Identification of spoilers
Project Proposal for NLP Course, Winter 2025

Magdalena Jeczeń
Warsaw University of Technology
magdalena.jeczen.stud@pw.edu.pl

Piotr Rowicki
Warsaw University of Technology
piotr.rowicki.stud@pw.edu.pl

Krzysztof Wolny
Warsaw University of Technology
krzysztof.wolny.stud@pw.edu.pl

supervisor: Anna Wróblewska
Warsaw University of Technology
anna.wroblewska1@pw.edu.pl

Abstract

The proposed project will conduct a comparative review of different methodologies for spoiler identification. The objective is to evaluate various Natural Language Processing (NLP) approaches, from pretrained Large Language Models (LLMs) to traditional models such as TF-IDF, on this task.

The proposed deliverable is a comprehensive evaluation that details the performance, strengths, and weaknesses of the different NLP methods applied to spoiler identification.

The study is conducted on a unified dataset composed of movie and book reviews originating from the IMDB Spoiler Dataset and the balanced Goodreads Spoiler Dataset. The evaluated approaches include classical machine learning models as well as transformer-based architectures.

1 Introduction

1.1 Scientific goal

The scientific goal of the proposed project is to create a comprehensive, benchmarked comparison of different NLP methodologies. Key research questions we would like to answer are:

1. Which methodology provides superior results in terms of metrics?

2. Which methodology proves most time-efficient in terms of predictions?

3. If there are any scenarios where traditional models would prove to be a better solution than LLMs?

The central hypothesis is that LLMs will provide significantly better results, albeit with worse time efficiency, but will prove to be the best solutions in real-world scenarios. The primary contribution of this work is a controlled and reproducible benchmark of spoiler detection methods that jointly evaluate predictive performance, robustness across repeated evaluations, and computational efficiency. In contrast to prior studies focusing solely on accuracy-oriented comparisons, this work emphasizes stability of results and inference-time trade-offs across fundamentally different modeling paradigms.

1.2 Significance of the project

The proposed project is highly significant because it provides an in-depth analysis of different spoiler identification methodologies. By conducting a comprehensive comparison, our work will yield empirical properties of the tested methodologies, which are crucial for practical application. Our

findings will offer actionable guidance, identifying the scenarios where the high resource cost of LLMs is justified by a significant performance gain, versus situations where simpler, more efficient traditional models provide a "good enough" solution for real-time, low-resource deployment.

Recent advances such as the GUSD framework (Zhang et al., 2025) underscore how modern spoiler detection systems increasingly benefit from multimodal information beyond raw text. While our project deliberately limits its scope to text-only methods, comparing classical models with LLMs will allow us to evaluate how far purely linguistic approaches can approach the performance of richer, metadata-enhanced architectures.

1.3 Literature overview

The task of spoiler detection is a specialized form of text classification that seeks to identify content that reveals critical, unreleased plot information about a piece of media (e.g., books, movies, TV shows). The evolution of methodologies in this field mirrors the broader trends in (NLP), moving from feature engineering and classical machine learning to deep learning and, more recently, LLMs.

Initial research framed the spoiler detection task as a binary classification problem: determining whether a given text snippet is a spoiler or not. One approach of this method was described in (Iwai et al., 2014) where Bag of Words(Bow) was used to extract features from text, and then classification rules were created by using Support Vector Machines(SVM) and Naive Bayes algorithms. Those methods however, often required additional metadata like movie genre to increase performance.

A major milestone in spoiler detection research was introduced by Wan et al. (Wan et al., 2019), who formalized the task as fine-grained spoiler detection and provided one of the first large-scale, systematically benchmarked evaluations on IMDB and Goodreads datasets. Their work established strong non-neural baselines and defined experimental protocols that later studies widely adopted. As a peer-reviewed contribution published at ACL, this work serves as a key reference point for subsequent developments in spoiler detection.

The introduction of Transformer-based models marked a significant breakthrough in NLP. Models such as BERT (Devlin et al., 2019) introduced bidirectional contextual representations, enabling deeper semantic understanding of entire input sequences. Subsequent empirical studies demonstrated that appropriate fine-tuning strategies are critical for achieving optimal performance in text classification tasks (Sun et al., 2020). As a result, transformer-based architectures have become state-of-the-art solutions across a wide range of NLP benchmarks (Wolf et al., 2020). In the context of spoiler detection, recent works employ transformers as strong textual backbones, often combined with additional metadata or multimodal components (Zeng et al., 2025).

Recent state-of-the-art research further expands the scope of spoiler detection beyond purely textual modeling. In particular, Zhang et al. (2025) introduce the GUSD framework (Zhang et al., 2025), which demonstrates that the effectiveness of spoiler detection can be significantly improved by incorporating non-textual signals such as movie genres and user-specific behavior patterns. Their findings show that spoiler frequency varies substantially across genres and that certain users systematically produce spoiler-heavy reviews. By combining graph neural networks with a genre-aware Mixture-of-Experts architecture, GUSD achieves leading results on benchmark datasets. Although our project focuses exclusively on comparing text-based NLP approaches—from traditional machine learning to LLMs—this work provides an important reference point, highlighting how additional metadata can influence model performance.

