characters are for special text processing and are typically not necessary.
Examples:

- `\u2062` (U+2062): Invisible Times
- `\u206f` (U+206F): Nominal Digit Shapes

Ideographic Space (U+3000)
Action: Replace with a regular space
Reason: Ensures consistency in spacing.
Example: `\u3000` (U+3000)

Zero Width No-Break Space (U+FEFF)
Action: Remove
Reason: Often used as a Byte Order Mark, which is not necessary in most text processing contexts.
Example: `\ufeff` (U+FEFF)

Private Use Area Characters (U+E000 to U+F8FF) **Description:** These are reserved for private use and have no standard meaning.
Action: Remove
Reason: These characters are custom and not standardized, likely unnecessary for general text analysis.
Examples:

- `\ue000` (U+E000) to `\uf8ff` (U+F8FF)

Special Characters Arabic End of Ayah (U+06DD)
Action: Remove
Reason: Unless processing specific religious or Arabic texts, this character is usually not needed.
Example: `\u06dd` (U+06DD)

Supplementary Private Use Area (U+10000 to U+10FFFF)
Action: Remove
Reason: These are for private use in higher planes and generally not needed.
Example: `\U000fe347` (in the Supplementary Private Use Area)

Summary of Other ASCII Character Replacement Actions

The table 3.7 provides an overview of the whole replacement pipeline.

Replacement of Other Common Characters

In addition to handling standard punctuation, numbers, and special symbols, the dataset also includes a variety of other characters that require normalization to ensure consistency and readability. The following replacements were carefully chosen to maintain the semantic integrity of the text while simplifying it for further processing.

Reasoning Behind Other Character Replacements

1. **Whitespace and Formatting Characters**:
    - **Tab (`\t`)** and **Newline (`\n`)**: These characters were retained to preserve the original formatting of the text. Tabs and newlines are crucial for maintaining the structure of paragraphs and lists within reviews.
    - **Space (`' '`)**: Spaces were retained as they are essential for separating words and maintaining the readability of the text.
2. **Inverted Punctuation Marks**:

Unicode Range	Character Types	Action	Reason
U+0000-U+001F	Control Characters	Remove	Non-printable, disrupt processing
U+007F	Delete	Remove	Non-printable, disrupt processing
U+0080-U+009F	Latin-1 Supplement Control Characters	Remove	Non-printable, disrupt processing
U+00A0	Non-Breaking Space	Replace (space)	Ensure word separation consistency
U+00AD	Soft Hyphen	Remove	Disrupts word boundaries
U+06DD	Arabic End of Ayah	Remove	Unnecessary unless processing specific text
U+200B-U+200D	Zero Width Characters	Remove	Formatting control, not needed
U+200E-U+202E	Directional Formatting Characters	Remove	Text direction control, not needed
U+2062-U+206F	Invisible Formatting Characters	Remove	Special text processing, not needed
U+3000	Ideographic Space	Replace (space)	Ensure spacing consistency
U+E000-U+F8FF	Private Use Area Characters	Remove	Custom, non-standardized
U+FEFF	Zero Width No-Break Space	Remove	Byte Order Mark, not needed
U+10000+	Supplementary Private Use Area Characters	Remove	Custom, non-standardized

Table 3.7: Summary of ASCII Character Replacements

- **Inverted Exclamation Mark (¡)**: Replaced with "exclamation" to maintain the emphasis conveyed by this punctuation in Spanish and other languages, while standardizing it to the regular exclamation mark used in English.
- **Inverted Question Mark (¿)**: Replaced with "?" to align it with the standard question mark used in English, maintaining the interrogative intent.

3. **Quotation Marks and Apostrophes**:

   - **Left and Right Single Quotation Marks**: Replaced with a standard apostrophe (') to maintain consistency across text where quotation marks are used.
   - **Left and Right Double Quotation Marks**: Replaced with a standard double quote (") to standardize text containing quotes.
   - **Single and Double Low-9 Quotation Marks (‚, „)**: Replaced with a comma (,) or double quote ("), respectively, to maintain consistency with common English punctuation.

4. **Mathematical and Technical Symbols**:

- **Mathematical Angle Brackets**: Replaced with < and > to retain their directional meaning in mathematical or technical contexts.
- **Hyphens and Dashes**: All variations were standardized to a simple hyphen (-) to maintain consistency in word separation and sentence structuring. This ensures uniformity in the representation of these often similar-looking characters.

5. **Ellipsis and Dots**:

   - **Ellipsis (…)**: Replaced with a period (.) to standardize the representation of pauses or unfinished thoughts in the text. This makes it easier to process and analyze sentence boundaries.

6. **Asian and Arabic Punctuation**:

   - **Japanese and Fullwidth Characters**: These characters were replaced with their standard ASCII equivalents (,, ., <, >, !, ", (,), etc.) to align with the English punctuation system. This replacement is crucial for maintaining consistency in reviews that mix languages.
   - **Arabic Comma, Semicolon, and Question Mark**: Replaced with their English equivalents (,, ;, ?) to maintain uniformity and facilitate easier text processing in mixed-language content.

7. **Miscellaneous Symbols**:

   - **Section Sign (§)**, **Pilcrow (¶)**, **Reference Mark (※)**, **Undertie (‿)**, **Asterism**: Replaced with whitespace as these characters generally serve formatting or typographical roles, which are not necessary for the analysis of the text's semantic content.
   - **Middle Dot (·)**: Replaced with a hyphen (-) to retain word separation while standardizing the text.
   - **Katakana Middle Dot**: Similarly replaced with a period (.) for consistency in punctuation.

Table of Other Character Replacements

Handling of Uncommon and Non-Printable Characters

In the process of text cleaning, certain uncommon and non-printable characters need to be addressed to ensure that the text is clean, readable, and ready for analysis. These characters, often referred to as "unconventional characters," can be the result of encoding issues, copy-paste errors, or other anomalies that introduce non-standard symbols into the text. This section details the approach taken to handle such characters in the dataset.

Identified unconventional Characters The characters on table 3.9 were identified as unconventional or non-standard in the dataset.

These characters include a mixture of non-printable ASCII control characters, zero-width spaces, non-breaking spaces, and other special symbols from different Unicode blocks.

Character	Replacement Text	Reasoning
\t	(Tab)	Retains original formatting
\n	(Newline)	Preserves paragraph and list structure
' '	(Space)	Maintains word separation
¡	exclamation	Preserves emphasis in Spanish or other languages
§	(Whitespace)	Reduces typographical noise
«	<	Standardizes quotation marks
¶	(Whitespace)	Removes unnecessary formatting marks
·	-	Standardizes word separation
»	>	Standardizes quotation marks
¿	?	Standardizes question marks
Low-9	"	Aligns with English punctuation
Arabic	,, ;, ?	Aligns Arabic punctuation with English
‒, –, —, ―	-	Standardizes hyphens and dashes
‘, ’	'	Standardizes quotation marks
", "	"	Standardizes quotation marks
‚, „	,, "	Aligns with English punctuation
•, …, ⁕	(Whitespace)	Reduces noise and unnecessary typographical elements
<, >	less than, greater than	Retains technical/mathematical meanings
Others	Standard ASCII equivalents	Aligns with English punctuation for consistency

Table 3.8: Summary of Other Character Replacements

Approach to Handling unconventional Characters

1. **Removal of Non-Printable and Control Characters**:
   - **Control Characters (\x00 to \x1f, \x7f to \x9f)**: These characters are non-printable and primarily used for control purposes in text processing. They were removed because they do not contribute any meaningful content to the text and can interfere with text processing algorithms.
   - **Special Symbols and Private Use Area Characters (\ue000 to \ufeff)**: These characters belong to the Unicode Private Use Area (PUA) and other special symbols that may have been incorrectly rendered or included due to encoding errors. Since these symbols do not have a standard interpretation and are unlikely to contribute meaningful information, they were removed.
2. **Handling of Non-Breaking Spaces and Zero-Width Characters**:
   - **Non-Breaking Space (\xa0)**: This character was replaced with a regular space (' '). The non-breaking space is commonly used in web pages and documents to prevent line breaks between words or elements, but in plain text, it can be safely replaced with a regular space to maintain readability.
   - **Zero-Width Characters (\u200b to \u200f, \u202a to \u202d, \u2062 to \u206f)**: These characters are invisible in the text and are typically used for

Characters
\x00, \x03, \x08, \x0f, \x10, \x11, \x15, \x18, \x19, \x1a, \x1b, \x1e, \x1f, \x7f, \x80, \x81, \x82, \x83, \x84, \x85, \x86, \x87, \x88, \x89, \x8a, \x8b, \x8c, \x8d, \x8e, \x8f, \x90, \x91, \x92, \x93, \x94, \x95, \x96, \x97, \x98, \x99, \x9a, \x9b, \x9c, \x9d, \x9e, \x9f, \xa0, \xad, \u06dd, \u200b, \u200c, \u200d, \u200e, \u200f, \u202a, \u202c, \u202d, \u2062, \u206e, \u206f, \u3000, \ue000, \ue40d, \uf02d, \uf042, \uf04a, \uf04c, \uf08c, \uf08d, \uf08e, \uf08f, \uf090, \uf091, \uf092, \uf095, \uf0a7, \uf0ab, \uf0b7, \uf0bb, \uf0d8, \uf0e0, \uf0e8, \ufeff, \U000fe347

Table 3.9: Unconventional Characters

formatting in certain languages. They were removed as they do not contribute to the text's content and can introduce inconsistencies in text analysis.

3. **Other Special Symbols**:

   - **Tibetan Mark (`\u06dd`)**, **Enclosed Characters**, and **Miscellaneous Symbols**: These characters were removed as they are unlikely to contribute meaningful content to movie reviews. They may have been introduced due to encoding issues or user input errors.

Final Action: Removal and Replacement Summary

The table 3.10 provides an overview of the whole procedure.

Character Type	Action	Reasoning
Control Characters (`\x00` to `\x1f`, `\x7f` to `\x9f`)	Removed	Non-printable, interferes with text processing.
Special Symbols (Private Use Area, Enclosed Characters, etc.)	Removed	Non-standard, likely due to encoding issues, no meaningful contribution.
Non-Breaking Space (`\xa0`)	Replaced with `' '`	Standardizes spacing, improves readability.
Zero-Width Characters	Removed	Invisible characters, removed to maintain text consistency.
Other Miscellaneous Symbols	Removed	Unlikely to contribute meaning, often due to encoding issues.

Table 3.10: Summary of Unconventional Characters Replacements

Replacement of Punctuation Characters

Punctuation marks are integral to the structure and meaning of sentences. In the context of movie reviews, they can significantly impact the tone, emphasis, and clarity of the content. However, for tasks like spoiler detection, it's important to standardize these punctuation marks to maintain consistency while retaining their semantic impact.

Reasoning Behind Punctuation Replacements

1. **Semantic Clarity**:
    - **Exclamation Mark (`!`)**: Replaced with "exclamation" to retain the emphasis or strong emotion conveyed in the text. This replacement ensures that the intensity of the reviewer's sentiment is preserved.
    - **Question Mark (`?`)**: Replaced with "question" to highlight uncertainty or inquiry within the review. This can be crucial for understanding a reviewer's hesitation or doubt, which might be related to plot details.
    - **Comma (`,`) and Period (`.`)**: Removed to simplify text processing while maintaining sentence structure through whitespace. In some cases, periods can be replaced with spaces to ensure proper tokenization without losing sentence boundaries.
2. **Financial and Mathematical Symbols**:
    - **Dollar Sign (`$`)**: Replaced with "dollar" to clearly indicate monetary references. This is important for reviews discussing costs, budgets, or financial aspects of a movie.
    - **Percentage Sign (`%`)**: Replaced with "percentage" to preserve references to statistical or rating data, which are common in reviews.
    - **Plus Sign (`+`)**: Replaced with "plus of" and **Multiplication Sign (`*`)** with "multiplied by" to retain the mathematical context, which might be used in numerical comparisons or descriptions.
3. **Commonly Used Symbols**:
    - **Ampersand (`&`)**: Replaced with "and" to maintain conjunctions in text, ensuring the logical flow of sentences.
    - **At Symbol (`@`)**: Replaced with "at" to maintain references, especially in social media handles or email addresses mentioned in reviews.
4. **Neutral or Non-Informative Symbols**:
    - Symbols like quotation marks (`"`) and brackets (`(`, `)`, `[`, `]`) were removed to reduce noise. These characters often serve structural roles that are less relevant to the semantic content of the text.
    - Hyphens (`-`) and underscores (`_`) were left unchanged or removed, depending on their usage, to preserve word compounds or formatting consistency.
5. **Directional and Relational Symbols**:
    - **Less Than (`<`), Greater Than (`>`), and Equal Sign (`=`)**: These symbols were replaced with "less than," "greater than," and "equal" to retain comparative expressions in the text.

Table of Punctuation Replacements

Punctuation Character	Replacement Text	Reasoning
!	exclamation	Preserves emphasis or strong emotion
?	question	Highlights uncertainty or inquiry
$	dollar	Indicates monetary references
%	percentage	Retains statistical or rating context
&	and	Maintains logical conjunction in text
+	plus of	Retains mathematical context
*	multiplied by	Retains mathematical context
@	at	Preserves references, especially in social media handles
.	(Whitespace)	Simplifies text processing while retaining sentence structure
,	(Whitespace)	Simplifies text processing while maintaining list structure
<	less than	Preserves comparative expressions
>	greater than	Preserves comparative expressions
=	equal	Retains relational meaning
"	(Whitespace)	Reduces noise
()	(Whitespace)	Reduces noise
[]	(Whitespace)	Reduces noise
-	-	Preserves word compounds or separation

Table 3.11: Summary of Punctuation Replacements

Replacement of Numbers

Numbers play a vital role in movie reviews, often used to describe ratings, release years, box office statistics, or other quantitative information. It's important to standardize number characters to ensure that all numerical data is processed consistently.

Reasoning Behind Number Replacements

1. **Consistency in Numerical Representation**:
    - The standard digits (0-9) were retained as they are universally recognized and essential for numerical expressions in the text.
    - Superscript numbers such as ¹, ², and ³ were converted to their standard digit equivalents (1, 2, 3) to maintain consistency in numerical representation. This standardization prevents issues during data processing and analysis where superscripts might be misinterpreted or overlooked.
2. **Normalization of Non-Standard Digits**:
    - Some non-standard digits (full-width forms used in certain Asian languages) were converted to their standard digit counterparts (1, 2, 3, 5, 6). This

conversion ensures that the numerical data is uniform across the dataset, facilitating easier parsing and analysis.

Number Character	Replacement Text	Reasoning
0-9	0-9	Standard digits retained for accurate numerical representation
¹	1	Normalizes superscript numbers to standard form
²	2	Normalizes superscript numbers to standard form
³	3	Normalizes superscript numbers to standard form
1	1	Converts full-width digits to standard digits for consistency
2	2	Converts full-width digits to standard digits for consistency
3	3	Converts full-width digits to standard digits for consistency
5	5	Converts full-width digits to standard digits for consistency
6	6	Converts full-width digits to standard digits for consistency

Table 3.12: Summary of Number Replacements

Table of Number Replacements

Reasoning for Emoji Replacement

1. **Semantic Preservation**:

   Emojis can carry substantial semantic content. For instance, a *smiley face emoji* might indicate satisfaction or approval, while a *crying face emoji* could signify sadness or disappointment. In the context of movie reviews, these sentiments can reflect a viewer's reaction to specific plot developments, which are crucial for spoiler detection.

   Replacing emojis with descriptive text preserves the emotional tone and context that might be otherwise lost if emojis were simply removed. This ensures that the spoiler detection model can analyze the full sentiment conveyed by the reviewer.

2. **Text Uniformity**:

   By converting emojis to text, the review content becomes more uniform, which is beneficial for natural language processing tasks. This uniformity allows the text to be processed consistently, improving the accuracy and performance of machine learning models, including those designed for spoiler detection.

3. **Improved Sentiment Analysis**:

   Emojis can significantly affect the sentiment of a text. For example, a sentence ending with a *heart emoji* has a different tone than the same sentence ending with a neutral period. By replacing emojis with descriptive text, the sentiment analysis component of spoiler detection can better gauge the review's emotional content, leading to more accurate predictions.

4. **Capturing Implicit Spoilers**:

   Emojis can sometimes convey implicit spoilers. For example, the use of a *coffin emoji* in a review could hint at the death of a character, which is critical information for spoiler detection. Replacing such emojis with their text equivalents ensures that these subtle hints are not overlooked by the model.

Pseudocode for Emoji Replacement

Algorithm 4 Replace Emojis with Descriptive Text

```
Input: Text 𝒯
Output: Modified text 𝒯′ with emojis replaced by text
Define ℰ as a regex pattern to match all emojis
for each match e in 𝒯 do
    if e is an emoji then
        Replace e with its descriptive text equivalent from the emoji dictionary
    end if
end for
Return Modified text 𝒯′
```

Example Transformations

To further clarify the impact of emoji replacement, consider the following examples:

- **Original Review**: "I was so happy when the hero won! *grinning_face_with_smiling_eyes*"
- **After Emoji Replacement**: "I was so happy when the hero won! grinning_face_with_smiling_eyes"
- **Original Review**: "The ending was heartbreaking... *crying_emoji*"
- **After Emoji Replacement**: "The ending was heartbreaking... crying_face"

In both cases, the replacement ensures that the sentiment conveyed by the emojis is retained in text form, making it accessible for the spoiler detection algorithm.

3.10.4 Preprocessing Pipeline

Preprocessing text data is essential before starting machine learning tasks, such as sentiment analysis or detecting spoilers in movie reviews. Preprocessing ensures that the text is uniformly presented, free of unnecessary noise, and ready for analysis. Below is a comprehensive overview of the preprocessing pipeline used in this study, with a step-by-step explanation of each phase.

Preprocessing Pipeline

The preprocessing pipeline involves several key steps designed to clean, standardize, and prepare the text data:

1. **Expanding Contractions**: Initially, The text is passed through a function to expand common English contractions (for instance, "can't" is transformed into "cannot"). Expanding contractions ensures that the text is more formal and easier for models to understand. It also helps in reducing the complexity of language by converting informal contractions into their full forms, which can improve the accuracy of tokenization and subsequent text processing.

2. **Removing Extra Apostrophes**: Subsequent to the expansion of contractions, any remaining apostrophes (`'`) are removed. This step helps text purification by eliminating unnecessary characters that may affect tokenization.

3. **Removing Extra Whitespace**: To maintain consistent formatting, extra whitespace is removed, resulting in single spaces between words. This enhances the text's structure and mitigates potential difficulties during analysis caused by irregular spacing.

4. **Replacing Punctuation**: Punctuation marks are replaced according to a predefined mapping. Punctuation can convey important semantic information or simply add noise. Standardization enables the model to concentrate on critical information without being misled by punctuation discrepancies. By replacing them with standardized equivalents or descriptive text, the punctuation's role is clarified, making the text more suitable for processing by machine learning models.

5. **Replacing Emojis with Descriptive Text**: Emojis, which frequently carry significant emotional or contextual information, are transformed into descriptive text. This modification enables the model to interpret these symbols as vital to the broader sentiment or context without forfeiting essential information.

6. **Final Preprocessing (Combining All Steps)**: Each of the above processes is executed sequentially, ensuring that the text becomes more refined and organized with every stage. This sequential method ensures a meticulously prepared dataset, suitable for processing by machine learning models.

7. **Cased and Uncased Processing**: After preprocessing, the text is handled in two different formats:

    - **Cased Process**: The text is stored in its original case, which is crucial for tasks that may require sensitivity to capitalization, such as identifying named entity recognition or sentiment analysis, where uppercase letters might convey emphasis.
    - **Uncased Process**: The text is additionally transformed to lowercase after all other preprocessing steps are completed. Lowercasing the text is a common practice in NLP to reduce the vocabulary size and eliminate distinctions between capitalized and uncapitalized words that may not be semantically significant. This is particularly useful in tasks where case sensitivity does not add value.

Pseudocode for Text Preprocessing Pipeline

Here is a concise pseudocode representing the preprocessing pipeline:

Algorithm 5 Text Preprocessing for Model Input

Input: List of raw reviews $\mathcal{R}$
Output: Lists of cased and uncased processed reviews $\mathcal{C}$ and $\mathcal{U}$
Initialize $\mathcal{C} \leftarrow [\,], \mathcal{U} \leftarrow [\,]$ ▷ Initialize empty lists for processed reviews
for each review r in $\mathcal{R}$ **do**
    $r \leftarrow \textsc{ReplacePunctuation}(r, \text{punctuation\_mapping})$
    $r \leftarrow \textsc{ReplaceEmojisToText}(r)$
    $r \leftarrow \textsc{Preprocess\_}(r)$ ▷ Expand contractions, remove extra apostrophes and whitespace
    Append r to $\mathcal{C}$ ▷ Store cased version of the review
    $r \leftarrow \textsc{Lowercase}(r)$
    Append r to $\mathcal{U}$ ▷ Store uncased version of the review
end for
Return $\mathcal{C}, \mathcal{U}$

Serialization of Operations

The serialization of operations in this preprocessing pipeline is crucial for several reasons:

1. **Logical Flow of Text Transformation**:

   The pipeline begins with the most fundamental transformations—expanding contractions and removing extra apostrophes—before moving on to punctuation and emoji replacement. This order ensures that the text is standardized and clean before tackling more complex replacements, such as converting emojis to descriptive text.

2. **Prevention of Conflicts**:

   By expanding contractions first, the pipeline avoids potential conflicts where punctuation marks or apostrophes within contractions might be misinterpreted. Similarly, removing extra whitespace last ensures that any whitespace introduced during earlier steps (e.g., by replacing punctuation) is appropriately handled.

3. **Case Sensitivity Consideration**:

   The decision to maintain both cased and uncased versions of the text allows for flexibility in model input. Certain tasks may benefit from case sensitivity, while others might not, making this dual processing approach beneficial for various downstream applications.