1.4 Relevant Datasets

One of the most widely used benchmark dataset for movie-related spoiler detection is the IMDB Spoiler Dataset, proposed by (Misra, 2022). This dataset contains over 573,000 user reviews across approximately 1,500 movies and television shows. Roughly 26% of the reviews are labeled as spoilers. Beyond the raw text, the dataset includes metadata, such as movie genres and user ratings. For research focusing on literary reviews, benchmark datasets include the Goodreads Dataset (Wan et al., 2019). This dataset provides a significant amount of over 1.3 million book reviews, where individual sentences are tagged whether they contain spoiler content.

The TV Tropes dataset (Boyd-Graber et al., 2013) leverages crowd-sourced labels from the TV Tropes library. This dataset provides labels at a more granular level. It pinpoints the exact boundaries of segments containing spoilers. It creates a valuable resource for spoiler detection across diverse media, including film, games, and literature.

1.5 Concept and work plan

- **Phase 1: Data acquisition, preparation, and initial analysis.** In this foundational step, we will take a closer look at the available datasets. Many of the referenced papers provide sources to these sets. We plan to review them and select a subset appropriate for our task. After analyzing the data, we will transform it into a unified format suitable for all models and methodologies.

- **Phase 2: Models Selection:** This is a short but crucial step where we will exactly define the models for our experiments. As previously mentioned, we plan to compare classical approaches with modern LLMs. In this phase, we will explicitly define the pipelines for the classical models (e.g., Vectorizers and classification algorithms) and determine which pre-trained LLMs to fine-tune.

- **Phase 3: Training Models:** With all the data prepared and models defined, we will move to training our models. Classical models will not cause significant issues related to computational power. For the LLMs, we will utilize our private GPUs, and if necessary, we also have Cloud Providers at our disposal.

- **Phase 4: Preparing testing suite:** After training our models, we will proceed to testing them. For this purpose, we plan to create a unified testing interface, which will allow us to standardize the results and facilitate the extension of our research to other methodologies in the future.

- **Phase 5:Testing and documenting** The final step of our project will consist of performing both performance and efficiency tests on our models, followed by documenting our findings comprehensively. We will answer our research questions and confront our initial hypotheses based on the quantitative results

All phases outlined above have been completed and are reflected in the experimental results presented in the subsequent sections. The final report consolidates data analysis, model training, benchmarking, and evaluation outcomes into a coherent experimental study.

1.6 Approach & Research methodology

The final project evaluation will consist of extensive quantitative analysis. Indicators will include not only classical classifier metrics (like F1-score or recall) but also performance-based metrics, which are most crucial for our analysis. Model training and predictions will be performed using HuggingFace, PyTorch, and sklearn APIs. Data analysis and transformation will be handled by the NumPy and Pandas libraries. For visualization, we will use Matplotlib and possibly Seaborn. We will utilize both local and cloud resources as our development and testing environments.

To contextualize our findings, we will also relate our results to state-of-the-art multimodal systems such as GUSD (Zhang et al., 2025). Although replicating full multimodal architectures such as GUSD lies outside the scope of this project, this limitation is primarily driven by practical constraints typical for a one-semester academic study. Multimodal approaches rely on additional signals such as user graphs, genre metadata, or interaction histories, which are not consistently available in public datasets or require substantial preprocessing and annotation effort. Consequently, this work deliberately focuses on text-only models to enable a controlled, reproducible, and fair comparison across fundamentally different modeling paradigms.

2 Data Analysis

We constructed a dataset by merging two primary sources: IMDB Spoiler Dataset (Misra, 2022) with film and television data and balanced Goodreads (Wróblewska et al., 2021) with literature. This cross-domain approach ensures the model can identify spoiler patterns across different types of reviews. The final dataset consists of 100,000 samples, created by taking a sample of 50,000 reviews from each source. Within this combined set, 38.1% of the reviews are labeled as containing spoilers, while the remaining 61.9% are classified as spoiler-free as presented in Table 1. The temporal range of the data spans nearly

two decades, with the earliest review dated July 28 1998, and the most recent entry recorded on January 7 2018.

Table 1: Comparison of Dataset Distributions

Dataset	Total Reviews	Spoiler %
IMDB	50,000	26.20%
Goodreads	50,000	50.00%
Combined	100,000	38.10%

2.1 IMDB Spoiler Dataset

The IMDB Spoiler Dataset (Misra, 2022) is a collection of IMDB user reviews obtained from the Kaggle IMDB Spoiler Dataset. The data was collected by scraping publicly available IMDB reviews. The IMDB_reviews.json file contains individual movie reviews along with metadata such as review text, spoiler label, rating, user ID, movie ID, and review date. The dataset includes 573,913 reviews, of which about 26.2% are marked as spoilers, providing a large and balanced-enough corpus for text-based modeling.

The IMDB dataset (Misra, 2022) was sampled with 50,000 rows to ensure computational efficiency Dataset offers a movie reviews spanning nearly two decades. The temporal distribution begins on July 28 1998 and extends through January 7 2018 showing a steady upward trajectory in engagement over time (see Figure 1). While the early years of the dataset show relatively sparse activity, there is a significant increase in the amount of reviews published monthly starting around 2005, reflecting the growing popularity of online film criticism.

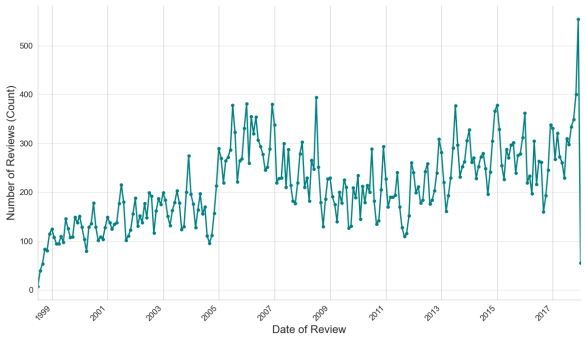

Figure 1: Time-series line graph representing the monthly count of IMDB reviews published from 1998 to 2018.

When examining the structure of these reviews, the distribution of text lengths and word counts reveals that most contributors prefer medium-length critiques; the majority of reviews fall within the range of 500 to 2,000 characters, or roughly 100 to 300 words (see Figure 2), though there is also a small tail of more exhaustive analyses.

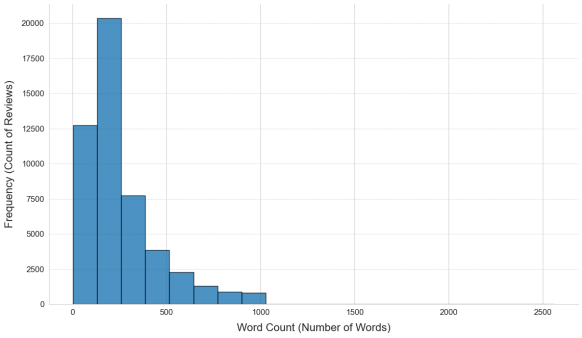

Figure 2: Distribution of review word counts in IMDB dataset.

Stopwords are dominated with words such as "the," "and," "a," and "of," with the word "the" alone appearing over 700,000 times in this sample. However, once stopwords are filtered out to reveal content words, the dataset's movie focus becomes clear. The most frequent content terms are "movie" and "film", followed by "story", "character" and "scene", which highlight the topic of the dataset (see Figure 3). Words "like", "good" and "great" suggest reviewers focus on personal opinion.

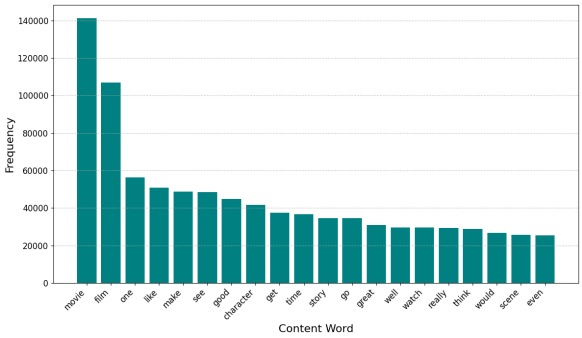

Figure 3: Top 20 most frequent content words found in IMDB reviews.

2.2 Goodreads

The Goodreads dataset (Wan et al., 2019) is a comprehensive collection of book reviews and metadata originally scraped from the Goodreads website. We are using the balanced version of the Goodreads dataset (Wróblewska et al., 2021). This balanced version was constructed by first

identifying and including every review that contained at least one spoiler sentence. An equal number of reviews without any spoilers were then randomly selected and added to the set. The final size of this balanced dataset is equal to 179,254 reviews. The dataset has an equal split between spoiler and non-spoiler reviews, with exactly 50% (25,003 reviews) containing spoilers and 50% (24,997 reviews) being spoiler-free.

We sampled the Goodreads balanced dataset with 50,000 rows for efficient analysis. Reviews in the dataset begin on May 21, 2007 and end on November 3, 2017 (see Figure 4). The temporal distribution shows a significant rise in the monthly count of reviews, starting from nearly zero in 2007 and peaking at over 1,000 reviews per month by 2017. This growth indicates a substantial increase in user activity on the platform during this period.