4. **Maximizing Clarity and Consistency**:

   Each step is designed to build upon the previous one, progressively refining the text to ensure maximum clarity and consistency. This structured approach results in a dataset that is well-prepared for machine learning models, minimizing the risk of noise or errors in analysis.

Chapter 4
Methodology

This thesis focuses on the feature engineering side of NLP to enhance language model performance for spoiler detection and identification tasks. This study seeks to illustrate the continued significance and possibly enhanced efficacy of task-specific models by strategic feature engineering, despite the swift progress in large language models (LLMs). This investigation is essential for comprehending how various attributes can augment the fundamental functionalities of models such as BERT and RoBERTa for particular applications.

The primary objectives of the implementation were as follows:

1. **Establish a Baseline**: To set a benchmark for performance using a basic BERT-base-uncased model trained on the entire uncleaned dataset.
2. **Explore Feature Engineering**: To systematically test and evaluate the impact of various feature integrations such as review sentiment, review length, and summaries on model performance.
3. **Model Selection and Optimization**: To choose the best model and feature combination by systematic comparison with an emphasis on improving the precision and effectiveness of spoiler identification.
4. **Comprehensive Model Training**: To refine and finalize the model training on a large dataset which will ensure the robustness and scalability of the chosen solution.

A model's performance in certain tasks can be greatly improved by using specialized feature engineering. This chapter will demonstrate through the procedures followed from the initial model training to the end model's selection and optimization.

4.1 Experimental Setup

Due to limitations in computational resources, the experimental methodology was structured to evaluate multiple feature engineering strategies within a feasible framework. A

subset of data was selectively extracted from 20 movies with the most comprehensive contextual information available. This sample consisted of 33,752 data points, with 20,000 designated for training and 13,752 for testing.

4.1.1 Rationale for Model and Feature Choices

The selection of `roberta-base` as the principal model for some experiments was primarily motivated by its comparatively smaller size and efficiency in comparison to larger models such as `bert-base`. This choice was made since it is more suitable for situations that have limited computational power. Furthermore, it has been demonstrated that RoBERTa offers superior performance on tasks that need contextual comprehension. This is a result of its improved training technique on longer sequences, which is advantageous for comprehending intricate movie storylines and reviews.

Feature selection was guided by the hypothesis that integrating various textual characteristics could significantly influence model performance. The features chosen for experimentation were:

- **Review Text Alone**: To establish a baseline performance metric.
- **Review & Review Sentiment**: To assess the impact of sentiment analysis on spoiler detection.
- **Review & Review Length with Review Sentiment**: To determine if additing review length and sentiment change model performance.
- **Review & Review Summary with Review Sentiment**: To test the combined effect of multiple features, hypothesizing that a richer feature set could enhance detection performance.

4.2 Diagram of Experimental Workflow

Below is a detailed diagram illustrating the experimental setup and flow for feature engineering and model selection:

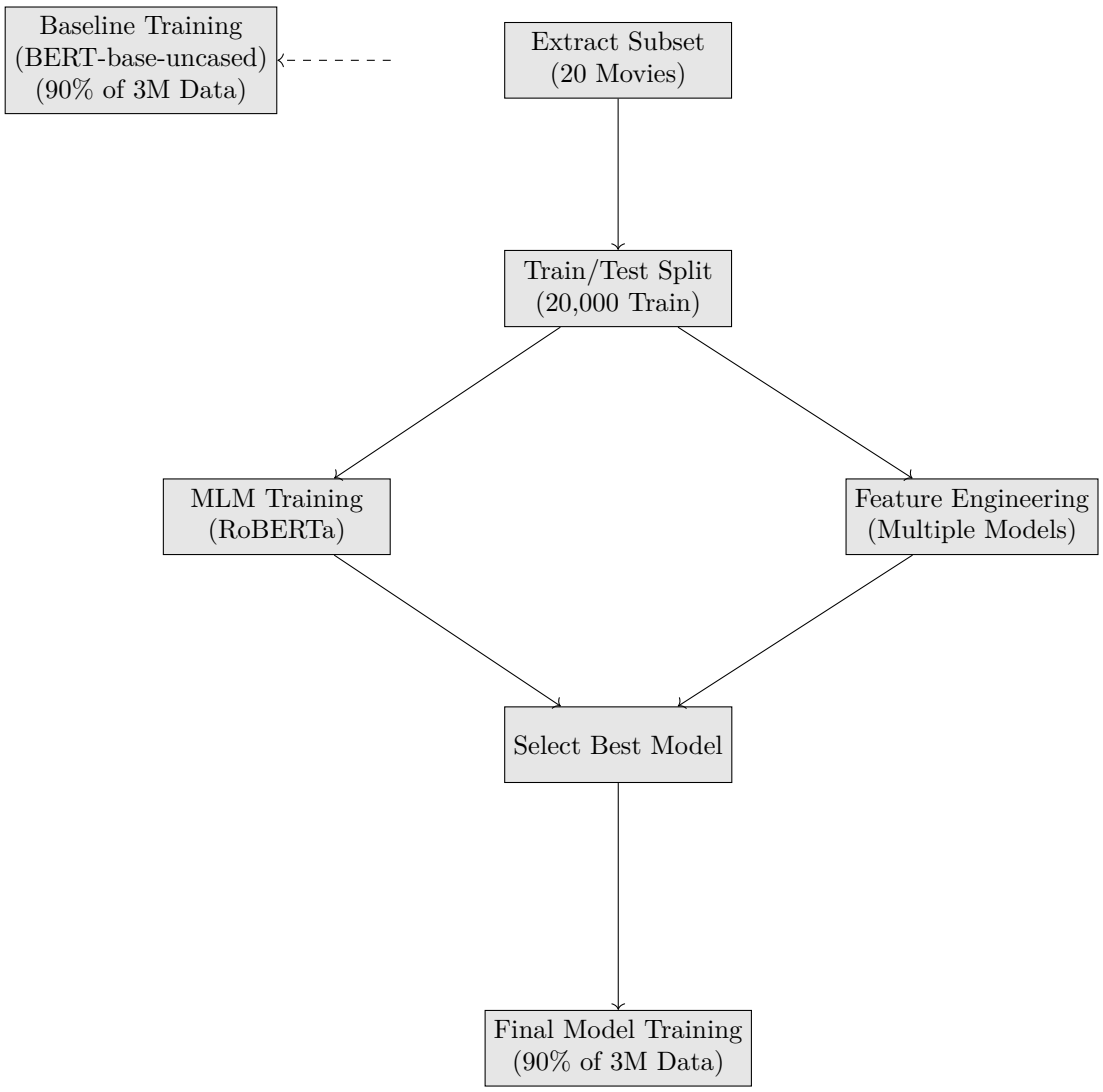

Figure 4.1: Detailed Experimental Workflow

Workflow Description When it comes to the experimental design of this thesis, figure 4.1 provides an illustration of the systematic method that was employed. An initial step involves the extraction of a specific subset of data, with the goal of focusing on twenty films that are well-known for the considerable contextual content they provide. This subset includes full movie synopses and summaries. For the purpose of facilitating robust model evaluation, this subset is then divided into two sets: a training set, which contains 20,000 of the data, and a testing set, which contains the rest of the data points from the extracted subset.

After the data has been prepared, the experimental process is divided into two unique pathways: the first pathway is for training the Masked Language Model (MLM) with the RoBERTa model, and the second pathway is for substantial feature engineering that involves numerous model architectures. Both the MLM approach and the feature

engineering path investigate various aspects of feature utilization and model performance. The MLM approach focuses on contextual embedding optimization, while the feature engineering path tests various combinations of textual features such as sentiment, length, and summarization in conjunction with standard model inputs.

Following this, the results of these parallel routes are subjected to a critical evaluation, and the model that had the highest weighted F1 score on the testing set is chosen as the model with the best performance. After that, this model is trained on a substantially bigger dataset, making use of 90% of the 3M data points that are accessible. This ensures that the final model is both robust and scalable. This extensive training regimen is accompanied by a baseline model training process. This process utilizes the `bert-base-uncased` model, which is trained on the same dataset as the selected model without any extensive preprocessing.

4.3 Baseline Model Setup

In this particular implementation, the baseline model that has been selected is the `bert-base-uncased` model. This particular configuration of the BERT model, which was developed by Google, is among the most often utilized configurations. This model has been pre-trained on a substantial body of English data, and it has garnered widespread recognition for its efficiency in handling tasks involving natural language comprehension. The selection of this model serves as a baseline for assessing the influence that feature engineering has on the task of detecting spoilers.

4.3.1 Details of the Dataset Used for Training the Baseline Model

The dataset that was used for training and assessing the baseline model is comprised of movie review data that has not been cleaned or organized. I have decided to use this method in order to test the performance of the model on raw data without performing any preprocessing (detailed preprocessing can be found in section 3.10) in order to imitate a situation that occurs in the real world, where models are frequently required to deal with faulty data.

Tokenizing the reviews was a part of the training process. This was accomplished by utilizing the Hugging Face Transformers library's `AutoTokenizer`, which guarantees that the input data is in the appropriate format for the BERT model.

4.3.2 Training Configuration

The model was trained using the following configuration:

- Learning rate of 2e-5,
- Batch sizes of 32 for both training and evaluation,
- A total of 5 training epochs,
- Weight decay set to 0.01 to prevent overfitting.

Training was conducted on the split datasets, with evaluation at the end of each epoch to monitor performance improvements. The `Trainer` class from the Hugging Face library

was utilized, providing an efficient loop for training and evaluation while automatically handling device placement.

The model's performance was evaluated using weighted metrics for precision, recall, F1-score, and accuracy. These metrics were calculated from the predictions made on the test dataset, providing insights into the model's effectiveness straight out of the box without fine-tuning domain-specific data.

4.4 Dataset Production for Model Selection

The concept of feature engineering is an incredibly significant component when it comes to enhancing the performance of machine learning models, particularly in the context of natural language processing tasks. The purpose of this section is to provide an overview of the methodical approach that is applied in the process of creating and selecting characteristics that enhance the contextual understanding skills of models. The performance of revealer identification in movie reviews is going to be improved as a result of this section's efforts. A total of 20 different film evaluations were included in the dataset that was utilized because they provide thorough and lengthy movie summaries and synopses. This small dataset contains a total of 33,752 reviews over its entirety, which serves to provide a considerable amount of textual data to verify the hypothesis.

4.4.1 Training and Test Set Composition

As discussed in the 4.1, the first step in the process of training the model consisted of extracting extensive summaries and reviews from the context dataset. This procedure was developed with the intention of assembling the most comprehensive and pertinent contextual information that is accessible inside the movie database. The training set comprises 20,000 reviews, which were randomly selected to represent a diverse range of movies and review styles. The test set consists of 13,752 reviews, ensuring that the models are evaluated against unseen data to assess their generalizability and robustness.

4.5 MLM Training Details

To fine-tune the RoBERTa model and improve its capacity to comprehend and anticipate context inside movie reviews and synopses, the MLM (Masked Language Model) training approach is a crucial deciding factor.

4.5.1 Data Preparation for MLM

One of the first steps in the process is preparing the input text for the model. Tokenization of the text data was performed without truncation to maintain the complete context of the inputs. This was done because RoBERTa is designed to process inputs of a specific length. The outputs of the tokenization process were broken up into manageable parts so that the vast amount of movie reviews and synopses could be managed. Size of 512 tokens were used to divide each input into segments, which is a size that strikes a balance between the preservation of context and the efficiency of computing. Through the use of chunking, the model can effectively handle lengthy texts while preserving the integrity

of the narrative structure of the reviews and synopses. The tokenized text chunks were then grouped for the MLM training by utilizing a data collator that was customized for language modeling. The tokens in the text are masked randomly by the data collator, which prepares them for MLM, which is where the model makes predictions about the masked tokens. It is essential to complete this stage because it involves training the model to comprehend and produce material that is contextually appropriate. This enhances the model's capacity to cope with the sophisticated language that is present in spoilers.

4.5.2 MLM Model Configuration and Training

After adapting a pre-trained version to movie content, the RoBERTa model was fine-tuned on this data. The training was carried out under conditions that were meticulously calibrated: a relatively modest learning rate was utilized to make gradual changes to the model's weights, which assists in refining its predictions without overfitting. The training was monitored in an iterative manner utilizing evaluation techniques that were programmed to become active at the end of each epoch. This was done to guarantee that every training phase contributed to the learning of the model.

4.5.3 Parameter Selection and Optimization

To get the greatest possible performance out of the RoBERTa model while it is being used for MLM training, parameter optimization is necessary. Within the context of this optimization, the selection of the learning rate and batch sizes was an essential component. Both of these factors have a substantial influence on the training dynamics and the convergence of the model. Following the results of preliminary experiments, a learning rate of 5×10^{-5} was selected since it enables constant convergence without exceeding the minimum loss threshold. This rate is ideal for fine-tuning a pre-trained model where smaller, incremental updates retain and improve pre-learned weights.

The batch sizes for training and evaluation were set at 8 to strike a balance between the demand for computational efficiency and the requirement to reliably estimate gradient updates from a suitably diverse sample of examples. Smaller batch sizes increase gradient estimate noise, which can help avoid local minima but can also ruin training. Through the selection of this moderate size, the training process can remain steady while yet preserving sufficient diversity to guarantee extensive learning.

To avoid overfitting, the training setup employed a weight decay of 0.01 as a regularization method. Complex models like RoBERTa remember training data better than generalize it, making regularization essential. For training, weight decay favors simpler models. This improves generalization to new data.

4.5.4 Classification Training Parameters

After MLM fashion training, the model was trained for the classification task on the smaller train set. In this section, I will outline the key training parameters used in the model optimization process. The parameters are critical to ensuring the proper learning behavior of the model and fine-tuning the performance on the dataset. It was kept the same while training with different features 4.7.

- **Learning Rate:** `2e-5`

The learning rate determines how much to change the model in response to the estimated error each time the model weights are updated.

- **Train Batch Size:** `16`
  The number of training examples used in one forward/backward pass. A batch size of 16 is used for training.
- **Evaluation Batch Size:** `16`
  The batch size used during evaluation.
- **Number of Epochs:** `3`
  The number of complete passes through the training dataset.
- **Weight Decay:** `0.01`
  A regularization technique used to prevent overfitting by penalizing large weights in the model.
- **Random Seed:** `42`
  A fixed random seed to ensure reproducibility of the results.
- **Optimizer:** `AdamW`
  The AdamW optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 1e-8$ was used for parameter updates.

4.6 LLM Inference

To leverage the reasoning capabilities of Large Language Models (LLMs) for spoiler classification, we implemented a chain-of-thought prompting approach that guides the model through a structured decision-making process. This methodology enables the model to articulate its reasoning steps before making final classification decisions about potential spoilers in movie reviews.

4.6.1 Chain-of-Thought Implementation

The implementation of chain-of-thought prompting involves constructing carefully designed prompts that encourage the LLM to break down its analysis into discrete steps. Rather than directly classifying text segments as spoilers or non-spoilers, the model first evaluates various aspects of the content, including plot significance, reveal timing, and narrative impact. This step-by-step reasoning process helps ensure more reliable and interpretable classifications.

4.6.2 Prompt Design and Structure

The prompting structure was developed to guide the model through several critical analytical steps:

- Initial content assessment to identify key story elements.
- Evaluation of whether these elements reveal crucial plot points.
- Analysis of the temporal placement of revelations within the narrative.

- Final determination of spoiler status based on accumulated reasoning.

The prompt template was carefully calibrated to maintain consistency across different types of movie reviews while remaining flexible enough to accommodate varying content structures and writing styles (prompts can be found on Appendix D). This approach helps the model maintain reliable performance across diverse input formats.

4.6.3 Inference Parameters and Optimization

To ensure optimal performance during inference, several key parameters were carefully tuned:

- **Temperature Parameter:** Set to **0.2**, striking a balance between deterministic responses and creative reasoning. This setting allows the model to maintain consistent classification decisions while still considering nuanced context variations.
- **Maximum Token Length:** Configured to accommodate both the input text and the chain-of-thought reasoning process, ensuring that no critical context is truncated during analysis.
- **Response Formatting:** Structured to maintain clear delineation between reasoning steps and final classifications. This facilitates both human interpretation and automated processing of model outputs, enabling effective post-processing and integration with broader classification systems.

4.6.4 Performance Considerations

While chain-of-thought prompting introduces additional computational overhead compared to direct classification, the improved accuracy and interpretability justify the increased processing time. The approach provides several key advantages:

- Enhanced explainability of classification decisions.
- More robust handling of edge cases through explicit reasoning.
- Easier identification of potential classification errors through examination of the reasoning chain.
- Greater consistency in handling complex narrative structures.

These benefits make the additional computational cost an acceptable trade-off for achieving more reliable spoiler detection in movie reviews.

4.7 Model Training with Specific Features

The experimentation with particular features was conducted to determine how different combinations of textual properties could affect the efficiency of various model architectures in detecting spoilers. This method assists in comprehending the influence of supplementary information, which includes sentiment, text length, and summaries, in addition to the conventional text input. The premise that additional textual information can boost a

model's capacity to recognize complicated patterns and contexts that are relevant to spoiler detection was the basis for the creation of each feature set, which was developed to test the hypothesis. To analyze the performance of the system over a variety of topologies, many model classes were utilized. This practice helped to improve the robustness and generalizability of the findings. Each feature set was ran on transformer based `BERT-base-uncased`, `DistilBERT-uncased`, and `RoBERTa-base-uncased`.

Each of these feature sets was carefully designed and used to study how extra data layers affect spoiler detection methods. The architectures were chosen based on their computing efficiency and ability to handle several data inputs to ensure a diverse testing environment.

4.7.1 Review Text Only

This portion concentrates on training three separate models with solely the review text, adhering to the methods employed in the baseline model configuration. This method facilitates a direct comparison of the performance of several models under consistent settings, utilizing the sanitized, pre-processed text data from the previously outlined data purification pipeline (3.10).

Training Setup Consistency

All models in this experiment were trained with the same learning rate, batch size, number of epochs, and evaluation procedures to ensure comparability with the baseline (see 4.5.4). This uniformity ensures that any observed performance disparities are due to the model architectures themselves, rather than variations in training parameters.

Model Architectures

The models selected for this experiment include:

- `BERT-base-uncased`: Known for its efficiency and effectiveness in handling a variety of NLP tasks, serving as the initial benchmark (further information on C.6.1).
- `DistilBERT-uncased`: A lighter and faster version of BERT that maintains 97% of BERT's performance while being significantly smaller and more efficient. It is ideal for scenarios requiring reduced computational overhead (further information on C.6.4).
- `RoBERTa-base`: An optimization of BERT architecture that has been pre-trained on a larger corpus and with more training steps, designed to provide better performance on tasks requiring a deep understanding of context (further information on C.6.3).

Hypothesis

Using simply the review text was to test each model's spoiler detection with no extra features or context. This setup explores the idea that advanced models like BERT and RoBERTa can distinguish spoiler-containing and non-spoiler reviews from text alone. This methodological consistency across model architectures provides a solid framework for assessing model capabilities. This investigation will help determine NLP models' spoiler detection baseline effectiveness and provide a comparative basis for future feature-augmented experiments.

4.7.2 Review Text with Review Sentiment Combined

Using review sentiment in the training phase is an advanced feature engineering technique to assess how emotional context affects spoiler identification accuracy. This section examines how sentiment analysis and review text are used for binary classification.

Sentiment Extraction

Sentiment scores were extracted using the state-of-the-art sentiment analysis model provided by Hugging Face, specifically `roberta-llama3.1405B-twitter-sentiment` (further information on C.6.6). This machine can accurately analyze text emotions and provide a continuous sentiment score for spoiler identification.

Model Architecture

A BERT-based architecture that is supplemented with a sentiment-aware feature transformation is utilized in the design of the bespoke spoiler detection model. Detailed explanations of the mathematical formulation and structure of the model components are provided in this section.

BERT Module The core of the model is based on the BERT (*Bidirectional Encoder Representations from Transformers*) architecture. The BERT module is denoted as:

$$\mathbf{h} = \text{BERT}(\mathbf{x})$$

where $\mathbf{x}$ represents the input token embeddings and $\mathbf{h}$ denotes the sequence of hidden states corresponding to each token. The model utilizes the pre-trained weights from the BERT-based models.

Feature Transformation for Sentiment Sentiment scores which were extracted via the RoBERTa-LLaMA model are incorporated to enhance the contextual awareness of the BERT outputs. The transformation of the sentiment score s into a representation compatible with the hidden states from BERT is achieved through a linear transformation:

$$\mathbf{s}' = \mathbf{W}_s \mathbf{s} + \mathbf{b}_s$$

where $\mathbf{W}_s \in \mathbb{R}^{d_h \times 1}$ and $\mathbf{b}_s \in \mathbb{R}^{d_h}$ are trainable parameters, d_h is the dimensionality of the hidden state, and $\mathbf{s}'$ is the transformed sentiment feature.

Classification Layer The final classification decision is made by concatenating the transformed sentiment vector $\mathbf{s}'$ with the pooled output of the BERT module, typically using the representation of the [CLS] token, denoted as $\mathbf{h}_{[CLS]}$. This combined feature vector is then passed through a linear classification layer:

$$\mathbf{z} = \mathbf{W}_c[\mathbf{h}_{[CLS]}; \mathbf{s}'] + \mathbf{b}_c$$

$$\hat{y} = \sigma(\mathbf{z})$$

where $\mathbf{W}_c \in \mathbb{R}^{2\times(d_h+d_h)}$ and $\mathbf{b}_c \in \mathbb{R}^2$ are the weights and biases of the classifier, respectively, designed to output logits for the two classes (spoiler and non-spoiler). The function σ represents the softmax function that normalizes the logits into probabilities.

Loss Function During training, the model optimizes the cross-entropy loss between the predicted probabilities $\hat{y}$ and the true labels y. This ensures that the model effectively learns to discriminate between spoilers and non-spoilers based on both textual and emotional cues:

$$L = -\frac{1}{N}\sum_{i=1}^{N} y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)$$

where N is the number of training examples.

Training Process

The training process involved the following steps:

- **Data Preparation**: Tokenized review text and sentiment scores were fed to the model. Special attention was made to match sentiment scores to reviews.
- **Loss and Optimization**: The model was trained using cross-entropy loss to optimize primary text and sentiment components. A learning rate of 2×10^{-5} was used for dual-focus training to balance quick learning with avoiding overshooting ideal weights.
- **Batch and Epoch Configuration**: The model was trained across 5 epochs using 32 batches for training and evaluation to ensure data exposure and computational efficiency.

Hypothesis

The integration of the review sentiment hypothesis states that spoilers greatly affect review's emotional tone. Spoilers often contain happy or negative emotions since they discuss key story details. Emotional extremes in reviews may indicate spoiler-filled narrative surprises. The program uses sentiment analysis to predict spoilers by using these emotional cues. Testing this hypothesis entails measuring model performance metrics including accuracy, recall, and F1-score when sentiment is added to textual content.

4.7.3 Review Text, Review Sentiment, and Review Length Combined

It is based on the assumption that reviews with longer lengths and more emotive bias have a higher likelihood of including spoilers because they may provide more in-depth explanations or discussions that unintentionally expose plot secrets. The incorporation of review duration as a feature alongside review sentiment is based on this premise. This section delineates the model adaptation and training process. The technique considers both the length of the review and the content of the text.

Model Architecture

The architecture of the Spoiler Classifier model comprises a pre-trained BERT encoder and an auxiliary feature processing algorithm. The model incorporates the primary review text together with additional features, including sentiment and review length, to execute binary classification. The architecture is mathematically delineated as follows:

Let $\text{BERT}(X)$ represent the BERT model that converts input sequence X to hidden representations, and $\text{CLS}(H)$ extract the [CLS] token representation from hidden states H. Furthermore, the concatenation of the sentiment and review length characteristics a is subjected to a linear transformation, $\mathbf{F}_\theta(a)$, where θ represents the transformation's parameters.

The model takes the following inputs:

- x: The tokenized input of the review text is processed by the BERT model to extract contextualized word embeddings.
- m: The attention mask corresponds to the tokenized review text.
- s: The sentiment feature which represents an external numeric value indicating the sentiment score of the review.
- l: The review length feature which encodes the length of the review as a numeric value.
- y: The ground truth labels.

The forward pass is described as follows:

Main Review Representation: The tokenized review input x is passed through the BERT model along with the attention mask m. BERT outputs the hidden states of each token in the sequence:

$$H = \text{BERT}(x, m)$$

From the hidden states H, the [CLS] token representation h is extracted. This token is specifically designed to capture the aggregated meaning of the entire input sequence:

$$h = \text{CLS}(H)$$

Additional Features Transformation: The sentiment feature s and review length feature l are concatenated to form a combined feature vector:

$$a = [s; l]$$

This concatenated feature vector a is passed through a linear transformation layer to map the combined features into a hidden size compatible with the BERT output dimension:

$$f_a = \mathbf{F}_\theta(a)$$

Here, $\mathbf{F}_\theta$ is a linear transformation that projects the additional features into the same space as the BERT hidden states.

Concatenation of Features: The transformed features f_a are concatenated with the BERT [CLS] token representation h:

$$h_c = [h; f_a]$$

Classification Layer: The concatenated or final representation h_c is passed through a linear classifier. The classifier applies a linear transformation to produce logits, which represent the unnormalized predictions for the two output classes (spoiler or not spoiler):

$$z = W \cdot h_c + b$$

where W is the weight matrix and b is the bias term of the classifier.

Loss Function: During training, the model computes the cross-entropy loss $\mathcal{L}$ by having the ground truth labels y:

$$\mathcal{L} = \text{CrossEntropy}(z, y)$$

This loss is minimized by adjusting the model parameters during training to improve classification accuracy.

Hypothesis

The primary hypothesis of this experiment reveals a correlation between review length and spoiler likelihood when both parameters are included. Incorporating review length and sentiment into the model is expected to improve its capacity to identify lengthy reviews with sentimental bias as potential spoiler signs. The model will exploit both text content and volumetric properties through this connection.

4.7.4 Review Text, Review Sentiment, and Review Summary Combined

This subsection discusses the integration of review text, review summary, and sentiment analysis into a unified model to potentially enhance the detection of spoilers in reviews. This approach is designed to exploit both the detailed content of reviews and the condensed essence in summaries, along with the emotional tone captured through sentiment analysis.

Model Architecture

The architecture of the Spoiler Classifier model integrates a pre-trained BERT encoder with additional layers designed to incorporate the main review text, review summary, and an optional sentiment feature. The model performs binary classification to detect the presence of spoilers. The architecture can be described as follows:

Let BERT(X) denote the BERT model that maps an input sequence X to its hidden states, and let CLS(H) extract the [CLS] token representation from the hidden states H. Additionally, a linear transformation, $\mathbf{F}_\theta(s)$ is applied to the sentiment feature s where θ are the transformation parameters.

The model takes the following inputs:

- x_r: The tokenized input of the main review.

- m_r: The attention mask for the main review.
- x_s: The tokenized input of the review summary.
- m_s: The attention mask for the review summary.
- s: An external sentiment feature.
- y: The ground truth labels.

The forward pass is described as follows:

Main Review Representation: The tokenized input of the main review x_r is passed through the BERT model along with the attention mask m_r, producing hidden representations for each token in the sequence:

$$H_r = \text{BERT}(x_r, m_r)$$

From the hidden states H_r, the [CLS] token representation h_r is extracted. This token is designed to summarize the entire review sequence for classification purposes:

$$h_r = \text{CLS}(H_r)$$

Review Summary Representation: The tokenized review summary x_s is processed in the same manner by the BERT model along with the attention mask m_s, generating hidden states H_s for the summary:

$$H_s = \text{BERT}(x_s, m_s)$$

The [CLS] token representation h_s is then extracted from the review summary's hidden states:

$$h_s = \text{CLS}(H_s)$$

Sentiment Feature Transformation: The optional sentiment feature s is first expanded and transformed using a linear layer to match the hidden dimension:

$$f_s = \mathbf{F}_\theta(s)$$

Here, $\mathbf{F}_\theta$ is a linear transformation that adjusts the sentiment feature to be compatible with the BERT-based representations.

Concatenation of Features: The final representation is formed by concatenating the main review representation h_r, the review summary representation h_s, and the transformed sentiment feature f_s:

$$h_c = [h_r; h_s; f_s]$$

Classification Layer: The concatenated representation h_c is passed through a linear classification layer, which produces logits representing the model's raw predictions for the binary classification task (spoiler vs. non-spoiler):

$$z = W \cdot h_c + b$$

where W is the weight matrix and b is the bias term of the classifier.

Loss Function: If the ground truth labels y are provided, the model computes the cross-entropy loss $\mathcal{L}$:

$$\mathcal{L} = \text{CrossEntropy}(z, y)$$

Hypothesis

Combining textual analysis from the full review and summary with sentiment analysis' emotional insights may provide a more accurate spoiler predictor. This underpins this model setup. This technique suggests that summaries should focus on key narrative elements that may expose spoilers, while emotions should highlight emotional responses that help understand spoilers. If it can examine inputs synergistically, the model should spot spoilers more accurately.

4.8 Final Model Training

After an intensive investigation of numerous models, as described in sections 4.5 and 4.7, the final model was chosen due to its superior performance across key criteria as presented in the result chapter 5. The model that uses review text, and sentiment analysis had the highest average F1-score, indicating that it can detect spoilers fairly and accurately. The full dataset was used for the final training phase, which was much bigger than the one used in this testing section.

Training approach improvements were considered and executed for this larger dataset to optimize model performance. The learning rate was fine-grained and adjusted to prevent overfitting and allow the model to converge on a global minimum. Batch sizes and regularization methods were chosen to balance model accuracy and processing efficiency. The model's training was continuously monitored with expanded recording and metrics review to ensure it was going as planned and to allow real-time modifications. These adjustments were designed to improve the model's robustness and generalization over a wide range of content, preparing it for real-world spoiler detection situations. As expected, this final training phase should increase the model's predicted accuracy and generalization by accessing more data. The upcoming chapter of the report will include a detailed performance review.

4.9 Decision Visualization

In this section, I present a novel approach to visualizing the decision-making process of a machine learning model. This visualization technique involves splitting the input text into overlapping chunks using a sliding window approach. For each chunk, I analyze its contribution to the model's prediction by comparing the model's classification logits with the original prediction. This process helps prioritize specific text segments that have a greater influence on the final classification outcome.

4.9.1 Methodology

Text Chunking Using Sliding Window

The input text is divided into overlapping chunks using a sliding window technique. The text is split into a series of chunks of a fixed size, and for each chunk, we also generate the rest of the text excluding that chunk.

The following function is used to split the text:

$$\text{chunking}(T, c_s, s_s) \rightarrow (C, R)$$

Where:

- T: the input text.
- c_s: the chunk size.
- s_s: the step size.
- C: the set of generated chunks.
- R: the corresponding set of remaining text without each chunk.

The sliding window mechanism allows each chunk to overlap with adjacent ones, providing an in-depth retrival of the text segments.

Impact Analysis of Chunks

To quantify the influence of each chunk on the model's decision, I compare the extracted logits of the original text with the logits of the chunked text and the remaining text. The impact of removing a chunk from the text is measured by computing the L2-norm between the original and modified logits:

$$\text{impact}(L_{orig}, L_{chunk}) = \|L_{orig} - L_{chunk}\|_2$$

Where:

- L_{orig}: logits of the original text.
- L_{chunk}: logits when only the chunk is classified.
- L_{rest}: logits when the rest of the text is classified.

This process is repeated for all chunks, and the results are stored for visualization.

4.9.2 Mathematical Representation

The following algorithm 6 describes the process of chunking the text and calculating the impact values for each chunk and the remaining text:

Algorithm 6 Analyze Chunks and Compute Impact

Input:

- T: Text to be analyzed
- c_s: Chunk size
- s_s: Step size for sliding window

Output:

- *results*: List of tuples containing chunk, rest text, impact values, and labels
- L_{orig}: Original label of the full text

$C, R \leftarrow \text{chunking}(T, c_s, s_s)$
$L_{orig}, \text{label}_{orig} \leftarrow \text{classify\_spoiler}(T)$
$results \leftarrow []$
for each (c_i, r_i) in (C, R) **do**
    $L_{chunk}, \text{label}_{chunk} \leftarrow \text{classify\_spoiler}(c_i)$
    $L_{rest}, \text{label}_{rest} \leftarrow \text{classify\_spoiler}(r_i)$
    $\text{impact\_chunk} \leftarrow \text{impact}(L_{orig}, L_{chunk})$
    $\text{impact\_rest} \leftarrow \text{impact}(L_{orig}, L_{rest})$
    $results \leftarrow \text{Append } (c_i, r_i, \text{impact\_chunk}, \text{impact\_rest}, \text{label}_{chunk}, \text{label}_{rest})$
end for
return *results*, L_{orig}

Visualization of Results

The next is to use the output of algorithm 6 to visualize the results. I have visualized the impacts of the chunks and the rest of the text using horizontal bar charts. The bars represent the magnitude of impact, and their color indicates the model's classification label for that text segment. Specifically:

- Red bars represent text chunks or rest classified as spoilers.
- Green bars represent text classified as non-spoilers.

This allows us to visually identify which text segments (chunks or rest) contribute more significantly to the overall prediction. An example of the visualization is as follows by the figure 4.2. More on the discussion chapter 6.

4.9.3 Rationale and Reasoning

The intuition behind this approach is that text classification models rely on specific segments of text to form their decisions. By using a sliding window chunking technique, we can isolate portions of the text and assess their direct influence on the model's output. The L2-norm is chosen as a measure of impact because it provides a straightforward and interpretable distance between the original and modified logits. This highlights the degree of change in the model's decision space.

This approach should work effectively because it dissects the input text into manageable parts by allowing for a granular understanding of the model's decision-making process.

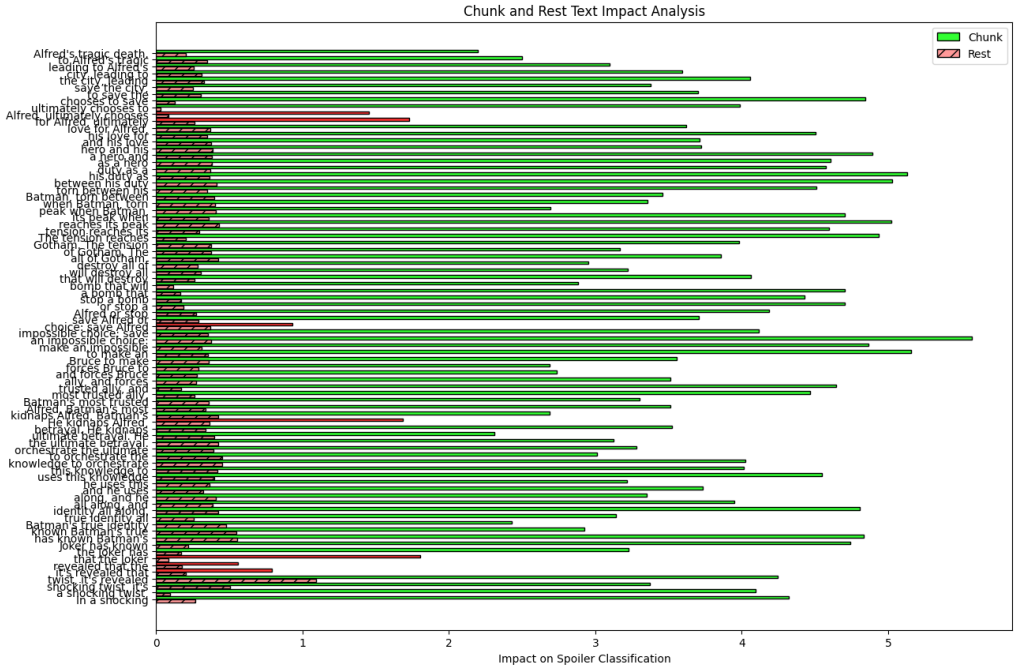

Figure 4.2: An example of a spoiler classification chunk visualization.

The impact values serve as a proxy for the model's sensitivity to different portions of the text.

Chapter 5
Results

This chapter will compare the effects of various feature set combinations of the three models: BERT, RoBERTa, and DistilBERT. Additionally, I will evaluate the performance of the top models in comparison to the baseline spoiler detection model. In conclusion, I will compare the present architectures with the most effective one.

5.1 Masked Language Model (MLM) Pretraining

In this section, I will describe the pretraining of the Masked Language Model (MLM) and its impact on the subsequent classification task. Only RoBERTa base was used for this task considering computational resources.

5.1.1 MLM Pretraining Results

During the pre-training phase, the training loss and validation loss decreasing steadily over time, as shown in Table 5.1. This indicates that the model was learning effectively during pretraining.

Epoch	Training Loss	Validation Loss
1	4.6788	4.7173
2	4.0424	3.5975
3	3.3600	3.1186
4	3.1241	2.8922
5	2.8573	2.8449

Table 5.1: Training and Validation Loss for MLM Pretraining (Epochs 1-5)

The consistent drop in training and validation loss shows that the model generalized well and minimized overfitting. From 4.7173 to 2.8449, validation loss improved dramatically during pretraining.

5.1.2 Classification Task Results

The model was fine-tuned for binary classification after MLM pretraining. The purpose was to classify input instances as 0 (non-spoiler) or 1 (spoiler). To determine how MLM pretraining affected the downstream task, the model's performance was compared before and after pretraining.

Initial Classification Results (Without MLM Pretraining)

Before using the MLM-pretrained model, the classification task was performed with a standard pre-trained model. The results are summarized in Table 5.2.

Class	Precision	Recall	F1-Score
0	0.77	0.47	0.58
1	0.24	0.54	0.33
Accuracy		0.49	
Macro Avg	0.50	0.50	0.46
Weighted Avg	0.64	0.49	0.52

Table 5.2: Classification Results Without MLM Pretraining

The model without MLM pretraining had a 49% accuracy. The model had 0.24 accuracy and 0.33 F1-score for the spoiler class, but 0.77 for the non-spoiler class. This suggests that the model performs better on detecting non-spoiler than spoiler.

Classification Results After MLM Pretraining

After incorporating MLM pretraining, the model was again fine-tuned for the classification task on the small training dataset. Table 5.3 presents the performance metrics after MLM pretraining.

Class	Precision	Recall	F1-Score
0	0.77	1.00	0.87
1	0.00	0.00	0.00
Accuracy		0.77	
Macro Avg	0.38	0.50	0.43
Weighted Avg	0.59	0.77	0.66

Table 5.3: Classification Results After MLM Pretraining

While MLM pretraining improved the overall accuracy to 77%, the model still exhibited significant shortcomings. It achieved perfect recall for class 0, but completely failed to predict the spoiler class (precision and recall of 0.00). The weighted average F1-score improved slightly (0.66 compared to 0.52), but the model became highly imbalanced while favoring class 0 almost exclusively.

5.1.3 Conclusion

The results of the classification task highlight several key takeaways:

- **Improved Accuracy but Imbalanced Predictions:** Although MLM pretraining increased classification accuracy from 49% to 77%, the model became strongly skewed towards the non-spoiler class. This shows that the MLM-pretrained model mastered the task better but overfitted to the non-spoiler class due to dataset or learning dynamics imbalances.
- **Spoiler Class Detection:** Before and after MLM pretraining, the model had trouble detecting the spoiler class instances. This could be caused to a dataset class imbalance or inadequate the spoiler class training instances.

In conclusion, MLM pretraining improved accuracy but left the model unbalanced. Future work should address this bias and improve the spoiler class instance detection to produce a more robust and balanced model.

5.2 Review Text Alone

Each model was trained using only the review text, and their performance was compared across several key metrics. Full comparison of this task is present on figure 5.1.

F1 Score Comparison Among the models, RoBERTa demonstrated the highest average F1 score of 66.05%, followed by BERT with 64.39% and DistilBERT with 63.43%. RoBERTa's superior F1 score indicates its ability to balance precision and recall effectively, making it the most reliable model for spoiler detection tasks. The slight differences in F1 scores suggest that all models perform similarly, but RoBERTa consistently manages false positives and false negatives better than the others.

Precision and Recall For spoiler detection, recall is more critical than precision. Missing a spoiler is more harmful than misclassifying a non-spoiler as a spoiler. RoBERTa had the highest recall at 76.50%, ensuring that most spoilers were correctly identified. BERT followed closely with a recall of 74.64%, and DistilBERT had the lowest recall at 74.26%.

Precision, while less important than recall in this case, still affects user experience. RoBERTa also led in precision, achieving 59.07%, whereas BERT had 57.25%, and DistilBERT had 56.02%. This means RoBERTa was better at minimizing false positives, providing a better balance between identifying spoilers and not over-flagging non-spoiler content.

Accuracy and Loss RoBERTa achieved the highest average test accuracy of 76.50%, with BERT at 74.64% and DistilBERT at 74.25%. This aligns with the previous metrics, confirming that RoBERTa was the most accurate at classifying spoiler and non-spoiler reviews.

Regarding loss, both RoBERTa and DistilBERT had a similar average test loss, around 0.4408 and 0.4408 respectively, while BERT had a slightly higher loss at 0.4495. Lower test loss indicates better generalization, and RoBERTa's performance shows that it generalizes well to unseen data.

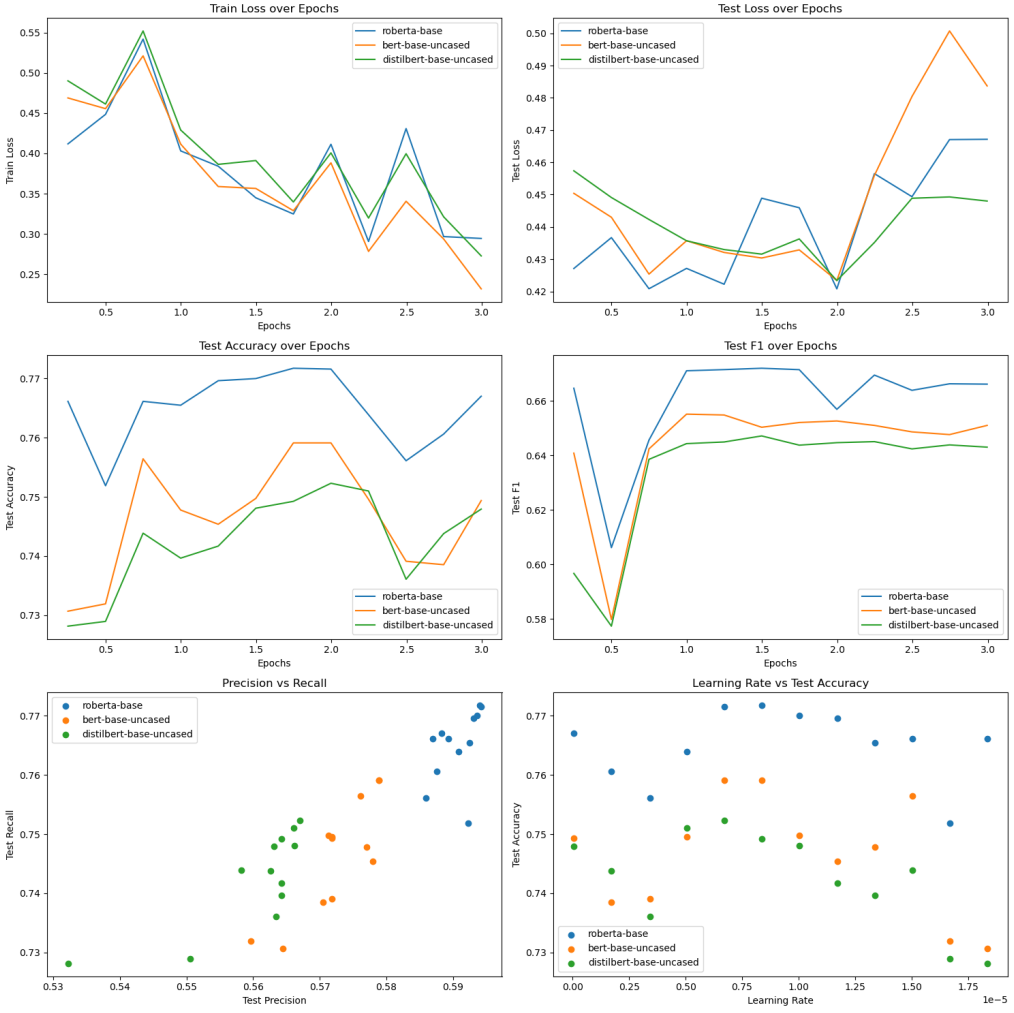

Figure 5.1: Model performance comparison: Review Text Alone

Conclusion Based on these results, RoBERTa emerges as the best model for spoiler detection using review text alone. It consistently outperformed BERT and DistilBERT across critical metrics. RoBERTa's high recall ensures that most spoilers are detected, while its balance with precision minimizes false positives. Its higher accuracy and lower loss further highlight its robustness and reliability.