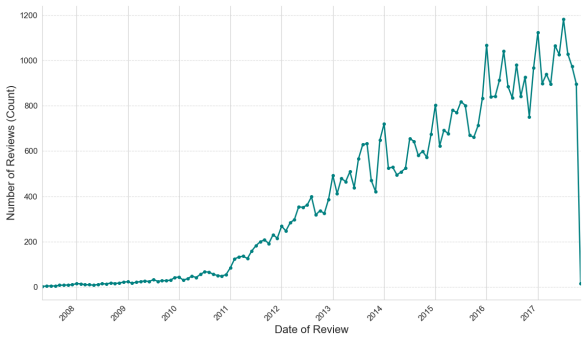

Figure 4: Time-series line graph representing the monthly count of Goodreads reviews published from 2007 to 2017.

In terms of linguistic structure, the dataset is characterized by relatively concise reviews, as evidenced by the distribution of text lengths where the vast majority of reviews are under 2,500 characters and 500 words (see Figure 5). While a small number of reviews extend significantly further, the bulk of user contributions are short-to-medium length.

The lexical analysis reveals that most popular stopwords are words such as "the," "and," "i," and "to", with "the" appearing over 600,000 times. When these are removed to highlight content words, the primary focus is on book-connected terminology like "book", "read" and "character" followed by opinion words such as "like", "love" and "good" (see Figure 6).

A comparison of the two datasets reveals notable structural differences. IMDB reviews tend to be longer and more descriptive, often focusing on narrative and cinematic elements. In contrast, Goodreads reviews are generally shorter and focus on the reading experience. Additionally, the balanced nature of the Goodreads dataset contrasts with the natural class imbalance observed in IMDB. This introduces different modeling challenges. The temporal analysis of these two platforms highlights different times of internet adoption. The IMDB data span from 1998, with a major increase in activity around 2005, where the Goodreads dataset starts much later in 2007. Combining these two datasets enables evaluation across heterogeneous domains and mitigates dataset-specific bias.

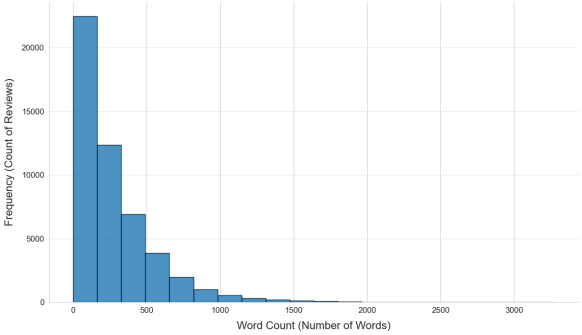

Figure 5: Distribution of review word counts in Goodreads dataset.

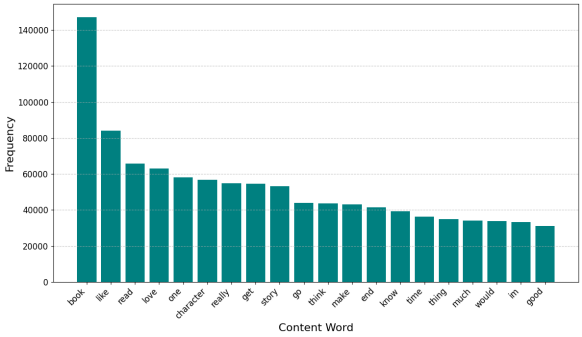

Figure 6: Top 20 most frequent content words found in Goodreads reviews.

3 Experiments

3.1 Experimental Setup

3.1.1 Data Splits and Preprocessing

The merged dataset was split into training and test sets using a stratified split with a test size of 30%, preserving the original class distribution. All experiments were conducted on the same fixed split to ensure comparability across models. Random seeds were fixed to guarantee reproducibil-

ity. All stochastic components of the experimental pipeline were controlled using fixed random seeds for NumPy, PyTorch, and the HuggingFace Transformers framework.

3.1.2 Evaluation Metrics

Model performance was evaluated using multiple metrics: Accuracy, F1-Score, Matthews Correlation Coefficient (MCC). While accuracy provides a general performance overview, MCC was selected as the primary metric for model selection, as it offers a more reliable assessment under class imbalance. Unlike accuracy and F1-score, MCC provides a balanced evaluation by accounting for all four entries of the confusion matrix. This choice is supported by prior studies demonstrating that MCC offers superior reliability for imbalanced binary classification tasks (Chicco and Jurman, 2020).

3.1.3 Hyperparameter Optimization

For classical models, hyperparameters were optimized using grid search combined with 5-fold cross-validation on the training set. The optimization focused on vectorization parameters, including the maximum number of features and n-gram range. Model performance was evaluated using the mean F1-score across folds, and the best-performing configurations were selected for final evaluation.