Therefore, RoBERTa is chosen as the most suitable model for this task, setting a solid baseline for future experiments involving additional features or context to improve performance.

5.3 Review Text with Review Sentiment Combined

Each model was trained using both the review text and the review sentiment as features, and their performance was compared across several key metrics. Full comparison of this task is presented in figure 5.2.

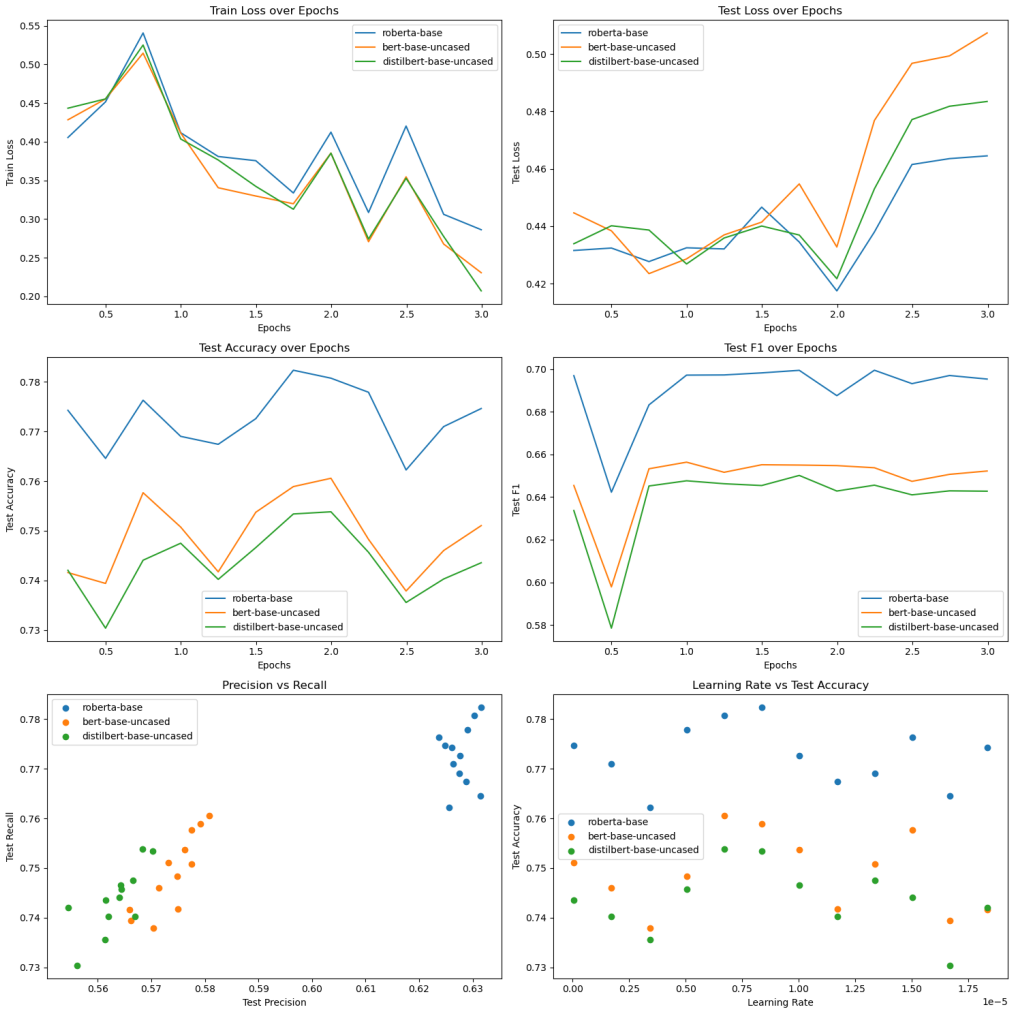

Figure 5.2: Model performance comparison: Review Text with Review Sentiment Combined

F1 Score Comparison Among the models, RoBERTa demonstrated the highest average F1 score of 69.06%, followed by BERT with 64.78%, and DistilBERT with 63.84%. RoBERTa's superior F1 score reflects its ability to incorporate both text and sentiment information effectively, making it the most reliable model for spoiler detection in this setup. The higher F1 score of RoBERTa suggests it consistently balances precision and recall better than the other models.

Precision and Recall For spoiler detection, recall remains more critical than precision due to the importance of identifying spoilers correctly. RoBERTa achieved the highest recall at 77.27%, ensuring that most spoilers were detected. BERT followed with a recall of 74.89%, and DistilBERT had the lowest recall at 74.36%.

While recall is prioritized, precision plays a role in minimizing false positives. RoBERTa also led in precision with 62.77%, outperforming BERT's 57.40% and DistilBERT's 56.33%. This shows that RoBERTa is more effective at avoiding misclassifications when combining sentiment and text.

Accuracy and Loss RoBERTa achieved the highest average test accuracy of 77.27%, with BERT at 74.90% and DistilBERT at 74.36%. RoBERTa's accuracy aligns with its strong performance across other metrics, confirming its superiority for this task.

In terms of loss, RoBERTa exhibited the lowest average test loss at 0.4402, indicating better generalization to unseen data compared to BERT (0.4568) and DistilBERT (0.4475). This reinforces RoBERTa's robustness and its ability to avoid overfitting.

Conclusion Based on these results, RoBERTa once again emerges as the best model for spoiler detection when using both review text and sentiment. Its high recall ensures that spoilers are detected more effectively, while its balance with precision reduces false positives. Its higher accuracy and lower loss further confirm its reliability in this setup.

Therefore, RoBERTa is the most suitable model for this task, establishing a strong baseline for future research involving more complex features or additional context.

5.4 Review Text, Review Sentiment, and Review Length Combined

Each model was trained using the review text, review sentiment, and review length as combined features, and their performance was compared across several key metrics. Full comparison of this task is presented in figure 5.3.

F1 Score Comparison RoBERTa demonstrated the highest average F1 score of 66.20%, followed by BERT with 64.64%, and DistilBERT with 63.77%. RoBERTa's superior F1 score indicates that it handles the combination of text, sentiment, and length features better than the other models, achieving a more consistent balance between precision and recall.

Precision and Recall For spoiler detection, recall remains more critical, as missing a spoiler is more detrimental than over-flagging non-spoiler content. RoBERTa achieved the highest recall at 76.16%, followed by BERT at 74.60%, and DistilBERT at 74.31%.

Precision, while still important, plays a secondary role. RoBERTa led with a precision of 59.07%, whereas BERT reached 57.21%, and DistilBERT achieved 56.30%. RoBERTa's ability to balance both recall and precision makes it the most effective model for this task.

Accuracy and Loss RoBERTa achieved the highest test accuracy at 76.16%, with BERT at 74.60% and DistilBERT at 74.31%. RoBERTa's test loss was also the lowest,

averaging 0.4408, which shows that it generalizes better compared to BERT (0.4581) and DistilBERT (0.4434). This demonstrates RoBERTa's ability to learn effectively from the combined features while avoiding overfitting.

Conclusion Based on these results, RoBERTa once again emerges as the best model when using the combined features of review text, sentiment, and length. Its high recall ensures spoilers are detected more reliably, while its better balance with precision reduces false positives. RoBERTa's higher accuracy and lower loss confirm its superiority and reliability in this configuration.

Therefore, RoBERTa is chosen as the most suitable model for this task, setting a strong foundation for future experiments involving more complex features.

5.5 Review Text, Review Sentiment, and Review Summary Combined

Each model was trained using review text, review sentiment, and review summary as combined features, and their performance was compared across several key metrics. Full comparison of this task is presented in figure 5.4.

F1 Score Comparison Among the models, RoBERTa demonstrated the highest average F1 score of 66.32%, followed by BERT with 64.43%, and DistilBERT with 63.78%. RoBERTa's higher F1 score shows its ability to effectively balance precision and recall using the combined features of text, sentiment, and summary.

Precision and Recall For spoiler detection, recall is critical, as it ensures that spoilers are detected more reliably. RoBERTa achieved the highest recall at 76.36%, followed by BERT at 74.76%, and DistilBERT at 74.41%.

Although recall is prioritized, precision still impacts the model's ability to avoid false positives. RoBERTa had the highest precision at 58.96%, while BERT and DistilBERT reached 57.09% and 56.29%, respectively. RoBERTa's better balance between recall and precision shows its robustness in handling this task.

Accuracy and Loss RoBERTa had the highest average test accuracy at 76.36%, with BERT at 74.76% and DistilBERT at 74.41%. RoBERTa also achieved the lowest test loss at 0.4404, further proving its ability to generalize effectively. In comparison, BERT had a loss of 0.4585, and DistilBERT followed with 0.4407.

Conclusion Based on these results, RoBERTa once again emerges as the most suitable model for spoiler detection when combining review text, sentiment, and summary features. Its high recall ensures that most spoilers are detected, while its balance with precision reduces false positives. RoBERTa's superior accuracy and lower loss reinforce its reliability in this setup.

Therefore, RoBERTa remains the top choice for this task, establishing a strong baseline for future experiments involving additional contextual features.

5.6 Pipeline Performance Comparison

This particular experimental evaluation considered several key performance indicators across the implemented pipelines. Figure 5.5 presents a comprehensive visualization of these metrics.

F1 Score Analysis The experimental results revealed varying levels of effectiveness across my implementations. LLM Inference demonstrated strong performance with an F1 score of 71%, while the Feature Pipeline achieved 70%. The MLM Training approach showed promising but slightly lower results at 66%. These scores suggest that both LLM Inference and Feature Pipeline implementations offer robust solutions for the classification task.

Precision-Recall Characteristics The analysis revealed interesting trade-offs between precision and recall across implementations. The Feature Pipeline exhibited strong recall performance at 78%, closely matched by MLM Training at 77%. LLM Inference showed a different performance profile, achieving 69% recall but leading in precision at 76%. This precision advantage of LLM Inference stands in contrast to the Feature Pipeline's 63% and MLM Training's 59% precision rates.

Overall Accuracy Examining accuracy metrics revealed notable variations among approaches. The Feature Pipeline achieved 78% accuracy, with MLM Training following closely at 77%. Despite its strong precision, LLM Inference reached 69% accuracy. These results demonstrate the Feature Pipeline's consistency across different evaluation criteria.

Synthesis of Findings My experimental results suggest that the Feature Pipeline offers the most balanced performance profile for practical applications. While each implementation showed distinct strengths - such as LLM Inference's superior precision and the Feature Pipeline's strong recall - the overall metrics favor the Feature Pipeline approach. Its robust performance across multiple criteria, particularly in recall and accuracy, indicates strong potential for real-world applications. These findings provide valuable insights for future research directions and potential system improvements.

5.7 Feature Combination and Final Model Selection

In this experiment, I have evaluated four different combinations of features:

1. Review Text Alone
2. Review Text with Review Sentiment
3. Review Text with Review Sentiment and Review Length
4. Review Text with Review Sentiment and Review Summary

Each feature combination was tested across three models: BERT, DistilBERT, and RoBERTa. The performance of each model was compared based on key metrics such as F1 score, precision, recall, accuracy, and test loss.

5.7.1 Best Performing Feature Combination

The results indicate that the "Review Text with Review Sentiment Combined" feature set consistently delivered the best performance. RoBERTa achieved the highest F1 score of 69.06%, along with the best recall of 77.27% and a strong precision score of 62.77%. This combination of features significantly improved the model's ability to detect spoilers, surpassing other feature combinations in terms of both accuracy and generalization capabilities.

The use of sentiment as an additional feature proved highly beneficial for spoiler detection. Sentiment provides valuable emotional context, helping the model distinguish spoiler content based on the emotional tone of the review, which was not as effectively captured in the other combinations. This feature combination outperformed even more complex configurations like the inclusion of review length or summary.

5.7.2 Final Model Recommendation

RoBERTa consistently outperformed both BERT and DistilBERT in all feature combinations, but its performance peaked when using the "Review Text with Review Sentiment" feature combination. RoBERTa's high recall ensures that the majority of spoilers are correctly identified, while its balanced precision minimizes false positives. The model's accuracy and low test loss further reinforce its stability and capability to generalize well to unseen data.

Conclusion Based on the results of these experiments, the "Review Text with Review Sentiment" combination offers the best performance for spoiler detection. RoBERTa, with this feature combination, achieved the highest F1 score, recall, and accuracy, making it the most suitable model. Therefore, I have decided to train RoBERTa on the larger dataset with the "Review Text with Review Sentiment" feature combination for optimal results.

5.8 Baseline Model Performance

The performance of the baseline model, which was applied straight to raw text data without any kind of preprocessing like tokenization, stemming, or stopword removal, is assessed in this section. Table 5.8 presents the findings.

No prior transformation or cleaning was done to the input before the baseline model was run on raw text data. In text classification jobs, preparation processes are typically used to enhance performance by lowering noise and converting the data into a more readable format. Nevertheless, no such methods were used for this baseline to evaluate the model's raw ability to handle raw data. Although this method enables us to assess the resilience of the model. It is anticipated that the absence of preprocessing may have had a detrimental effect on the model's capacity for generalization, especially when dealing with intricate linguistic patterns or non-standard textual content.

The model's ability to correctly forecast the non-spoiler class reflects the fact that this class was significantly more prevalent in the dataset. Table 5.8 shows that the model's precision for non-spoilers was 0.85, meaning that 85% of the cases that were predicted to be non-spoilers were accurate. At 0.95, the recall for this class was even higher, indicating

Class	Precision	Recall	F1-score	Support
0 (Negative Class)	0.85	0.95	0.90	232,481
1 (Positive Class)	0.64	0.34	0.44	60,572
Accuracy	0.73			
Weighted avg	0.74	0.64	0.67	293,053

Table 5.4: Classification report for the baseline model applied on raw text data

that 95% of the real non-spoiler cases could be properly identified by the model. Despite the lack of preprocessing steps, the model's outstanding performance for this class is demonstrated by its F1-score of 0.90. Given that the non-spoiler class dominates the dataset with 232,481 instances supported, its strong performance can be mostly ascribed to the class distribution. Considering that no linguistic changes (such text normalization or feature engineering) were used, the model's performance in this area can be regarded as strong. It is crucial to remember, though, that the model's strong recall in the non-spoiler class might potentially be a result of some bias in favor of the majority class.

However, the spoiler class's performance was noticeably worse. While most of the anticipated spoilers were accurate, a significant portion were misclassified, according to the model's 0.64 precision for spoiler predictions. The model's recall for spoilers, which was only 0.34 and meant that it detected fewer than half of the real spoilers, is more worrisome. This implies that a significant percentage of spoilers are not being captured by the model, most likely as a result of the class imbalance as well as the difficulty of identifying spoilers in unprocessed text without feature extraction or linguistic context. The model's performance in identifying spoilers is unsatisfactory, as evidenced by the spoiler class's F1-score of 0.44, which shows the imbalance between precision and recall. It is hardly surprising that the absence of preprocessing has had a detrimental effect on the model's capacity to accurately identify spoilers, considering the intrinsic complexity of spoiler detection, which entails comprehending subtle context and content clues. Furthermore, this problem has probably been made worse by the reduced support for spoilers (60,572 instances) in comparison to non-spoilers, as the model might be overfitting to the majority class.

The model's overall accuracy is 0.73, meaning that 73% of all predictions were accurate. Although this accuracy might appear to be sufficient at first, it's crucial to understand that the model's ability to predict the majority class (non-spoilers) has a significant impact on this statistic. The significant decline in the spoiler class's performance, especially in terms of recall, suggests that accuracy by itself does not give a whole view of the model's efficacy. The model's bias towards the majority class and the imbalance between the classes are reflected in the weighted averages for precision (0.74), recall (0.64), and F1-score (0.67). The disproportionate support of non-spoilers skews these figures, which offer an overall picture of the model's performance across both classes. These findings might have been further influenced by the absence of preprocessing because noisy, raw text data sometimes contains deceptive or irrelevant qualities that might mask important trends, particularly for minority classes like spoilers.

The baseline model reveals text categorization model performance on raw data. The model has great precision and recall in identifying non-spoiler material but low recall in detecting spoilers. The model's accuracy of 0.73 shows its success with the majority class but the difficulties of detecting spoiler content without preprocessing.

5.9 Final Model Performance

In this section, I will present the detailed performance of the model after training across five epochs. The training and validation loss along with precision, recall, F1 score, and accuracy were recorded for each epoch. Table 5.5 shows the results obtained during each training epoch.

Epoch	Training Loss	Validation Loss	Precision	Recall	F1 Score	Accuracy
1	0.3933	0.4045	0.8281	0.8389	0.8127	0.8389
2	0.3700	0.3943	0.8359	0.8467	0.8269	0.8467
3	0.3542	0.3762	0.8390	0.8498	0.8322	0.8498
4	0.3375	0.4058	0.8402	0.8512	0.8357	0.8512
5	0.3176	0.3985	0.8394	0.8507	0.8360	0.8507

Table 5.5: Training and Validation Metrics for Each Epoch

The results in Table 5.5 show that the model's performance improved steadily across the epochs with a gradual decrease in both training and validation losses. The training loss consistently decreased which indicated that the model learned over each epoch. However, the validation loss showed slight fluctuations. I was seen particularly an increase during the fourth epoch, suggesting a possible overfitting issue at that point. Despite this fluctuation, the final epoch saw a slight reduction in validation loss, showing a possible correction (Figure 5.6).

The precision, recall, and F1 score metrics also show an overall positive trend across the epochs. Precision increased from 0.8281 in the first epoch to 0.8394 in the final epoch. Similarly, recall improved from 0.8389 to 0.8507, reflecting the model's enhanced ability to correctly identify true positives over time. The F1 score increased from 0.8127 to 0.8360, indicating that the model achieved a more balanced performance by reducing false positives and false negatives.

The final evaluation results which is presented in Table 5.6, provide additional insights into the model's performance on the test dataset after five epochs.

Metric	Value
Evaluation Loss	0.3985
Evaluation Precision	0.8394
Evaluation Recall	0.8507
Evaluation F1 Score	0.8360
Evaluation Accuracy	0.8507
Evaluation Runtime (s)	1428.1496
Samples per Second	205.198
Steps per Second	12.825

Table 5.6: Evaluation Metrics After Five Epochs

The F1 score, which eventually attained a value of 0.8360, is the primary focus of this investigation because it is the most important factor. When dealing with imbalanced datasets, where one class may predominate, this statistic is very important for understanding the balance that exists between precision and recall. The macro-average F1 score of the model was 0.73, which indicates that there seems to be some difficulties in efficiently managing the minority class. A weighted average F1 score of 0.84, on the

other hand, indicates that the overall balanced performance was achieved. Whereas this indicates that the majority class is given a greater weight.

Class	Precision	Recall	F1 Score	Support
Non-spoiler	0.87	0.96	0.91	232481
Spoiler	0.73	0.44	0.55	60572
Accuracy	0.85			
Macro Avg	0.80	0.70	0.73	293053
Weighted Avg	0.84	0.85	0.84	293053

Table 5.7: Classification Report for Test Set

Table 5.7 reveals that the model performs much better on the majority class (label 0), attaining a high precision of 0.87 and recall of 0.96, which results in an F1 score of 0.91. On the other hand, the precision was lower for the minority class (label 1), coming in at 0.73, and the recall was 0.44, which eventually led to an F1 score of 0.55. According to these findings, the model is capable of accurately recognizing the majority class; however, it struggles to accurately identify the minority class, which is often the case in datasets that are imbalanced.

5.9.1 Visual Analysis

To further understand the performance of the model, I present visualizations that capture the trends in key metrics like training loss, validation loss, precision, recall, and F1 score across epochs. Figure 5.6 provides a combined visualization of the training and validation losses as well as the precision, recall, and F1 score across all epochs.

The left subplot in Figure 5.6 illustrates the trends in training and validation loss across the epochs. The consistent reduction in training loss signifies that the model is acquiring knowledge efficiently. The validation loss shows minor variations, with a rise in the fourth epoch indicating possible overfitting. The decline in the last period suggests a possible correction, indicating that the model improved its generalization following adjustments.

The right subplot in Figure 5.6 illustrates the trends in precision, recall, and F1 score throughout epochs. The F1 score steadily rose, indicating enhancements in the model's equilibrium between precision and recall. The concluding period attains the highest F1 score of 0.8360, indicating robust overall performance.

The Receiver Operating Characteristic (ROC) curve for the optimal model is depicted in Figure 5.7. The ROC curve visually depicts the model's efficacy at various classification thresholds by displaying the True Positive Rate (TPR) against the False Positive Rate (FPR). The curve illustrates the model's efficacy in differentiating between spoiler and non-spoiler situations.