3.1.4 Fine-tuning Protocol for Transformer Models

Transformer-based models were fine-tuned using the HuggingFace Transformers library. Input texts were tokenized with truncation and a maximum sequence length of 256 tokens. Training was performed using the AdamW optimizer with a learning rate of 2×10^{-5} and a batch size of 16. Model checkpoints were selected based on the highest MCC score on the evaluation set. All transformer-based experiments were conducted in a GPU-enabled environment to ensure feasible training and evaluation times.

3.2 Models

The selected models represent complementary methodological paradigms commonly employed in text classification tasks. Classical models provide efficient and interpretable baselines, while transformer-based architectures represent the current state of the art in contextual language modeling. An overview of all evaluated models, together with their categories and trainable parameters, is presented in Table 2.

3.2.1 Classical Models

Classical models were selected as lightweight and interpretable baselines commonly used in text classification tasks. Both approaches rely on sparse lexical representations and class-balanced objective functions to mitigate label imbalance.

3.2.2 Transformer-based Models

The transformer-based models were initialized from publicly available pre-trained checkpoints provided by the HuggingFace model hub. Specifically, BERT experiments were based on the `bert-base-uncased` checkpoint, while RoBERTa experiments utilized the `roberta-base` checkpoint. Both models were adapted to the binary classification task through supervised fine-tuning as described in the experimental setup.

3.3 Testing Suite

The core of the proposed solution is the Testing Suite, which serves as a modular environment for rigorous model benchmarking. It is characterized by two primary functional components:

3.3.1 Tested Model interface

To facilitate seamless integration, the framework employs an abstraction layer for all candidate architectures. This interface enforces a uniform output format for model predictions, ensuring that the downstream analytical pipeline remains decoupled from specific model implementations.

3.3.2 ModelTestingSuite

The core evaluation framework benchmarks model architectures against a standardized dataset, yielding a comprehensive suite of performance metrics for comparative analysis. A novel feature of this implementation is the quantification of computational efficiency via inference latency profiling (done by measuring the time interval of predict method). We believe its a crucial feature, frequently omitted in comparisons. In order to ensure statistical robustness, execution time is recorded across multiple iterative batches, minimizing the impact of transient system noise. Users may parameterize both the batch count and the randomization seeds used for data partitioning to ensure reproducibility.

Table 2: Overview of evaluated models

Model	Category	Trainable Parameters
TF-IDF + SVM	Classical	All
BoW + Logistic Regression	Classical	All
BERT (`bert-base-uncased`)	Transformer	Last 2 layers + classifier
RoBERTa (`roberta-base`)	Transformer	All layers

4 Results

4.1 Performance Results

To evaluate the predictive performance of the candidate models, three distinct metrics were employed to provide a multidimensional view of classification quality: Accuracy, F-score (F_1), and the Matthews Correlation Coefficient (MCC).To visualize the variance and stability of these models across experimental runs, the results are presented as distribution summaries. Figures 7, 8, and 9 depict the boxplots of corresponding metrics. Results are also aggregated in table 3 As we can

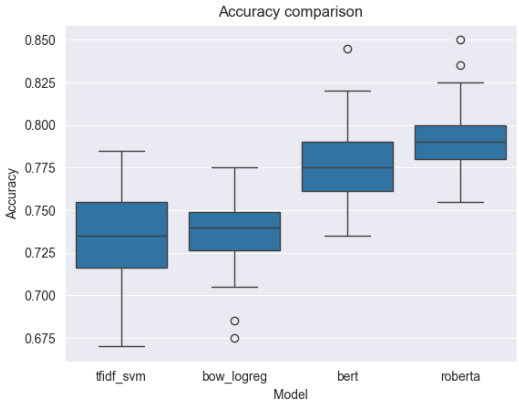

Figure 7: Comparative distribution of predictive accuracy across models

see, the general trend across all metrics shows that LLMs outperform traditional statistical methods.

We can see that RoBERTa has the strongest performance:

- It maintains the highest median across all metrics, establishing a higher performance ceiling compared to BERT and baseline models.
- The absence of low-end outliers suggests a high degree of reliability across various data partitions.
- By achieving the highest Matthews Correlation Coefficient (MCC), RoBERTa proves to be the most resilient architecture against class imbalances.

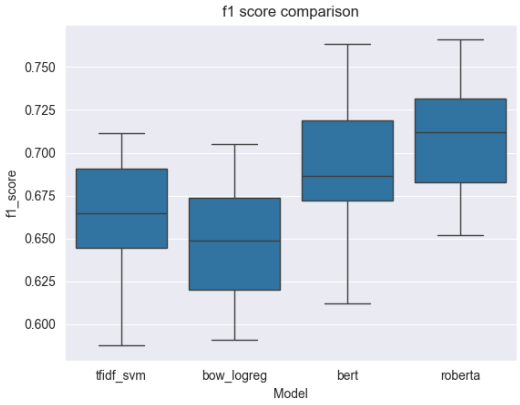

Figure 8: Comparative distribution of predictive F-score across models

While BERT demonstrates a clear performance advantage over classical baseline models, it exhibits greater variance in its predictive distribution compared to RoBERTa.