The AUC (Area Under the Curve) score of 0.8327 signifies that the model possesses a robust capacity to differentiate between positive and negative classes. An elevated AUC value indicates that the model effectively balances sensitivity (recall) and specificity, rendering it suitable for applications such as spoiler detection, where the trade-off between false positives and false negatives is essential. The curve nears the upper left corner of the graph, signifying that the model efficiently minimizes false positives while preserving a high true positive rate. Detecting spoilers (positive class) is particularly crucial, as failure to identify them may result in undesirable consequences for users.

These visual patterns offer a thorough insight into the model's learning process and

underscore areas for enhancement, especially for the management of imbalanced classes. Future research may explore resampling strategies or utilize cost-sensitive learning methods to improve the model's accuracy in classifying the minority class.

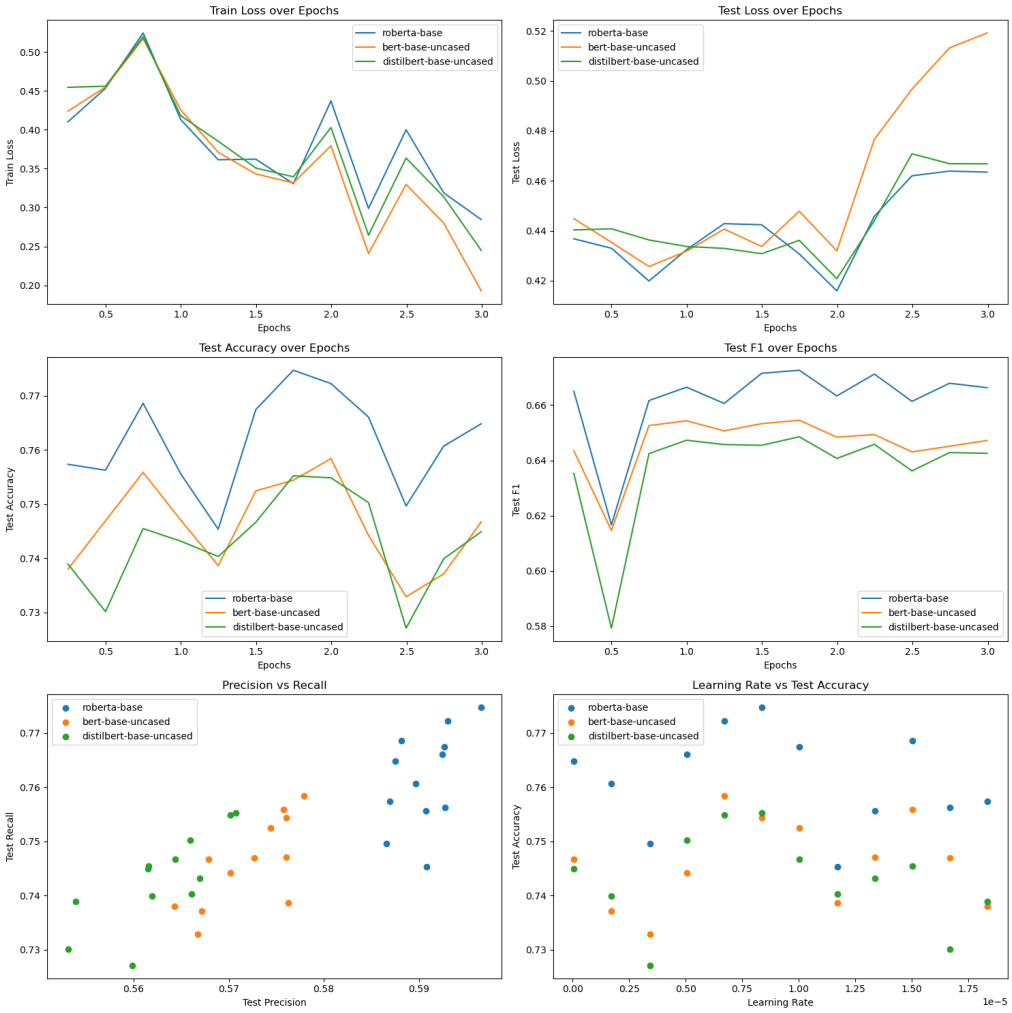

Figure 5.3: Model performance comparison: Review Text, Review Sentiment, and Review Length Combined

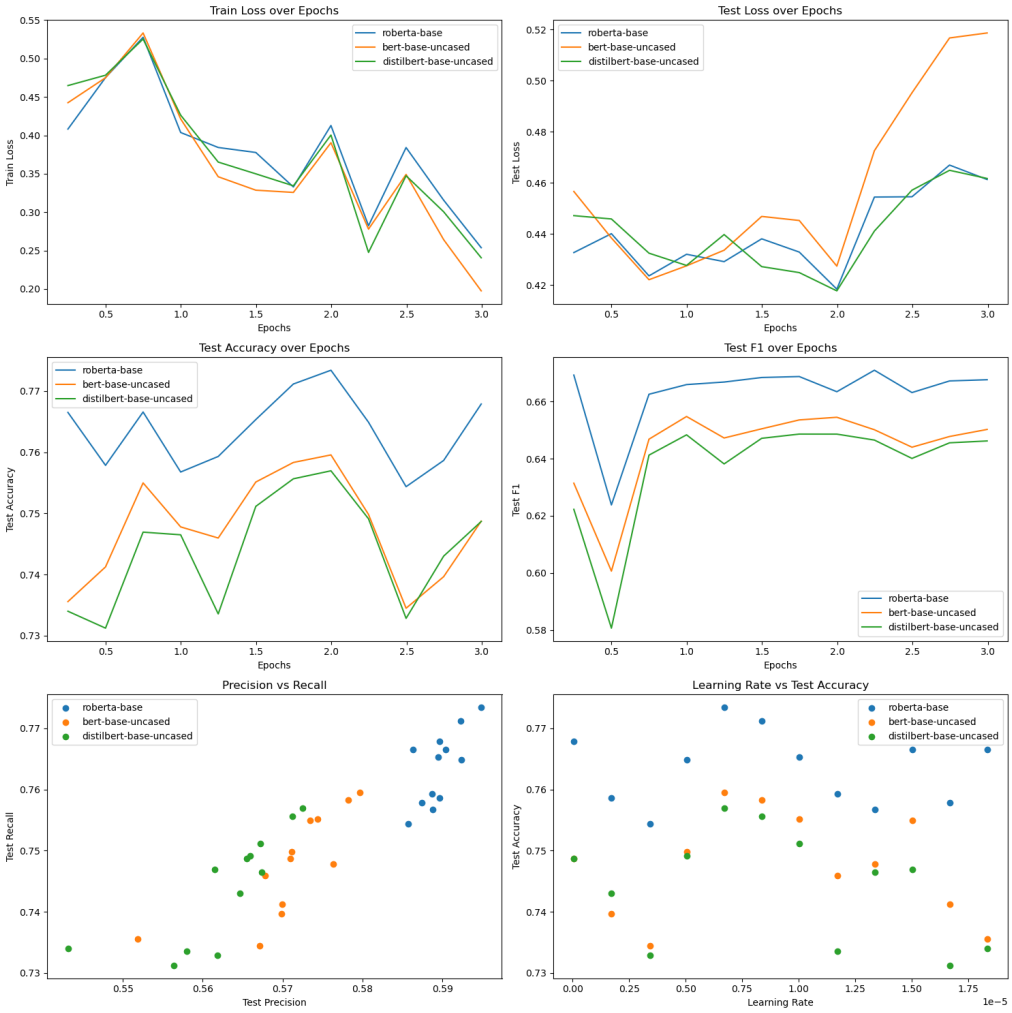

Figure 5.4: Model performance comparison: Review Text, Review Sentiment, and Review Summary Combined

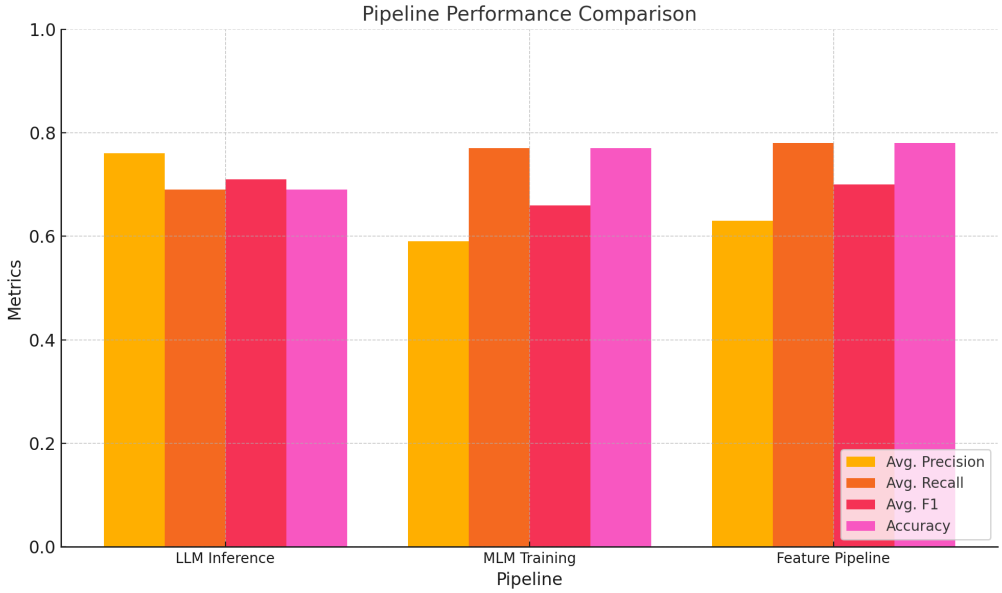

Figure 5.5: Comparative analysis of pipeline performance metrics.

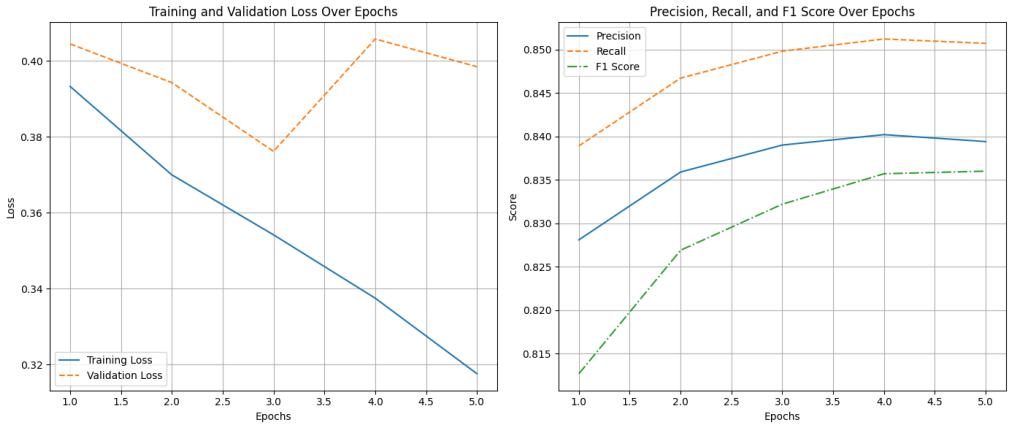

Figure 5.6: Training and Validation Loss (left) and Precision, Recall, and F1 Score (right) Over Epochs

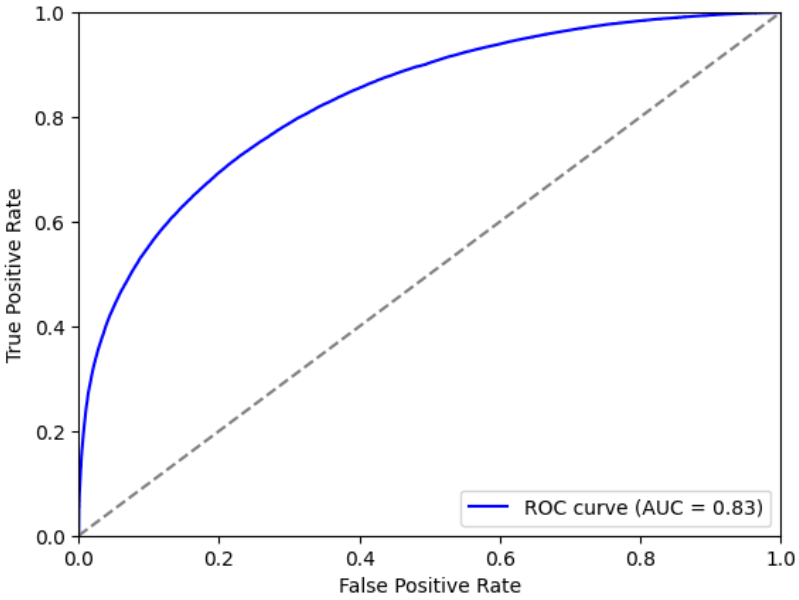

Figure 5.7: ROC Curve for Best Model with AUC of 0.8327

Chapter 6
Discussion

6.1 Spoiler Detection Performance Analysis

Are specific films or television shows difficult to spoil?

In this section, I will evaluate the efficacy of spoiler detection in film reviews by analyzing critical metrics across a varied selection of movies. The data encompasses spoiler ratios, review counts, accuracy, and F1 scores, providing insights into the detection system's overall performance and the variability in spoilability across various films.

The test dataset reveals an average spoiler ratio of 0.21, suggesting that around 21% of the content in movie reviews includes spoilers. Films with a greater number of reviews generally demonstrate reduced spoiler ratios. This indicates that films appealing to a wide audience may feature fewer spoilers in reviews, likely because of the general public's familiarity with their narratives, which diminishes the necessity to address spoilable elements.

The model demonstrates high accuracy, with the majority of films attaining scores exceeding 0.85. This trend demonstrates the model's reliable ability to differentiate between spoiler and non-spoiler content across diverse movie genres. Furthermore, the F1 score, which balances precision and recall alongside accuracy, demonstrates comparably high values, reflecting effective detection of spoilers within the sample.

An analysis of spoiler ratios indicates specific patterns related to spoilability. Films featuring distinctive plot twists, substantial character developments, or remarkable emotional trajectories typically exhibit elevated spoiler ratios. Films like *The Shawshank Redemption* and *The Godfather*, characterized by significant narrative developments and iconic scenes, demonstrate elevated spoiler ratios. In general, films recognized for their iconic or pivotal plot moments often include a greater amount of spoilable content in reviews.

In contrast, films featuring formulaic narratives or familiar tropes seem to be less susceptible to spoilers. Films such as *Captain Marvel* and *Wonder Woman 1984*, which prioritize

visual aesthetics and action sequences over complex narratives, exhibit reduced spoiler ratios. This suggests that these films are less susceptible to spoilage, as their primary allure is not predominantly dependent on plot twists but rather on the engagement with the genre's characteristic features.

Figure 6.1 displays four essential metrics (Review Count, Spoiler Ratio, Accuracy, and F1 Score) in relation to the movie index. This visualization facilitates the evaluation of distribution and performance among films.

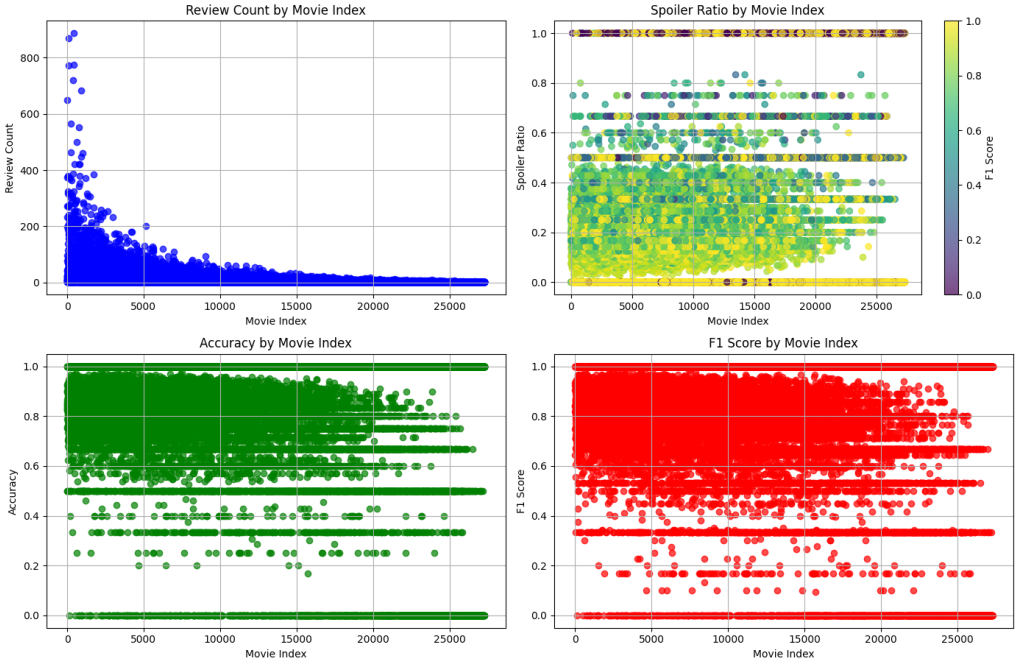

Figure 6.1: Review Metrics by Movie Index

- **Top-left (Review Count by Movie Index)**: This plot illustrates the distribution of review counts across various movies. Films with lower indexes, typically more popular, exhibit significantly higher review counts, which decline sharply as the movie index rises.
- **Top-right (Spoiler Ratio by Movie Index)**: This scatter plot illustrates the relationship between spoiler ratios and F1 scores, represented through a color gradient. Films exhibiting elevated spoiler ratios often congregate in particular areas, suggesting that certain titles are more susceptible to spoiler-related content. The gradient of the F1 score indicates improved performance (represented by brighter colors) at moderate spoiler ratios.
- **Bottom-left (Accuracy by Movie Index)**: The model's accuracy for each film is presented here, demonstrating a generally high performance across the dataset. Intermittent declines in accuracy indicate that specific films pose difficulties, probably attributable to distinctive narratives or review methodologies.
- **Bottom-right (F1 Score by Movie Index)**: F1 scores, which indicate the balance between precision and recall, are consistently high, clustering around 0.8 or

higher. Intermittent declines suggest films in which spoilers are more challenging to identify, potentially owing to nuanced or indirect phrasing in critiques.

The detection system demonstrates overall effectiveness, evidenced by high accuracy and F1 scores indicative of strong spoiler identification capabilities. The results indicates that films characterized by intricate narratives or significant plot twists are more susceptible to spoiling, whereas action-oriented or well-known films contain fewer elements that can be spoiled. This analysis highlights the complex dynamics of spoiler detection, indicating that a movie's spoilability is shaped by its narrative structure and the viewer's familiarity with the content.

6.2 Sentiment Influence on Model Decision-Making

In this section, I assessed the model's performance across positive, neutral, and negative sentiment classes to determine the influence of different sentiments on its decision-making. The classification metrics of precision, recall, and F1 score provide insights into the model's sensitivity to variations in sentiment.

The classification report for each sentiment is presented in Table 6.1. The model demonstrates high precision and recall for non-spoiler reviews (labeled as 0) across all sentiment categories, whereas its performance for spoiler reviews (labeled as 1) is comparatively lower, particularly in instances of positive and negative sentiments. This imbalance indicates that sentiment may influence the model's capacity to accurately classify spoilers, especially when reviews express strong opinions or emotions. However, this particular assumption can be disregarded due to the significant imbalance in the training dataset.

The model demonstrates an overall accuracy of 85.77% for positive sentiment, 80.59% for negative sentiment, and 84.93% for neutral sentiment. The F1 scores exhibit a comparable trend, with the peak score for positive sentiment at 0.85, while neutral and negative sentiments yield marginally lower scores. The results suggest that the model demonstrates consistent performance across various sentiments; however, its classification of spoilers is less effective in the presence of strong positive or negative sentiment in reviews. This observation indicates that sentiment, particularly polarized sentiment, may influence the clarity of spoiler cues, by resulting in reduced detection rates.

Table 6.1: Classification Performance by Sentiment

Sentiment	**Class**	**Precision**	**Recall**	**F1-Score**	**Accuracy**	**Support**
Positive	Non-spoiler	0.89	0.94	0.92	0.8577	133,581
	Spoiler	0.59	0.44	0.51		26,484
Negative	Non-spoiler	0.86	0.88	0.87	0.8059	94,940
	Spoiler	0.63	0.58	0.61		32,982
Neutral	Non-spoiler	0.88	0.94	0.91	0.8493	3,240
	Spoiler	0.69	0.53	0.60		874

The overall performance metrics for each sentiment class are summarized in Table 6.2. The model exhibits superior performance for positive and neutral sentiments, demonstrating higher accuracy and F1 scores in comparison to negative sentiment. This pattern indicates a potential area for enhancement in managing spoilers in reviews that convey negative sentiment, as spoilers may be integrated within strong opinions or critical language.

Table 6.2: Overall Performance by Sentiment

Sentiment	Accuracy	F1 Score
Positive	0.8577	0.8489
Negative	0.8059	0.8032
Neutral	0.8493	0.8415

In conclusion, the model demonstrates robust performance in spoiler detection across various sentiments, with positive and neutral sentiments producing marginally superior results compared to negative sentiment. The diminished performance on negative reviews indicates that sentiment may affect the model's decision-making, especially regarding polarized language. Improving the model's sensitivity to spoiler cues in highly opinionated reviews may enhance its overall effectiveness in spoiler classification.

6.3 Model Decision Making: In Depth Spoiler In Reviews

In this section, I will be discussing how the model is making decisions while leveraging 6. On top of that, I will provide a comparision against GPT-4o decision making process on review segments.

Primarily, these reviews were extracted using the chat interface of GPT-4o. One question might arise: Why not use actual reviews and parse them using the LLM? However, while going forward with that approach, it was observed that GPT-4o sometimes needs to catch up on more minor potential spoilers. To mitigate this issue and have more control, I have parsed three reviews for the three most popular movies. Each line was tagged, and potential spoiler segments corresponding to each line were extracted for the GPT-4o baseline. These were then compared with my model's decision-making.