It is noteworthy that the traditional baseline models (TF-IDF with SVM and BoW with Logistic Regression) exhibit significantly inferior predictive capacity compared to the transformer-based architectures. Furthermore, the performance delta between these two classical approaches is marginal; their overlapping interquartile ranges suggest that neither model maintains a statistically definitive superiority over the other across the evaluated metrics.

4.2 Efficiency Results

The efficiency evaluation was conducted across two distinct hardware environments to assess performance under varying computational constraints: a CPU-only architecture and a GPU-accelerated system.

To quantify the impact of data volume on execution time, benchmarks were performed using two specific configurations: batch sizes of 10 and 20 records. For each iteration, the framework

Table 3: Numerical comparison of model performance

Model	Accuracy (mean ± std)	F1 (mean ± std)	MCC (mean ± std)
tfidf_svm	0.735 ± 0.030	0.665 ± 0.035	0.450 ± 0.050
bow_logreg	0.740 ± 0.020	0.650 ± 0.030	0.440 ± 0.045
bert	0.775 ± 0.025	0.690 ± 0.040	0.520 ± 0.060
roberta	0.790 ± 0.020	0.710 ± 0.035	0.550 ± 0.040

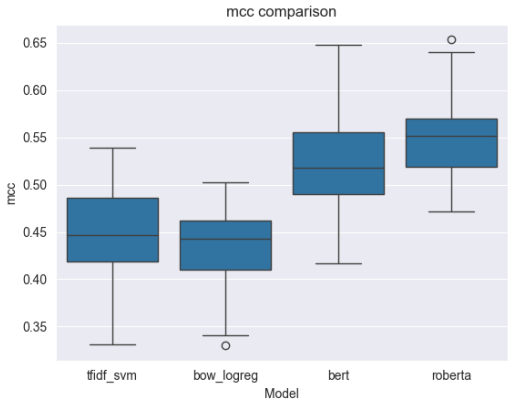

Figure 9: Comparative distribution of predictive MCC across models

recorded the aggregate latency required to process the entire batch.

Following the methodology established in the performance analysis, these outcomes are visualized as distribution summaries to account for system-level jitter. Figures 10, 12, 11, and 13 depict the resulting latency profiles for both hardware settings and batch configurations. The

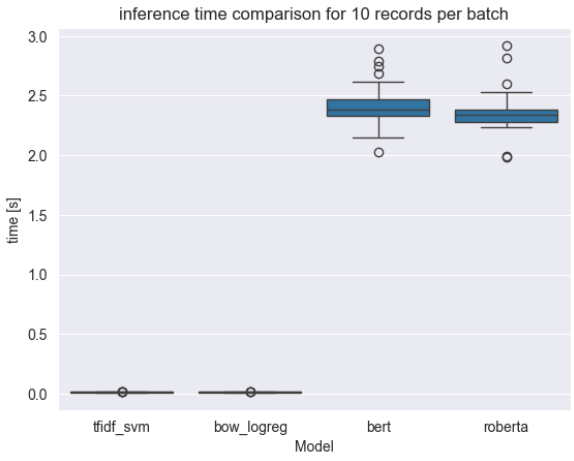

Figure 10: Comparative distribution of inference time across models for 10 records per batch on CPU

empirical results reveal a significant performance

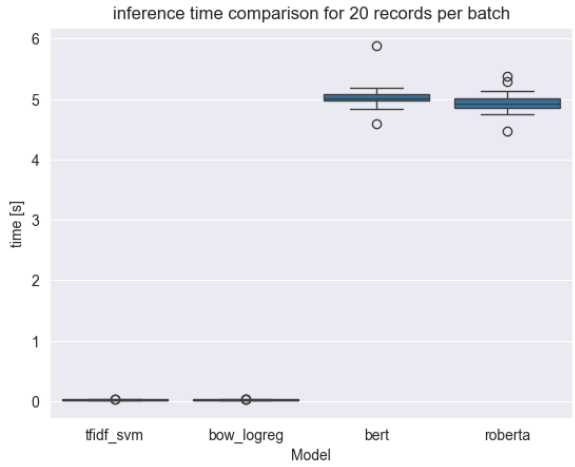

Figure 11: Comparative distribution of inference time across models for 20 records per batch on CPU

disparity between the hardware environments. In CPU-only configurations, the computational overhead of transformer-based architectures is prohibitive; the latency magnitude is so substantial that LLMs appear operationally unfeasible for real-time applications. While GPU acceleration significantly narrows the gap, the trade-offs remain nuanced:

- In scenarios requiring ultra-low latency, even GPU-accelerated LLMs may fail to meet the strict response-time thresholds maintained by traditional models.
- The integration of GPU hardware introduces significant operational complexity. This includes the increased cost of specialized Virtual Machine (VM) instances and the administrative overhead of managing CUDA drivers and environment dependencies on inference servers.