While extracting the spoiler detection models' decision-making using the chunking methods, it was observed that a chunk size of 4 could extract the most lines that contain spoilers. Through investigation, it was observed that while the chunk size is 2 or 3, the model could get the context properly. With larger chunk sizes 5 and 6, the model performed similarly. However, the problem with larger chunk sizes is that it would be hard for humans to determine the crucial segment of the spoiler. Moreover, it was observed that usually, a spoiler lies within 3 to 5 words.

6.3.1 Spoiler Detection Analysis for Movie Review: "Avengers: Endgame (2019)"

This section presents a comprehensive examination of spoiler detection for a review of the film "Avengers: Endgame". The review includes essential plot aspects that contain big spoilers on the deaths of important characters and pivotal events in the narrative.

Review Overview The analysis highlights pivotal events from "Avengers: Endgame," including Tony Stark's sacrifice, Black Widow's demise, Hulk's reversal of "The Snap," and Captain America's ultimate choice to remain in the past. These occurrences render the review significantly laden with spoilers, hence providing an optimal scenario to evaluate the spoiler detection technology. The evaluation is as follows -

> ”Avengers: Endgame” delivers a powerful conclusion to the Infinity Saga. The film explores the aftermath of ”The Snap,” with Tony Stark’s emotional sacrifice being the highlight. Watching Black Widow sacrifice herself for the Soul Stone on Vormir is heartbreaking, and the return of half the universe through Hulk’s snap is a thrilling payoff. Captain America’s final scene, where he chooses to stay in the past and live his life with Peggy Carter, adds an emotional and fitting closure to his arc.

The following sentences from the review contain significant spoilers from the movie:

- `Tony Stark's emotional sacrifice being the highlight.`
- `Watching Black Widow sacrifice herself for the Soul Stone on Vormir is heartbreaking.`
- `The return of half the universe through Hulk's snap is a thrilling payoff.`
- `Captain America's final scene, where he chooses to stay in the past and live his life with Peggy Carter, adds an emotional and fitting closure to his arc.`

Spoiler Detection Results

By utilizing the chunking method I was able to gather chunks which potentially got spoilers. Table 6.3 lists the chunks that were marked as containing spoilers, indicating their contribution to the overall spoiler prediction.

Chunk	**Impact Value**
”The Snap, with Tony”	1.4785
”Watching Black Widow sacrifice”	1.7022
”Black Widow sacrifice herself”	0.7424
”Widow sacrifice herself for”	1.2532
”sacrifice herself for the”	1.0317
”Stone on Vormir is”	1.2548
”universe through Hulk’s snap”	1.7021
”through Hulk’s snap is”	1.5837
”Hulk’s snap is a”	1.1600
”final scene, where he”	1.2230

Table 6.3: Chunks Identified as Spoilers by the System (Chunk Size = 4)

The system successfully identified several key spoilers:

- `"The Snap, with Tony"`: This chunk indicates Tony Stark’s involvement in reversing the effects of ”The Snap” which is a major plot point.
- `"Watching Black Widow sacrifice"`, `"Black Widow sacrifice herself"`, and `"Widow sacrifice herself for"`: These chunks refer to Black Widow’s death, which is a critical moment in the movie.
- `"Stone on Vormir is"`: This chunk relates to the location where Black Widow sacrifices herself for the Soul Stone and ended up contributing significantly to the spoiler content.

- `"universe through Hulk's snap"`, `"through Hulk's snap is"`, and `"Hulk's snap is a"`: These segments refer to Hulk's snap, which reverses "The Snap" and brings back the lost half of the universe.
- `"final scene, where he"`: This describes Captain America's decision to stay in the past, which is an emotional conclusion to his character arc.

These segments were correctly classified as spoilers and provide essential plot information similar to GPT-4o.

Missed Spoiler Segments

There were still some significant segments that were not classified correctly despite the model's ability to identify several key spoilers. The missed spoiler segments include:

- `"Tony Stark's emotional sacrifice being the highlight"`: This line describes Tony Stark's ultimate sacrifice, which is a crucial turning point in the movie.
- `"Captain America's final scene, where he chooses to stay in the past and live his life with Peggy Carter"`: The model did not fully detect this spoiler. Even though parts of it were identified. This is an important moment that brings closure to Captain America's storyline.
- `"The portals opening to bring back every snapped character"`: This describes the epic final battle, where all previously snapped characters return.

Visual Representation of Chunk Impact

Figure 6.2 provides a visual representation of the impact of individual chunks and the rest of the text on spoiler classification. Red bars represent spoiler chunks, while green bars represent non-spoiler chunks.

The visualization shows that the system assigned high impact values to the chunks classified as spoilers. However, the lower impact scores for important spoiler segment that the model did not perceive their importance sufficiently.

Discussion of System Performance

The use of a chunk size of 4 words shows improvements in capturing specific spoiler segments compared to smaller chunk sizes. The model was able to identify several key plot points involving significant character actions and twists.

However, the model still failed to capture some broader spoiler contexts. It struggled with segments that required an understanding of the entire event. These misses indicate that the model might still lack the ability to fully comprehend complex narrative structures when the spoiler spans in multiple connected events.

6.3.2 Spoiler Detection Analysis for Movie Review: "The Empire Strikes Back (1980)"

In this section, I will analyze the spoiler detection performance for a review of "The Empire Strikes Back" which was released on 198. Overall, the review contains several

major plot twists and reveals the iconic revelation of Darth Vader's true identity and other significant moments in the Star Wars saga.

Review Overview

The review contains critical moments from "The Empire Strikes Back". Specially, focusing on Luke Skywalker's training under Yoda, the revelation of Darth Vader's true identity, and Han Solo's fate. These scenes are some of the most famous and heavily spoiler-laden moments in the Star Wars saga, making the review an ideal case for spoiler detection testing. The review is as follows -

> "The Empire Strikes Back" is arguably one of the greatest sequels ever made, deepening the Star Wars saga with shocking revelations. Luke Skywalker's intense training on Dagobah under Yoda sets up his eventual confrontation with Darth Vader. The twist where Vader reveals, "No, I am your father," is perhaps the most iconic moment in cinematic history. Han Solo being frozen in carbonite is a gut-wrenching moment, especially as Princess Leia confesses her love for him just before he's taken away.

The following sentences from the review contain significant spoilers for the movie:

- `Luke Skywalker's intense training on Dagobah under Yoda sets up his eventual confrontation with Darth Vader.`
- `The twist where Vader reveals, "No, I am your father," is perhaps the most iconic moment in cinematic history.`
- `Han Solo being frozen in carbonite is a gut-wrenching moment, especially as Princess Leia confesses her love for him just before he's taken away.`

Spoiler Detection Results

We have broken down the spoiler element of the review using the chunking mechanism in Table 6.4 alongside their impact on decision making.

Chunk	**Impact Value**
"Darth Vader. The twist"	0.7861
"The twist where Vader"	1.5301
"twist where Vader reveals,"	0.7368
"where Vader reveals, No,"	1.4525
"Han Solo being frozen"	0.2795
"Solo being frozen in"	1.5030
"Leia confesses her love"	1.5441
"before he's taken away."	1.4523
"he's taken away. The"	1.0330

Table 6.4: Chunks Identified as Spoilers by the System (Chunk Size = 4)

The system correctly identified several key spoiler moments, including:

- `"Darth Vader. The twist"` and `"The twist where Vader reveals"`: These chunks presented the iconic scene where Darth Vader reveals that he is Luke's father. This moment is arguably the biggest spoiler in the movie.
- `"Han Solo being frozen"` and `"Solo being frozen in"`: These chunks highlight Han Solo being frozen in carbonite, is a major cliffhanger ending in the movie.
- `"Leia confesses her love"`: This refers to Princess Leia's confession of love to Han Solo.
- `"before he's taken away"` and `"he's taken away. The"`: These segments describe Han Solo being taken away after being frozen in carbonite.

Missed Spoiler Segments

Despite the system's success in identifying some major spoilers, several key segments were missed.The missed spoiler segments include:

- `"Luke Skywalker's intense training on Dagobah under Yoda"`: This describes Luke's training under Yoda, which plays a significant role in his journey.
- `"Yoda sets up his eventual confrontation with Darth Vader"`: The model failed to recognize this critical training moment that leads up to Luke's showdown with Darth Vader.
- `"I am your father," is perhaps the most iconic moment in cinematic history`: The model failed to fully capture the iconic line in its entirety while parts of the reveal were correctly classified, resulting in a partially missed context.
- `"The Rebel forces suffer massive setbacks, leaving audiences with a tense cliffhanger ending"`: This describes the Rebel forces' struggles and the cliffhanger ending that sets up the high stakes for the following movie.

Visual Representation of Chunk Impact

Figure 6.3 illustrates the impact of each identified chunk and the rest of the text on the overall spoiler classification.

Discussion of System Performance

The system detected several significant spoiler moments and assigned relatively high impact values to these chunks, indicating their importance in the overall spoiler classification.

However, the model still missed several key segments that were vital to understanding the full scope of the spoilers in the review. This includes the entire sequence of Luke's training under Yoda and the explicit context of the Rebel forces' setbacks. The difficulty in identifying these spoilers may arise from the model's limited understanding of the connections between different parts of the plot. Especially when a spoiler event extends beyond the immediate 4-word chunk.

6.3.3 Spoiler Detection Analysis for Movie Review: "The Sixth Sense (1999)"

The review for this section contains major spoilers which contains famous twist revealing the true nature of Dr. Malcolm Crowe and the abilities of the young protagonist named Cole.

Review Overview

The review covers crucial plot points from "The Sixth Sense," including Dr. Malcolm Crowe's relationship with young Cole, and the shocking twist where Crowe realizes he has been dead the entire time. The review is as follows -

> "The Sixth Sense" is a chilling psychological thriller that masterfully plays with perception. Bruce Willis's character, Dr. Malcolm Crowe, seems to help young Cole, who reveals he "sees dead people." The true twist is revealed at the end when Crowe realizes he has been dead the entire time. The chilling moment when Cole reveals his ability in the car with his mother, and the subsequent scenes where Crowe accepts his fate, make this film unforgettable and tragic.

The following sentences from the review contain significant spoilers for the movie:

- `Bruce Willis's character, Dr. Malcolm Crowe, seems to help young Cole, who reveals he "sees dead people."`
- `The true twist is revealed at the end when Crowe realizes he has been dead the entire time.`
- `The chilling moment when Cole reveals his ability in the car with his mother, and the subsequent scenes where Crowe accepts his fate, make this film unforgettable and tragic.`

Spoiler Detection Results

Table 6.5 presents the chunks identified as spoilers along with their respective impact values.

Chunk	Impact Value
"Cole, who reveals he"	0.4027
"the end when Crowe"	1.0452
"end when Crowe realizes"	1.3152
"when Crowe realizes he"	0.4177
"Crowe realizes he has"	0.5947
"when Cole reveals his"	0.4341

Table 6.5: Chunks Identified as Spoilers by the System (Chunk Size = 4)

The spoiler detection system correctly identified some of the crucial plot points:

- `"Cole, who reveals he"`: This refers to Cole's statement about his ability to "see dead people," which is one of the key plot twists in the movie.
- `"the end when Crowe"` and `"end when Crowe realizes"`: These chunks refer to the moment when Dr. Crowe realizes the truth about himself, which is the biggest twist in the movie.
- `"when Crowe realizes he"` and `"Crowe realizes he has"`: These segments continue to describe Dr. Crowe's realization about his own condition, which is essential to the spoiler content.
- `"when Cole reveals his"`: This chunk is part of Cole's conversation revealing his ability, which sets up the suspense for the twist.

Missed Spoiler Segments

Despite successfully identifying some critical spoiler segments, several key spoilers were missed by the system. The missed spoiler segments include:

- `"he s̈ees dead people.The"`: This is perhaps the most famous line in the movie, but it was partially missed by the system. The model failed to fully recognize the impact of this line.
- `"Crowe realizes he has been dead"` and `"has been dead the entire time"`: These segments explicitly reveal the movie's major twist. Despite correctly identifying parts of Crowe's realization, the model failed to capture the entire context.
- `"subsequent scenes where Crowe accepts his fate"`: This segment shows Crowe coming to terms with his fate, a key emotional moment that was not classified correctly.
- `"Crowe accepts his fate, make this film unforgettable and tragic"`: This describes how Dr. Crowe accepts the truth about his death, contributing to the emotional weight and tragic nature of the film. The failure to classify this as a spoiler highlights the model's difficulty with emotional conclusions.

Visual Representation of Chunk Impact

Figure 6.4 presents a visual representation of the impact of individual chunks and the rest of the text on the overall spoiler classification. Red bars represent chunks classified as spoilers, while green bars indicate non-spoilers.

The picture demonstrates that several pivotal spoiler phrases, particularly concerning Dr. Crowe's epiphany, attained elevated effect values when accurately identified. Nevertheless, certain critical areas that ought to have been recognized as spoilers were undervalued, resulting in overlooked detections.

Discussion of System Performance

A 4-word chunk size allowed the algorithm to collect spoilers like Cole's admission of seeing dead people and Dr. Crowe's discovery that he is dead. Short phrases referencing crucial moments dominated the selected sections.

The explicit confirmation of Dr. Crowe's death and Cole's legendary line's full phrase still challenged the model. These misses show that the model can recognize direct and brief spoilers but not longer narrative elements that require additional information.

The model undervalued overlooked pieces like "he has been dead the entire time," suggesting a lack of narrative structure knowledge.

6.3.4 Conclusion

The spoiler detection program correctly identified numerous significant spoiler chunks in "Avengers: Endgame," "The Empire Strikes Back," and "The Sixth Sense," but it missed key sections in each review. These skipped pieces generally related to the story or contained emotionally powerful situations spanning across numerous words.

In "The Sixth Sense," "I see dead people" was partially recognized, but Dr. Malcolm Crowe's death was not. In "The Empire Strikes Back," the model underestimated Luke Skywalker's training under Yoda and lost the emotional impact of Han Solo's carbonite imprisonment. The model missed elements of Hulk's "The Snap." moment in "Avengers: Endgame," but it caught Tony Stark's sacrifice.

These results show that the chunk-based paradigm works well for shorter, straight spoilers but difficulties with longer, context-driven reveals and emotionally complex sequences. Several potential additions could increase model performance:

- **Contextual Understanding:** Understanding long-range dependencies across sentences might improve the model. Latest, attention-based techniques can better capture narrative flow and textual links.
- **Emotional Context Detection:** Many overlooked spoilers were emotive scenes like character deaths or sacrifices. Superior sentiment analysis and expanded emotion explainability features may improve model accuracy.
- **Broader Context Training:** Training the algorithm on a dataset with more sophisticated and interrelated narrative structures may help it recognize spoilers throughout longer sequences rather than just small bits.
- **Hierarchical Approaches:** A hierarchical approach may identify larger chunks of a review as spoiler zones, then perform chunk-based analysis within them. This could detect spoilers that use information from numerous phrases or paragraphs.
- **Post-processing Strategies:** Post-processing to join relevant chunks and re-evaluate the context could fix fragmented spoiler identification. This helps when key plot twists are split across numerous sections.

The model is promising at recognizing direct and immediate spoilers, but it needs to improve at handling context-dependent spoilers and emotional moments. Addressing these restrictions improves the model's spoiler recognition and might yield more accurate and complete results.

6.4 Comparison of Best Model and LLM Inference

I examined the relative performance characteristics of two approaches: the optimized model and the LLM Inference pipeline. Testing utilized a substantial dataset of 55,000 samples,

revealing significant variations across multiple performance dimensions. Figure 6.5 illustrates these comparative results.

Metric Analysis My experimental results demonstrated marked differences between the two approaches. The optimized model achieved notably stronger results, with an F1 score reaching 83%, significantly exceeding the LLM Inference score of 72%. Precision measurements showed similar disparities, with my model achieving 82% versus 75% for LLM Inference. The recall rate displayed an even wider gap, where my model reached 84% compared to 71% for LLM Inference. These differences extended to overall accuracy, with the best model achieving 84% versus 72% for the LLM approach. Such consistent performance advantages suggest fundamental architectural benefits in the optimized implementation.

Processing Efficiency Runtime analysis revealed substantial differences in computational efficiency. The best model completed the full test suite processing in 13 minutes, whereas the LLM Inference pipeline required 233.33 minutes for identical data. This stark contrast in processing speed highlights significant implications for practical deployments, particularly in resource-constrained environments or applications requiring rapid response times.

Implementation Considerations While the LLM Inference pipeline offers advantages in result interpretability through its structured analysis approach, my experiments indicate notable limitations in its practical application. The combination of lower accuracy metrics and extended processing requirements presents significant challenges for large-scale deployment scenarios. Conversely, the optimized model demonstrates consistent performance across all evaluation criteria while maintaining exceptional processing efficiency.

Research Implications My findings suggest clear advantages for the optimized model in practical applications. Its superior performance across precision, recall, and overall accuracy, combined with remarkable processing efficiency, establishes it as the more viable solution for production environments. Although the LLM Inference approach offers certain analytical benefits, its performance characteristics and computational demands significantly limit its practical utility in most deployment scenarios.

6.5 Model Comparison and Performance Evaluation

In this section, I will compare the performance of my optimal model with various state-of-the-art models, including MVSD (Multi-View Spoiler Detection), BERT, RoBERTa, GCN, R-GCN, and SpoilerNet. Furthermore, I offer a comparison with the baseline model trained on unprocessed text data to demonstrate how data alone can play a huge role in performance.

My dataset has 3 million samples, surpassing the 1.86 million samples utilized in SpoilerNet and MVSD. The expanded dataset offers a more varied and thorough assessment of model efficacy, possibly enhancing generalization and detection performance.

6.5.1 Comparison with State-of-the-Art Models

Table 6.6 presents a comparison of the performance of the optimal model, the baseline model, and many state-of-the-art models, including MVSD and SpoilerNet. MVSD is a graph-based framework that creates heterogeneous information networks (HINs) to represent multi-view data, integrating movie-review subgraphs, user-review subgraphs, and knowledge subgraphs. Although MVSD attained a commendable accuracy of 86.37%, its F1-score of 69.22% suggests difficulties in addressing class imbalance, especially in the detection of spoilers.

The baseline model, trained on unprocessed text input, attained an accuracy of 0.73 and a macro F1-score of 0.67. The metrics are considerably inferior to those of the optimal model, underscoring the advantages of model optimization and preprocessing approaches utilized in the final model.

Model	**F1 Score**	**AUC**	**Accuracy**
BERT (Devlin et al., 2019)	46.14	64.82	79.96
RoBERTa (Liu et al., 2019)	47.72	65.55	80.16
BART (Lewis et al., 2020)	48.18	65.79	80.14
SpoilerNet (Wan et al., 2019)	62.86	74.62	83.23
MVSD (Xu et al., 2023)	69.22	78.26	86.37
MMoE (Zeng et al., 2024)	75.04	82.23	**88.58**
Baseline Model (BERT)	44.00	70.64	73.00
Best Model (RoBERTa w/ Sentiment)	**84.00**	**83.27**	85.00

Table 6.6: Comparison of My Models with State-of-the-Art Models Including MVSD and SpoilerNet

6.5.2 Detailed Comparison with MVSD

The MVSD model utilizes a multi-view architecture that amalgamates diverse information regarding films, users, and external knowledge. MVSD's capacity to model the interaction among subgraphs via hierarchical attention renders it an effective approach for spoiler detection. Nonetheless, despite MVSD's commendable accuracy, its comparatively lower F1-score (69.22%) indicates potential deficiencies in managing imbalanced classes comparable to my model.

My best model's exceptional F1-score of 84.00% and elevated AUC of 83.27% demonstrate its remarkable capability in managing the spoiler detection task, particularly in balancing precision and recall. The extensive dataset of 3 million samples undoubtedly enhanced the model's generalization performance.

My baseline model exhibits competitive performance relative to cutting-edge approaches like MVSD and SpoilerNet. Although MVSD attains more accuracy, my model's enhanced F1-score and AUC demonstrate a more effective capacity to equilibrate the trade-offs between precision and recall. Which is essential for spoiler detection tasks. Furthermore, SpoilerNet, although designed for spoiler detection for books, demonstrates inferior overall performance metrics. Presumably due to its smaller dataset and restricted incorporation of external knowledge relative to MVSD.

6.5.3 Conclusion

The assessment indicates that the superior model exhibits robust performance relative to cutting-edge techniques. Although MVSD exhibits great accuracy, the superior F1-score and AUC score of my model illustrate its efficacy in addressing class imbalance and identifying spoilers. The extensive dataset (3 million samples) utilized in my assessment certainly facilitated these enhancements, offering a more comprehensive foundation for model training and evaluation. The baseline model, however beneficial for preliminary benchmarking, was evidently surpassed by both the final model and advanced techniques like MVSD and SpoilerNet.

6.6 Future Work

This study demonstrates the effectiveness of the spoiler detection model across multiple contexts. Nonetheless, several opportunities for additional enhancement persist. Future research may investigate the incorporation of advanced machine learning techniques alongside novel dataset management methods to enhance the model's accuracy, efficiency, and adaptability.

A promising approach involves utilizing advanced large language models (LLMs) like GPT-4o for the purpose of tagging and extracting spoiler elements in reviews. Employing a high-performing model for pre-tagging spoilers may produce refined labels that effectively encapsulate the complexities of spoiler detection. Fine-tuning the model using tags generated by GPT may improve its capacity to detect spoilers with increased contextual awareness. This approach presents challenges, mainly due to the significant costs involved in tagging extensive datasets with advanced LLMs. Furthermore, dependence on external LLMs may lead to biases, given that these models were not explicitly trained for spoiler detection, which could compromise tagging accuracy.

Another area for investigation involves extracting embeddings from recently developed models designed for nuanced text representation. Training the spoiler detection model on these embeddings enables the capture of more complex linguistic features present in spoiler content. Embedding models may not align precisely with the requirements of spoiler detection, as their primary objective is to enhance general language understanding rather than to address specific narrative elements. Furthermore, embedding-based fine-tuning requires significant computational resources, potentially restricting its use with larger datasets.