5 Conclusions and Future Work

Following the procurement and preprocessing of the dataset, four candidate architectures were systematically trained and optimized:

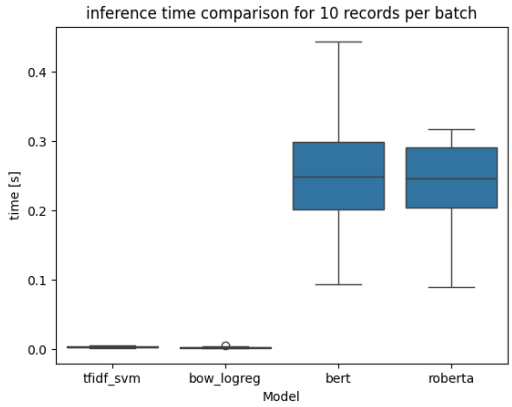

Figure 12: Comparative distribution of inference time across models for 10 records per batch on GPU

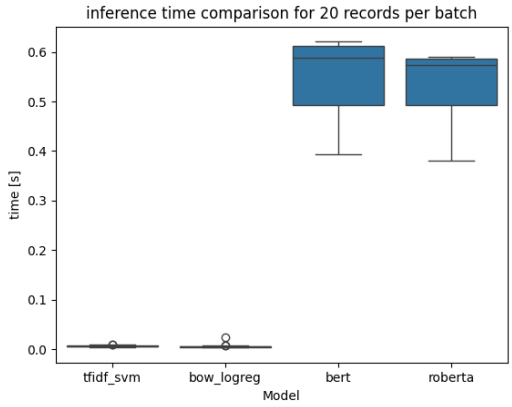

Figure 13: Comparative distribution of inference time across models for 20 records per batch on GPU

two baseline methodologies (TF-IDF with SVM and Bag-of-Words with Logistic Regression) and two Transformer-based models (BERT and RoBERTa). The experimental results demonstrate a clear performance-efficiency trade-off:

- Transformer-based architectures, specifically RoBERTa, yield significantly higher scores across all classification metrics (Accuracy, F_1, and MCC).
- Traditional statistical models maintain a decisive advantage in terms of inference latency. While GPU acceleration narrows this gap, the execution time for LLMs remains orders of magnitude higher than that of classical baselines.

Consequently, the selection of an optimal model is highly context-dependent. In latency-critical environments, the marginal loss in predictive precision associated with classical models is a necessary compromise to ensure real-time responsiveness. Conversely, in scenarios where accuracy is the primary objective and computational resources are non-limiting, RoBERTa represents the most robust solution.

5.1 Future Work

The current study establishes a clear baseline for the accuracy-efficiency trade-off within text classification. Future research will focus not only on bridging the gap between high-performance Transformer architectures and low-latency classical methods but also on expanding the core evaluation framework to incorporate a broader spectrum of models and functionalities. Three primary research avenues have been identified:

To identify a globally optimal architecture, the study could be extended to include a broader array of state-of-the-art (SOTA) pre-trained models. While acquiring and integrating these large-scale models presents logistical challenges, it would provide a more exhaustive landscape of the current field.

Future iterations could include simultaneous optimization of predictive accuracy and computational throughput. This requires a dual approach: leveraging analytical insights from classical model behavior while exploring low-level model interfaces for LLMs. Techniques such as quantization, pruning, and kernel optimization could potentially bring LLM latency closer to classical baselines without significant accuracy degradation.

To provide a more holistic comparison, the testing suite might be expanded to include automated memory profiling. Quantifying both System RAM and VRAM allocations is essential for understanding the true infrastructure cost of model deployment. While automating these measurements across heterogeneous hardware presents technical challenges, it may prove to be a beneficial step for comprehensive resource benchmarking.

References

Jordan Boyd-Graber, Kimberly Glasgow, and Jackie Sauter Zajac. 2013. Spoiler alert: Machine learning approaches to detect social media posts with revelatory information. In *ASIST 2013: The 76th Annual Meeting of the American Society for Information Science and Technology*.

Davide Chicco and Giuseppe Jurman. 2020. The advantages of the matthews correlation coefficient (mcc) over f1 score and accuracy in binary classification evaluation. *BMC Genomics*, 21(1):1–13.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina 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: Human Language Technologies (NAACL-HLT)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Hidenari Iwai, Yoshinori Hijikata, Kaori Ikeda, and Shogo Nishida. 2014. Sentence-based plot classification for online review comments. In *2014 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT)*, volume 1, pages 245–253.

Rishabh Misra. 2022. Imdb spoiler dataset. *arXiv preprint arXiv:2212.06034*.

Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. 2020. How to fine-tune bert for text classification? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1063–1073, Online. Association for Computational Linguistics.

Mengting Wan, Rishabh Misra, Ndapandula Nakashole, and Julian McAuley. 2019. Fine-grained spoiler detection from large-scale review corpora. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2605–2610, Florence, Italy. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In Qun Liu and David Schlangen, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online, October. Association for Computational Linguistics.

Anna Wróblewska, Paweł Rzepiński, and Sylwia Sysko-Romańczuk. 2021. Spoiler in a textstack: How much can transformers help? *arXiv preprint arXiv:2112.12913*.

Zinan Zeng, Sen Ye, Zijian Cai, Heng Wang, Yuhan Liu, Haokai Zhang, and Minnan Luo. 2025. Mmoe: Robust spoiler detection with multi-modal information and domain-aware mixture-of-experts.

Haokai Zhang, Shengtao Zhang, Zijian Cai, Heng Wang, Ruixuan Zhu, Zinan Zeng, and Minnan Luo. 2025. Unveiling the hidden: Movie genre and user bias in spoiler detection. *arXiv preprint arXiv:2504.17834*.

A Reviewers Feedback and Improvements

Two independent teams conducted a review of our initial report and solution, identifying several areas requiring attention.

Below, we explicitly map each major reviewer comment to the concrete actions taken in the revised version of the report, together with references to the affected sections.

The first significant issue raised was the gaps in our state-of-the-art analysis. To address this, we extended section 1.3. Specifically, we extended the literature review to include a peer-reviewed state-of-the-art reference by Wan et al. (Wan et al., 2019), published at ACL, which serves as a canonical benchmark for spoiler detection on IMDB and Goodreads datasets. Additionally, we strengthened the methodological grounding of transformer-based models by referencing empirical fine-tuning studies (Sun et al., 2020), clarifying the positioning of our work relative to existing state-of-the-art approaches. This revision directly addresses the reviewers' concerns regarding insufficient coverage of canonical benchmarks in prior versions of the report (Section 1.3).

Reviewers identified a need for a more robust justification of the efficiency comparison. In response, we expanded the documentation of the experimental setup to include a granular description of the inference profiling logic (Sections 3.3 and 4.2).

A secondary recommendation suggested evaluating models based on training duration. However, this metric was deliberately excluded from the final analysis. Since the proposed solution focuses on real-time deployment and model serving, training time—a one-off computational cost—was deemed to offer negligible insight for end-users compared to inference latency, which directly impacts the user experience.

Another significant refinement prompted by the review was the expansion of the evaluation framework (Testing Suite) to include the Matthews Correlation Coefficient (MCC). Recognizing that standard accuracy can be a misleading metric in certain classification contexts, we integrated MCC

to provide a more mathematically rigorous assessment of the models' predictive power. This extension allowed for a comprehensive re-evaluation of our experimental results, ensuring that the comparative analysis accounts for the balance between true and false positives and negatives across all candidate architectures. The inclusion of MCC also enables a more reliable comparison across models evaluated on imbalanced datasets, directly responding to the reviewers' recommendations regarding metric selection (Sections 3.1.2 and 4.1).

In summary, all major reviewer comments raised during the mid-term review were either directly addressed through methodological extensions and additional experiments, or explicitly discussed and justified when alternative design choices were adopted. This ensures that the revised report provides a more rigorous, transparent, and reproducible experimental study. Where applicable, design decisions that were not implemented (e.g., multimodal extensions) were explicitly justified based on scope and data availability constraints.

B Reproducibility

B.1 Environment and Dependencies

To ensure computational reproducibility and eliminate variance arising from environment-specific inconsistencies, all development and experimentation were conducted within a unified software stack. The system utilizes Python 3.13 as the standardized interpreter across all experimental phases. All third-party dependencies were strictly versioned and documented in the *requirements.txt* file, which is available in the project repository.

B.2 Experimental Reproducibility

To ensure the integrity of the experimental results, every process involving stochastic operations was initialized with a fixed seed of 42. This rigorous approach to pseudo-random number generation ensures that data partitioning, such as training and validation splits, as well as the random initialization of model weights, remains consistent across separate execution runs. By fixing the seed, we mitigate the risk of performance variance stemming from random chance, thereby ensuring that the comparative analysis of model architectures is both stable and reproducible.

C Workload division

The division of contributions from each team member is shown in table 4

Table 4: Division of responsibilities

Team member	Contributions
Magdalena Jeczeń	SOTA research, Classical model training, hyperparameters tuning, LLM fine-tuning, Results Analysis
Piotr Rowicki	Project scope definition, SOTA research, Testing suite design and implementation, Experiments orchestration, Results analysis
Krzysztof Wolny	Dataset research, Data preprocessing, Explanatory Data Analysis