Class imbalance is a significant issue to consider, as the dataset exhibits a skew between spoiler and non-spoiler classes. Current models may exhibit suboptimal performance on underrepresented classes, which adversely impacts recall for spoilers specifically. Utilizing methods such as class reweighting, oversampling of minority classes, or implementing synthetic data augmentation may address this imbalance and enhance the model's ability to detect spoilers across all categories.

An experimental approach may involve the development of a hybrid model that combines classical machine learning with deep learning techniques. Combining transformer-based contextual embeddings with rule-based systems may enable the model to capture explicit patterns in conjunction with deeper contextual cues, leading to a more comprehensive system adept at managing various types of spoilers. This hybrid system, by integrating interpretability with the advanced processing capabilities of modern transformers, has the potential to improve both accuracy and explainability in spoiler classification.

Furthermore, enhancing the model's robustness through training on varied cross-genre datasets may augment its capacity to generalize to unfamiliar genres and narrative structures. Spoiler detection systems incorporating cross-genre data may exhibit reduced dependence on domain-specific patterns, thereby enhancing their adaptability to changing content types and review styles. This approach enables the fine-tuning of specific genre sub-models, resulting in a modular system that can adapt to differing spoiler expectations across genres. However, a particular show can belong to several genres, which is a challenge while going forward with this.

Future research could incorporate human-tagged data derived from survey responses to enhance the understanding of the subjectivity associated with spoilers. A survey plan has been developed to assess human perceptions of spoilers (Appendix E), enabling participants to pinpoint lines they consider spoilers in diverse reviews. The integration of human-identified spoiler tags may enhance the model by enabling it to learn from a variety of human perspectives. The human-in-the-loop methodology would account for the variability in perceptions of spoilage and enhance the diversity of the model's training data. This approach enhances the system's capacity to mirror real-world spoiler sensitivities and subjective nuances by aligning model outputs with human judgment.

Given the subjectivity of spoiler identification, subsequent versions of the model could improve by utilizing architectures that integrate more sophisticated attention mechanisms to detect nuanced contextual changes. Training the model to prioritize phrases with specific attention weights indicative of spoiler likelihood, derived from human-tagged data may enhance its ability to differentiate between plot development and general context. This approach would advance the model's capability for human-like spoiler detection by considering both content and the implicit cues and context that render a sentence a potential spoiler.

In conclusion, while the model currently exhibits diversity, additional enhancements could aim at improving interpretability and tailoring to user-specific needs. Creating adaptive models that modify sensitivity to spoilers according to user preferences may enhance the personalization of the spoiler-filtering experience. In practical applications, allowing users to adjust spoiler sensitivity thresholds may enhance spoiler detection according to personal tolerances, thereby improving the system's utility.

Future improvements should concentrate on utilizing advanced tagging via LLMs, investigating novel embedding models, tackling class imbalance, integrating hybrid methods, and employing human-tagged data to reflect the diversity in human interpretations of spoilers. Enhancing the training dataset through diverse genres and user-driven customization would improve the model's adaptability and usability, thereby advancing automated spoiler detection.

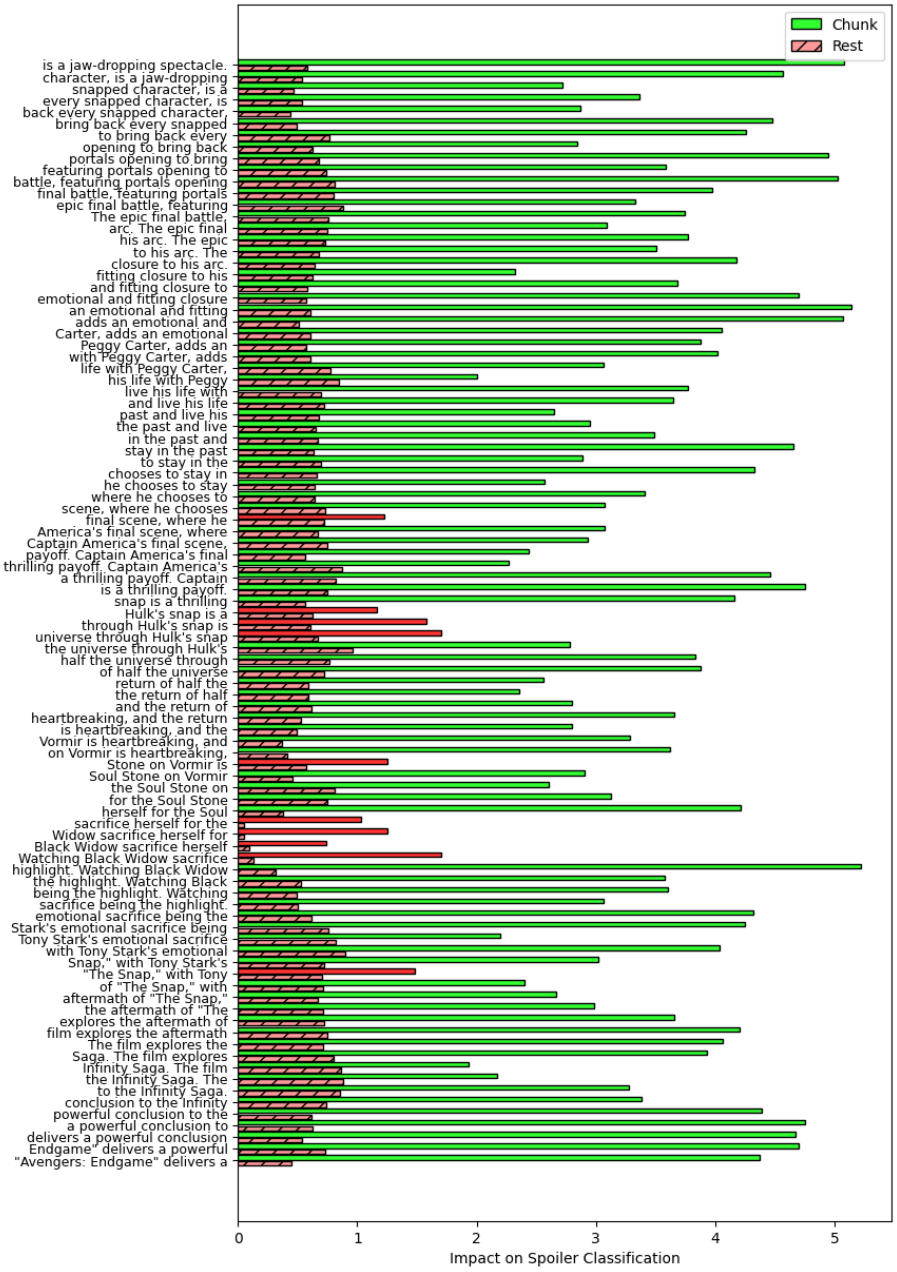

Figure 6.2: Impact of Chunks and Rest of Text on Spoiler Classification for "Avengers: Endgame" (Chunk Size = 4)

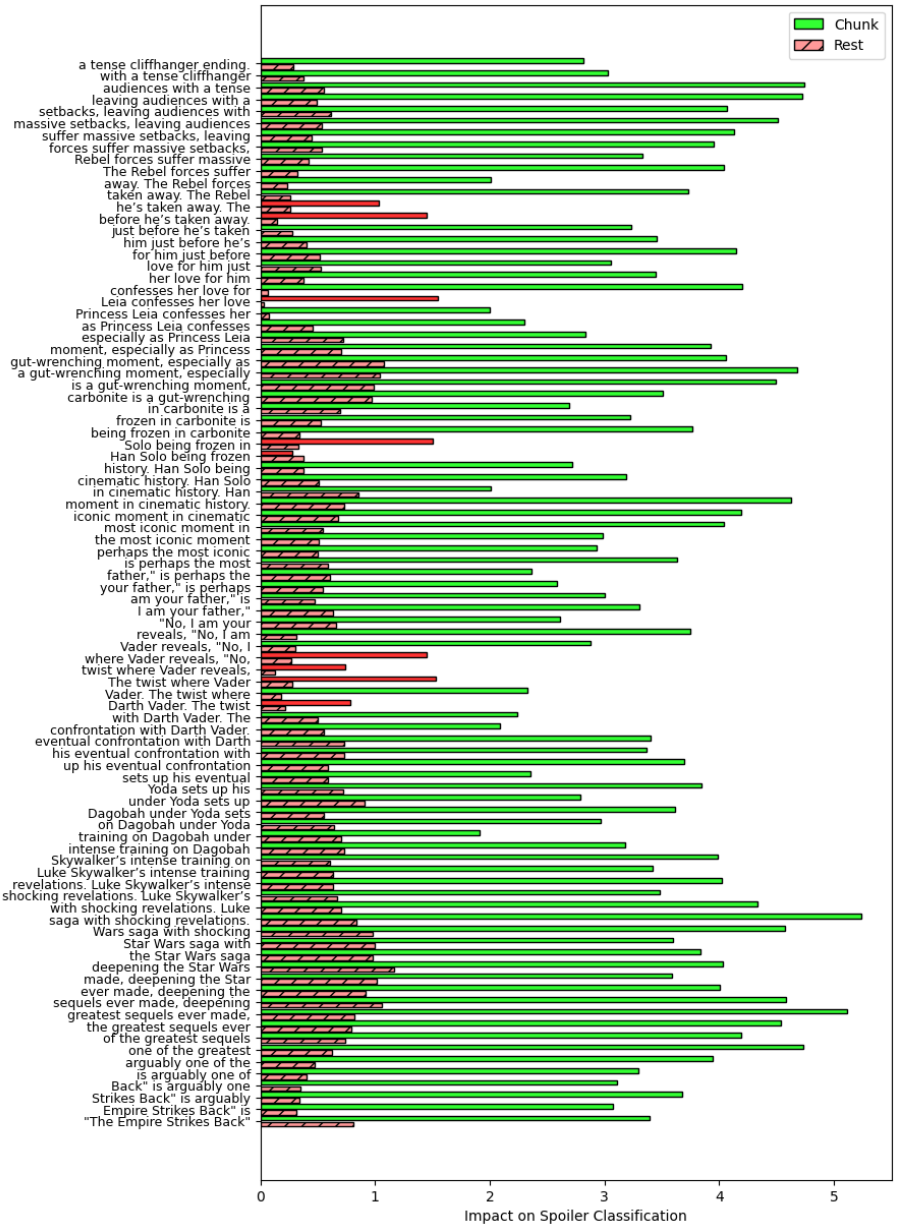

Figure 6.3: Impact of Chunks and Rest of Text on Spoiler Classification for "The Empire Strikes Back" (Chunk Size = 4)

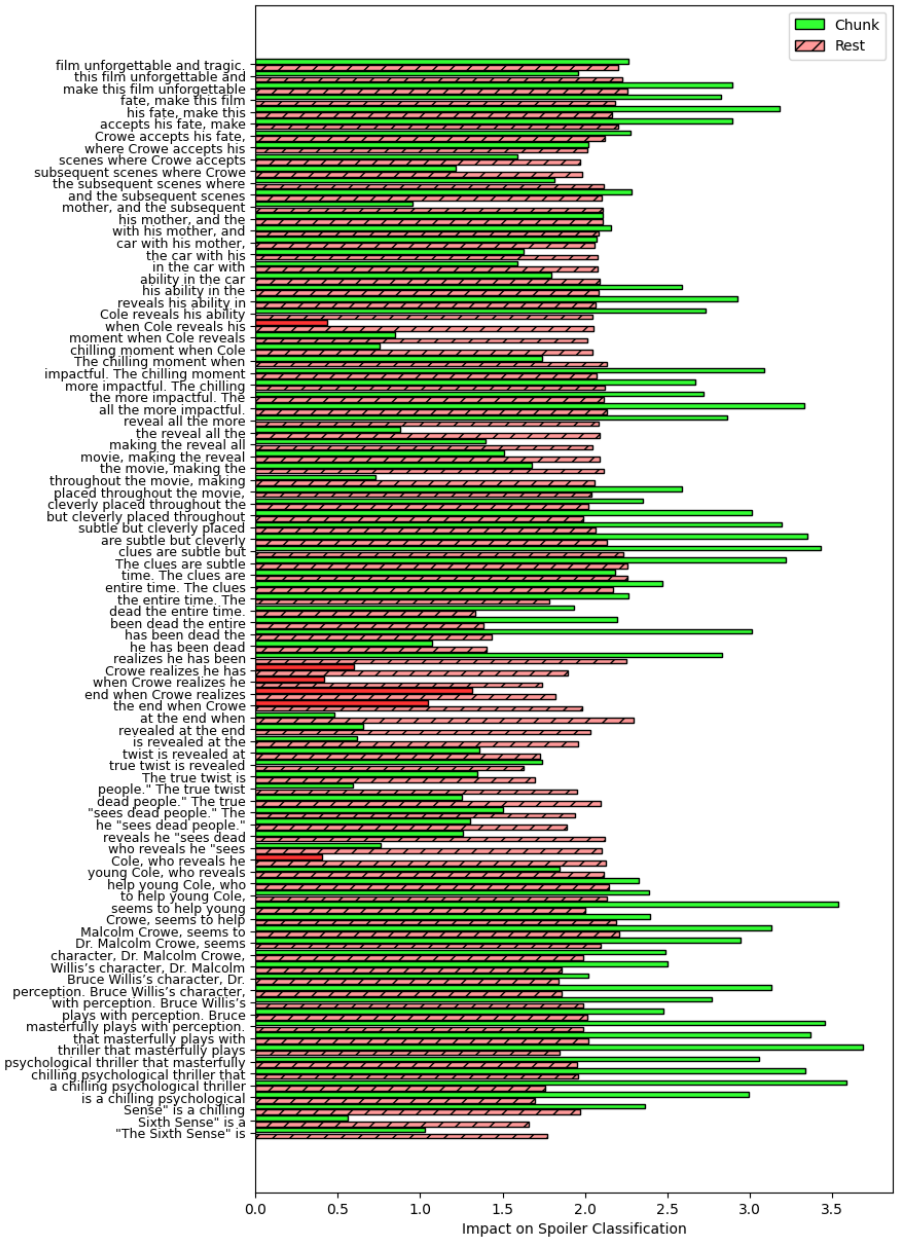

Figure 6.4: Impact of Chunks and Rest of Text on Spoiler Classification for "The Sixth Sense" (Chunk Size = 4)

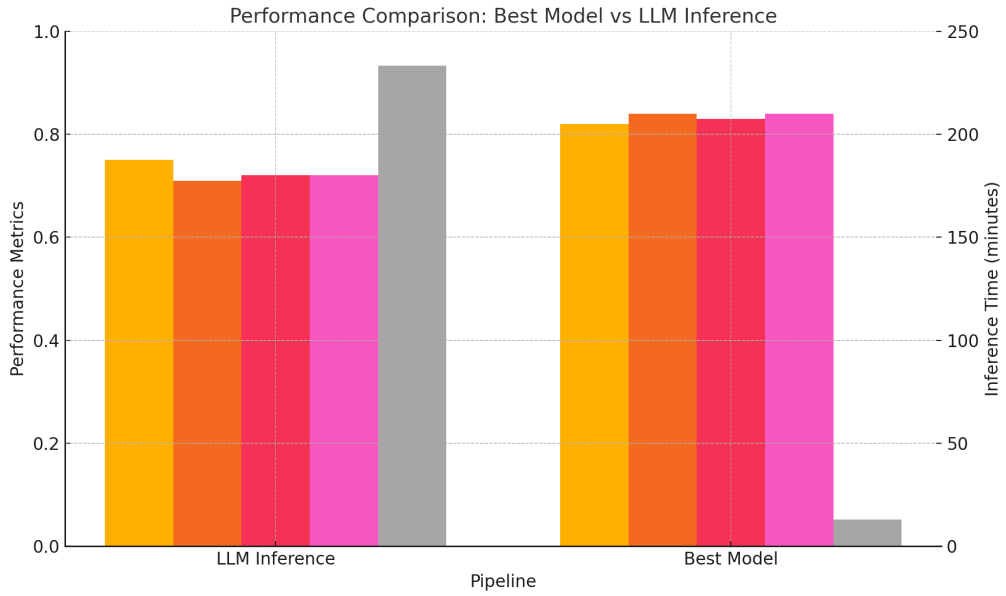

Figure 6.5: Performance metrics comparison between optimized model and LLM Inference across 55k samples.

Chapter 7
Conclusion

This research aimed to create a reliable model for spoiler detection utilizing contextual information derived from movie reviews. The model utilized advanced machine learning techniques and natural language processing to enhance traditional spoiler detection systems. This approach enables the model to effectively tackle the complexities of spoiler detection, especially in varied and contextually rich IMDb movie reviews. The research has produced a comprehensive, refined dataset that facilitates ongoing progress in content filtering and sentiment analysis in the media sector.

This research accomplished several significant objectives. A novel data cleaning method was developed, specifically designed for the complexities of character-level text inspection, resulting in a significant enhancement of dataset quality. The research involved the development of a spoiler detection model that incorporates contextual elements, including sentiment and review length, allowing for the recognition of intricate textual cues related to spoilers. A novel visualization technique was developed, facilitating a detailed analysis of the model's decision-making process. The model exhibited robustness and adaptability by achieving strong performance on previously unencountered movie reviews, highlighting its capacity for effective generalization across diverse content.

A human-centered approach to spoiler detection is developed via a survey (Appendix E). This survey will surve to gather varied human perspectives on the definition of a spoiler, an area characterized by significant individual sensitivity and subjectivity. Analyzing participant responses will enable future iterations of the model to more effectively address the diversity in human spoiler perception, thereby providing a more nuanced understanding of context-dependent spoiler markers. This insight may enable the model to tackle the subjectivity inherent in spoiler tagging and better align with user expectations.

In summary, this model establishes a new standard in spoiler detection; however, large language models (LLMs) such as GPT-4, despite their advanced capabilities, do not fully substitute for task-specific models in this domain. While LLMs are capable of recognizing general language patterns and executing a range of tasks, the task-specific model developed in this study is finely tuned to detect spoilers, incorporating distinct contextual cues relevant to movie reviews. Large language models encounter constraints stemming from their absence of targeted training on datasets specifically focused on

spoilers, resulting in difficulties in recognizing nuanced spoilers that are conveyed through subtle or indirect language. The costs and computational demands of fine-tuning large language models render them impractical for extensive spoiler tagging.

The findings indicate that ongoing refinement of targeted models, bolstered by human insights, will optimally address the changing requirements of spoiler detection systems. Task-specific models, refined through curated datasets are crucial for addressing the intricate demands of context-aware content moderation, thereby optimizing user experience across digital media platforms.

Bibliography

[1] A. Al Sulaimani and A. Starkey. 2021. Short Text Classification Using Contextual Analysis. *IEEE Access* (2021). DOI:http://dx.doi.org/10.1109/ACCESS.2021.3125768

[2] Jay Alammar. 2018. The Illustrated Transformer. (2018). https://jalammar.github.io/illustrated-transformer/

[3] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, and D. Amodei. 2020a. Language Models are Few-Shot Learners. In *Advances in Neural Information Processing Systems*, Vol. 33. 1877–1901.

[4] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, and D. Amodei. 2020b. Language Models are Few-Shot Learners. *arXiv preprint arXiv:2005.14165* (2020).

[5] V. Burr. 2015. *Social Constructionism*. Routledge.

[6] B. Chang, H. Kim, R. Kim, D. Kim, and J. Kang. 2018. A Deep Neural Spoiler Detection Model Using a Genre-Aware Attention Mechanism. In *PAKDD*. Retrieved from https://arxiv.org/abs/1809.00732.

[7] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv preprint arXiv:1810.04805* (2018). Retrieved from https://arxiv.org/abs/1810.04805.

[8] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics*. 4171–4186.

[9] N. B. Ellison, C. Steinfield, and C. Lampe. 2007. The benefits of Facebook "friends:" Social capital and college students' use of online social network sites. *Journal of Computer-Mediated Communication* 12, 4 (2007), 1143–1168. DOI:http://dx.doi.org/10.1111/j.1083-6101.2007.00367.x

[10] Rudolf Flesch. 1948. A new readability yardstick. *Journal of applied psychology* 32, 3 (1948), 221–233.

[11] T. Gao, A. Fisch, and D. Chen. 2021. Making Pre-trained Language Models Better Few-shot Learners. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. 3816–3830.

[12] B. K. Johnson and J. E. Rosenbaum. 2015. Spoiler alert: Story spoilers can hurt entertainment. *Communication Research* 43, 6 (2015), 863–882. DOI:http://dx.doi.org/10.1177/0093650214564051

[13] Y. Kim. 2014. Convolutional Neural Networks for Sentence Classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 1746–1751.

[14] J Peter Kincaid, Robert P Fishburne, Richard L Rogers, and Brad S Chissom. 1975. *Derivation of new readability formulas (automated readability index, fog count and flesch reading ease formula) for navy enlisted personnel*. Technical Report. Naval Technical Training Command Millington TN Research Branch.

[15] J. D. Leavitt and N. J. S. Christenfeld. 2011. Story spoilers don't spoil stories. *Psychological Science* 22, 9 (2011), 1152–1154. DOI:http://dx.doi.org/10.1177/0956797611417007

[16] J. D. Leavitt and N. J. S. Christenfeld. 2013. The fluency of spoilers: Why giving away endings improves stories. *Scientific Study of Literature* 3, 1 (2013), 93–104. DOI:http://dx.doi.org/10.1075/ssol.3.1.09lea

[17] M. J. Metzger, A. J. Flanagin, K. Eyal, D. R. Lemus, and R. M. McCann. 2003. Credibility for the 21st century: Integrating perspectives on source, message, and media credibility in the contemporary media environment. *Annals of the International Communication Association* 27, 1 (2003), 293–335. DOI:http://dx.doi.org/10.1080/23808985.2003.11679029

[18] A. Murdock. 2016. Spoiler alert: Spoilers make you enjoy stories more. *University of California News* (2016). Retrieved from https://www.universityofcalifornia.edu/news/spoiler-alert-spoilers-make-you-enjoy-stories-more.

[19] P. Thang. 2022. Do spoilers really ruin a story? Or can they make you enjoy it more? *Book Riot* (2022). Retrieved from https://bookriot.com/essays/do-spoilers-really-ruin-a-story-or-can-they-make-you-enjoy-it-more.

[20] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is All You Need. *Advances in neural information processing systems* 30 (2017), 5998–6008.

[21] D. Wadden, K. Lo, L. Wang, and I. Beltagy. 2019. Fact or Fiction: Verifying Scientific Claims. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. 3166–3175.

[22] M. Wan, R. Misra, N. Nakashole, and J. McAuley. 2019. Fine-Grained Spoiler Detection from Large-Scale Review Corpora. *arXiv preprint arXiv:1905.13416* (2019). Retrieved from https://arxiv.org/abs/1905.13416.

[23] H. Wang, W. Zhang, Y. Bai, Z. Tan, S. Feng, Q. Zheng, and M. Luo. 2023b. Detecting Spoilers in Movie Reviews with External Movie Knowledge and User Networks. *arXiv preprint arXiv:2304.11411v2* (2023). Available at: https://arxiv.org/abs/2304.11411v2.

[24] H. Wang, Y. Zhang, and R. Zhang. 2023a. Text FCG Fusing Contextual Information via Graph Learning for Text Classification. *Expert Systems with Applications* (2023). DOI:http://dx.doi.org/10.1016/j.eswa.2023.119658

[25] R. Watson. 2019. Did scientists prove that spoilers make you enjoy a movie more? *Skepchick* (2019). Retrieved from https://skepchick.org/2019/12/did-scientists-prove-that-spoilers-make-you-enjoy-a-movie-more.

[26] H. Yan and J. Guo. 2019. Leveraging Contextual Sentences for Text Classification Using a Neural Attention Model. *Computational Intelligence and Neuroscience* (2019). DOI:http://dx.doi.org/10.1155/2019/8320316

[27] Y. Yang and X. Cui. 2021. Bert-Enhanced Text Graph Neural Network for Classification. *Entropy* 23, 11 (2021), 1536. DOI:http://dx.doi.org/10.3390/e23111536

[28] Y. Zhang and B. Wallace. 2017. A Sensitivity Analysis of (and Practitioners' Guide to) Convolutional Neural Networks for Sentence Classification. In *Proceedings of the 8th International Joint Conference on Natural Language Processing (IJCNLP)*. 253–263.

Appendix A
APPENDIX: Use of Generative Digital Assistants

A.1 ChatGPT: For Text improvements

ChatGPT simplifies the preliminary review stages of report writing with interactive aid. This interactive digital assistant from OpenAI uses advanced natural language processing to understand and manipulate text according to directions. I find ChatGPT useful because it checks reports for common grammatical errors and other linguistic errors, greatly accelerating document revision. It is important to remember that ChatGPT is only used in the early phases of writing to identify and fix simpler errors. The assistance will improve editing, allowing me to focus on more vital research.

My ChatGPT prompt for starting the review process is intentionally simple:

> Please review this document for grammatical mistakes and suggest corrections where necessary. Focus mainly on verb tense consistency and proper use of technical terminology.

A.2 ChatGPT: Other Usecases

It has aided my research workflow in various ways besides text evaluation. Its ability to understand and analyze complex instructions makes it valuable for graph creation, output formatting, and code inspection and repair.

A.2.1 Output Formatting

I utilize ChatGPT to format computational outputs into LaTeX to improve the presentation of thesis research data. This ensures that the thesis looks clean and professional while meeting academic standards. One common prompt is:

> Convert the following JSON data into a LaTeX table format. Ensure the table is clear, well-organized, and suitable for inclusion in an academic paper.

A.2.2 Graph Generation

ChatGPT can create graphs from raw data. These graphics are crucial for thesis visualization. I have prompted the graph type and data points as part of the process. This example prompt shows how I use ChatGPT regarding this task:

> Generate a scatter plot using the dataset provided. Label the x-axis as 'Time (months)' and the y-axis as 'Efficiency (%)'. Please ensure the plot is suitable for academic presentation and include a brief description of the trend observed.

A.2.3 Code Debugging

I have frequently consulted ChatGPT for assistance with debugging while developing software components or simulations. The assistant efficiently identifies errors and suggests corrections. Which accelerated the development process and enhanced code reliability. An example prompt for debugging is:

> I am encountering an error with my Python script where the loop does not terminate. Here is the code snippet. Can you identify the issue and suggest a fix?

A.2.4 Other Use Cases

The versatility of ChatGPT extends to several other applications beneficial for my thesis:

- **Rephrasing and Refining Research Questions:** ChatGPT assisted me in refining research questions to ensure clarity and focus, which is vital for guiding the research methodology effectively.
- **Summarizing Research Articles:** I have used ChatGPT to summarize key points from extensive research articles to assimilate information quickly. Which allowed me to review and incorporate it into the literature study.
- **Idea Generation for Research Design:** It also served as a brainstorming tool for developing research methodologies and often provided novel approaches that I may not have initially considered.

The integration of ChatGPT into these diverse roles significantly impacts the efficiency and quality of my research output.

Appendix B APPENDIX: Description of the Dataset Used

B.1 Dataset Overview

This study used movie reviews from IMDb, an online database of film, TV, home video, video games, and streaming content information. The dataset is freely available on Kaggle at `https://www.kaggle.com/datasets/ebiswas/imdb-review-dataset`. The collection contains many textual user-generated movie reviews. Only English-language reviews were used for this investigation.

B.2 Content of the Dataset

Text-based movie reviews on IMDb reveal viewers' thoughts and sentiments regarding different films. The dataset has various fields for in-depth examination of reviews and their context.

The table below lists the dataset fields:

Field	Description
`review_id`	A unique identifier for each review.
`movie_id`	The identifier for the movie being reviewed.
`user_id`	The identifier for the user who wrote the review.
`review`	The full text of the user's review.
`rating`	The rating given by the user to the movie on a scale from 1 to 10.
`date`	The date on which the review was posted.

Table B.1: Fields available in the IMDb review dataset

This dataset was created for detailed textual content and user sentiment evaluations, making it a powerful tool for studying cinema viewership. Unique identities for each review and movie enable exact aggregation and extensive investigation, improving cinematic debate trends and patterns.

B.3 Data Collection Methodology

To ensure public accessibility, the data was originally compiled without logging into the IMDb website. Public evaluations were accessed without website membership or verification. This data was collected by viewing and downloading only publically available website content.

B.4 Ethical Considerations

An online data collection study has ethical issues. The dataset was carefully selected to exclude personally identifiable information. Personal data of review authors was not gathered; only publically available reviews were collected. Ethical research techniques preserve the privacy and anonymity of data subjects.

B.4.1 Public Accessibility

The dataset contains evaluations publicly uploaded on an open-access platform without privacy constraints. Since people share the data in public without expecting privacy, it can be used in academic study.

B.4.2 Use of Publicly Available Data

Using only publicly available data to collect the dataset reduces privacy concerns. The data was accessed like any other website user, without any special techniques or technologies that could compromise user privacy or data security.

B.5 Relevance to the Research

User-generated comments and feelings provide significant insights into public perceptions and reactions to films, making the dataset relevant to the research. English evaluations allow for extensive language and sentiment analysis, helping to grasp cinematic works' communicative and emotional effects.

This dataset has allowed the examination of broad thematic themes in film reviews and the testing of computational methods like sentiment analysis and natural language processing to analyze and interpret the data.

Appendix C APPENDIX: Summary of Reference Deep Learning Models

In this chapter, the deep learning models and essential principles that are discussed in this thesis are described. Each model and technique is broken down into its parts in a concise manner to eliminate the need to consult outside sources for fundamental information.

C.1 Deep Neural Networks (DNN)

Similar to the way the human brain processes information, Deep Neural Networks (DNNs) use multiple layers of neurons. With each successive layer, the data received becomes more abstract and composite. Because they can learn complex patterns from vast amounts of data, deep neural networks (DNNs) are useful for a variety of applications, including natural language processing, image and audio recognition, and more.

C.2 Introduction to Transformer

The Transformer model was first introduced by [20]. Since it's release, it has become foundational in modern natural language processing (NLP) models. This architecture diverges from traditional recurrent and convolutional models. Instead, it relies on self-attention mechanisms. In this report, I will summarize the core aspects of the Transformer architecture from visualizations provided by [2] for enhanced understanding.

C.2.1 Encoder-Decoder Structure

The Transformer consists of an encoder-decoder structure, where both components comprise multiple layers of self-attention and feedforward networks. The encoder processes the input sequence and the decoder generates the output sequence step-by-step. The self-attention mechanism in both parts allows the model to attend to different parts of the input when making predictions.

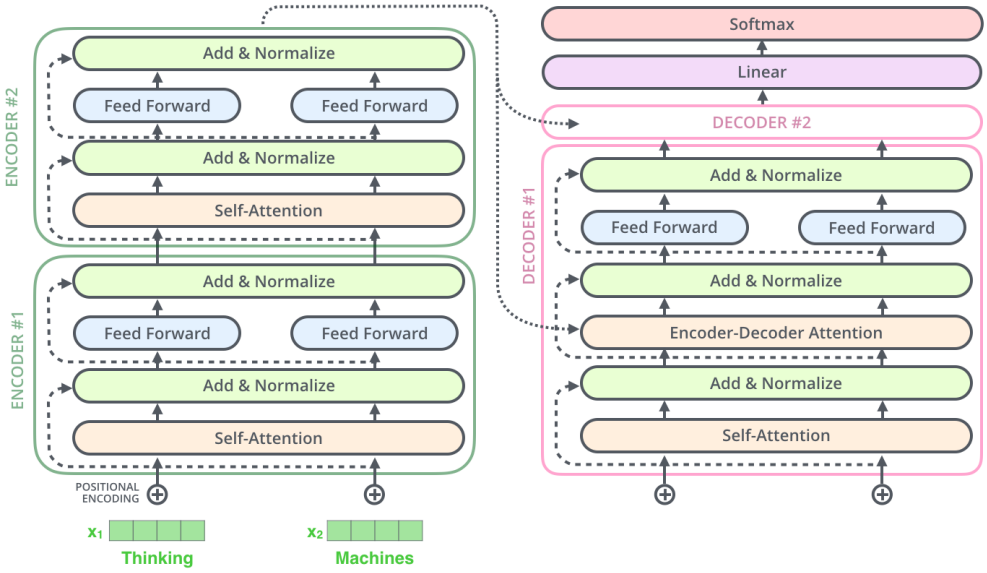

Figure C.1: An illustration of the Transformer architecture, adapted from [2].

Figure C.1 illustrates the architecture of the Transformer by showing how the encoders and decoders work together to process input and generate output.

C.2.2 Self-Attention Mechanism

Self-attention is the core idea behind the Transformer. It enables the model to weigh the importance of different words in a sequence relative to each other. This is computed using three vectors for each input word: Query (Q), Key (K), and Value (V). The attention score is calculated as:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{C.1}$$

where d_k is the dimension of the keys, and the softmax function is applied to normalize the attention weights.

C.2.3 Multi-Head Attention

The Transformer uses multi-head attention to allow the model to capture various features of the input at different positions. This splits the input into multiple heads and applies self-attention to each head. Then the results are concatenated which enables the model to consider multiple representations of the input sequence simultaneously.

C.2.4 Position-wise Feedforward Networks

Each layer of the encoder and decoder includes fully connected feedforward networks. These networks are applied independently to each position in the sequence with the same parameters across all positions.

C.2.5 Positional Encoding

Positional encodings are used to inject a sense of sequence into the input since the Transformer lacks an inherent understanding of word order. These encodings are added to the input embeddings and are derived using sine and cosine functions at varying frequencies:

$$PE_{(pos,2i)} = \sin\left(\frac{pos}{m^{2i/d}}\right) \tag{C.2}$$

$$PE_{(pos,2i+1)} = \cos\left(\frac{pos}{m^{2i/d}}\right) \tag{C.3}$$

Where pos is the position of the word in the sequence, i is the dimension, d is the total dimension, and m is a constant that controls the spread of the positional encoding.

C.3 Concrete Example: Translating "Thinking Machines"

Let's walk through an example using the phrase "Thinking Machines" to demonstrate how the Transformer processes input and generates output.

- **Positional Encoding**: The words "Thinking" and "Machines" are first converted into vectors (numerical representations). Positional encodings are added to these vectors to ensure the model understands the order of the words.
- **Self-Attention in the Encoder**: The encoder applies self-attention to the input, allowing the model to focus on how the words "Thinking" and "Machines" relate to one another. For example, it might recognize that "Thinking" modifies "Machines," so they are strongly related.
- **Multi-Head Attention**: The self-attention is performed multiple times in parallel using different heads. One head might focus on the relationship between "Thinking" and "Machines," while another might focus on the individual meanings of the words.
- **Feedforward Networks**: After the attention mechanisms, each word passes through a feedforward network for further refining its representation based on the information from the self-attention layer.

- **Decoding**: The decoder generates the output step-by-step. First, it might predict "Machines" and then use this prediction along with the encoded input to predict the next word, such as "Pensantes" (thinking in French). The final translation of "Thinking Machines" would be "Machines Pensantes."

C.4 Advantages of the Transformer

The Transformer architecture has several key advantages over older models like Recurrent Neural Networks (RNNs) and Convolutional Neural Networks (CNNs). The Transformer processes all words in a sentence simultaneously rather than one word at a time. This kind of mechanism allows it for faster training. The self-attention mechanism allows the model to capture relationships between words that are far apart in a sentence which RNNs often struggle to capture. In summary, the Transformer revolutionized natural language processing with its parallelized architecture and ability to model long-range dependencies. BERT (Bidirectional Encoder Representations from Transformers) is one of the first and most notable architecture to utilize this mechanism.

C.5 BERT Pre-training

BERT works with a Masked Language Model (MLM) and Next Sentence Prediction, two pre-training tasks. In MLM, random words in a sentence are hidden, and the model is trained to predict those words based on the context of the other words in the phrase. Using this method, BERT can interpret language in both directions.

C.6 Models Overview

This section breaks down the deep learning models included in this thesis into their respective designs, functions, and applications.

C.6.1 BERT-base-uncased

The BERT-base-uncased model is included in the Bidirectional Encoder Representations from Transformers (BERT) models developed by Google. This model does not consider case and is appropriate for applying to generic language interpretation. Twelve transformer layers, seven hundred and sixty-eight hidden layers, and twelve self-attention heads with one hundred and ten million parameters total. The entire English Wikipedia and BookCorpus are used to pre-train the BERT-base-uncased algorithm. Numerous applications, including named entity recognition, sentiment analysis, and question answering, make extensive use of it because of its ability to comprehend linguistic context.

C.6.2 BERT-large-uncased

The BERT-large-uncased architecture, consisting of 24 transformer layers with a disguised size of 1024 and 16 self-attention heads, can store more than 340 million parameters. Because of its expanded size, it is able to recognize more intricate data patterns and nuances, which improves its performance on difficult tasks such as fine-grained sentiment analysis, sophisticated question answering, and language inference.

C.6.3 RoBERTa-base-uncased

Facebook AI's RoBERTa (Robustly Optimized BERT Pre-training Approach) trains BERT on more data, with larger batches, and without the Next Sentence Prediction (NSP) aim, concentrating solely on the Masked Language Model (MLM) target. RoBERTa was developed by Facebook AI. RoBERTa-base-uncased makes use of the architectural foundation that BERT-base provides. Still, the foundation is optimized to improve its robustness and efficacy over a more significant number of datasets and benchmarks. Changing only the training regimen and the data it receives is enough to make it perform better than BERT on several NLP benchmarks.

C.6.4 DistilBERT-uncased

DistilBERT is developed by the Hugging Face team, is a distilled version of BERT (Bidirectional Encoder Representations from Transformers). Distillation is a compression technique that reduces the size of a model while retaining a significant percentage of its original performance. DistilBERT was trained to achieve approximately 97% of BERT's performance on several natural language understanding tasks while being 40% smaller and 60% faster. It does this by leveraging a smaller transformer network that closely mimics the behavior of the larger BERT model during pre-training while focusing on efficiency without substantially sacrificing accuracy. DistilBERT is ideal for use in production settings or scenarios with limited computational resources.

C.6.5 Llama 3.1 Instruct 8B

The more recent big language model, Llama 3.1 Instruct 8B, is capable of handling complex instructions. This 8-billion-parameter model can interpret prompts and write language that is reminiscent of human writing. Its features include the ability to summarize lengthy documents, produce content, and code.

C.6.6 RoBERTa-Llama 3.1405B Twitter Sentiment

Within this one-of-a-kind model, the robust pre-training process of RoBERTa is combined with the instructive capabilities of Llama models. When it comes to interpreting and comprehending the subtleties of sentiment in brief, informal writing, particularly on Twitter, this algorithm uses 1405 billion different elements. Monitoring social media, conducting brand sentiment analysis, and tracking trends in public opinion are all aided by this tool.

Appendix D
APPENDIX: Prompt for Spoiler Classification

The following prompt was used to guide the Large Language Model (LLM) in identifying spoilers within movie review segments. It ensures a structured and comprehensive analysis, enhancing the interpretability and reliability of classification decisions.

D.1 Spoiler Classification Prompt

You are analyzing movie review segments to identify potential spoilers. Please assess each review segment through the following analytical framework:

Context Analysis

Examine the review segment for key narrative elements:

- Identify explicit story events and revelations.
- Note character developments and relationships.
- Recognize described plot progressions.
- Consider thematic revelations.

Narrative Significance

Assess the centrality of revealed information:

- Evaluate the impact on major plot developments.
- Consider relationship to core story mysteries.

- Analyze connection to primary character arcs.
- Determine relevance to main narrative questions.

Temporal Positioning

Analyze when this information is typically revealed:

- Map the temporal location within standard story structure.
- Consider typical audience discovery timing.
- Evaluate relationship to major plot points.
- Note dependencies on prior revelations.

Experiential Impact

Determine the effect on viewer experience:

- Assess impact on narrative tension.
- Evaluate effect on story expectations.
- Consider influence on emotional resonance.
- Analyze effect on key story surprises.

Provide Your Analysis

1. Content Overview: [Describe identified story elements]
2. Plot Analysis: [Evaluate narrative significance]
3. Temporal Assessment: [Analyze reveal timing]
4. Impact Evaluation: [Assess viewer experience effect]

Classification

Based on your analysis, provide:

- Final Classification: [SPOILER / NON-SPOILER]
- Confidence Level: [HIGH / MEDIUM / LOW]
- Key Factors: [List primary reasons for classification]

D.2 Review Segment for Analysis

"[Review text here]"

Appendix E
APPENDIX: Survey Plan

Purpose of the Study

This survey plan explores how spoilers in movie reviews affect viewer experiences. It also examines how contextual information can improve automatic spoiler detection systems. The findings will support the development of more effective spoiler detection tools.

Study Procedures

Participants will complete a survey that takes about 10-15 minutes. The survey includes three preliminary sections:

1. **Demographic Information and Movie Preferences:** This section gathers information on age, gender, movie-watching habits, and genre preferences. It aims to identify trends based on demographics.
2. **Impact and Perception of Movie Reviews:** This section explores how participants interact with movie reviews. It covers their sensitivity to spoilers and their preferences for different review formats.
3. **Spoiler Awareness and Sensitivity:** This section assesses participants' ability to recognize spoilers. It also explores their sensitivity to spoilers and the role of context in spoiler detection. Participants can suggest improvements for spoiler detection methods.

Spoiler Identification Exercise

In the final section, participants complete a spoiler identification task. They will:

- **Select two movies**: One movie they know and one unfamiliar movie. Both are chosen from provided lists of popular and lesser-known movies.
- **Review Analysis**: For each selected movie, participants will read two IMDb reviews. They will mark any lines they believe contain spoilers. This task helps measure participants' spoiler recognition skills based on their familiarity with the movie.

The survey link for further details is available at this URL.

Confidentiality and Voluntary Participation: The research team will keep all responses confidential. Data will remain secure and accessible only to the team. Participation is voluntary. Participants may withdraw at any time without penalty.

Appendix F
APPENDIX: List of Abbreviations

This chapter enumerates the abbreviations and their corresponding full forms used throughout this thesis. Understanding these abbreviations will assist the reader in comprehensively grasping the content discussed in various chapters.

Abbreviation	**Full Form**
AI	Artificial Intelligence
AUC	Area Under the Curve
ASCII	American Standard Code for Information Interchange
BERT	Bidirectional Encoder Representations from Transformers
BLSTM	Bidirectional Long Short-Term Memory
BOW	Bag of Words
CISPA	Center for IT-Security, Privacy and Accountability
CNN	Convolutional Neural Network
COVID	Coronavirus Disease
ChatGPT	Chat Generative Pre-trained Transformer
DNN	Deep Neural Network
FC	Fully Connected (used in the context of neural networks as FC Layer)
FCG	Fully Connected Graph
F1	F1 Score
GCN	Graph Convolutional Network
GNN	Graph Neural Network
GPT	Generative Pre-trained Transformer
GRU	Gated Recurrent Unit
HAN	Hierarchical Attention Network
HGT	Heterogeneous Graph Transformer
IDF	Inverse Document Frequency
LCS	Longest Common Subsequence
LLaMA	Large Language Model A (a specific type of language model)
LLMs	Large Language Models
MLM	Masked Language Model
MSVD	Multi-Sentence Video Description Corpus
NLP	Natural Language Processing
NSP	Next Sentence Prediction
OpenAI	Organization associated with AI research
PUA	Potentially Unwanted Application
RNN	Recurrent Neural Network
TF	TensorFlow
VGCN	Variational Graph Convolutional Network

Table F.1: Abbreviations used in the thesis