Identification of spoilers
Project Proposal for NLP Course, Winter 2025

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

Piotr Rowicki
Warsaw University of Technology
piotr.rowicki@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.

1 Introduction

1.1 Scientific goal

Scientific goal of proposed project is to create 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 better solution than LLMs?

Central hypothesis is that LLMs will provide much better results, with worse time-efficiency, but will prove to be the best solutions in any real scenarios.

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 is 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.

The introduction of Transformer-based models marks a significant breakthrough in NLP tasks. Models like BERT(Devlin et al., 2019) introduced bidirectional context representation, allowing models to gain a deeper understanding of the entire sequence. Thanks to this, Transformers have achieved state-of-the-art results in many NLP tasks (Wolf et al., 2020). LLMs are also part of the current state-of-the-art solution for spoiler identification, where they serve as a crucial textual backbone (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 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

1.5 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 environment..

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 such architectures lies outside the scope of this project, referencing their performance helps position our work within the broader research landscape of spoiler detection.

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 (see Figure 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.

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 (see Figure 2).

The IMDB dataset (Misra, 2022) was sampled with 50,000 rows to ensure computational efficiency Datset offers a movie reviews spanning nearly two decades. The temporal distribution begins on July 28 1998 and extends through January

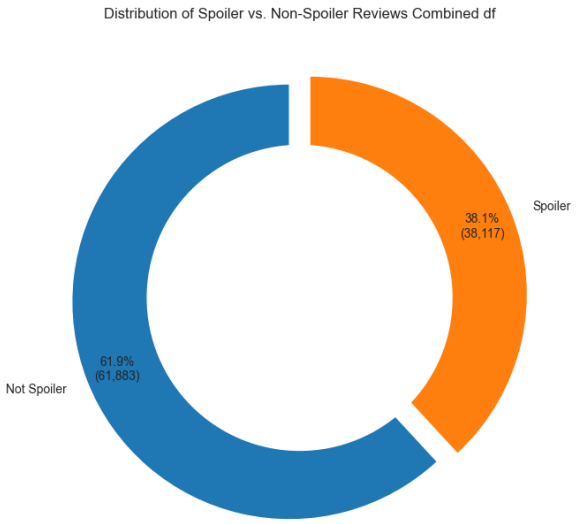

Figure 1: Distribution of spoiler vs. non-spoiler reviews in the merged dataset.

7 2018 showing a steady upward trajectory in engagement over time (see Figure 3). 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.

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 4), though there is also a small tail of more exhaustive analyses.

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 5). Words ”like”, ”good” and ”great” suggest reviewers focus on personal opinion.

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

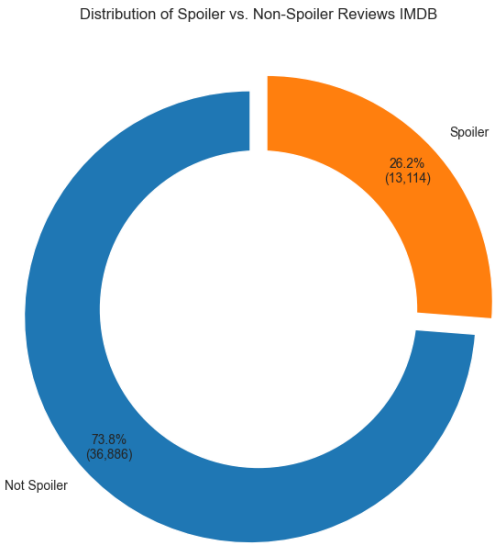

Figure 2: Distribution of spoiler vs. non-spoiler reviews in the IMDB dataset.

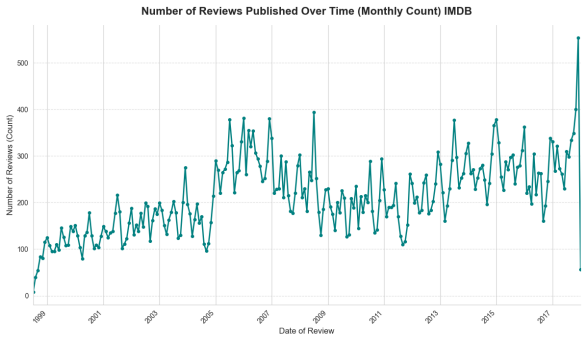

Figure 3: Time-series line graph representing the monthly count of IMDB reviews published from 1998 to 2018.

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. Dataset has equal split between Spoiler and Non-Spoiler reviews, with exactly 50% (25,003 reviews) containing spoilers and 50% (24,997 reviews) being spoiler-free (see Figure 6).

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 7). The temporal distribution shows a significant rise in the monthly count of reviews, starting from nearly zero in 2007

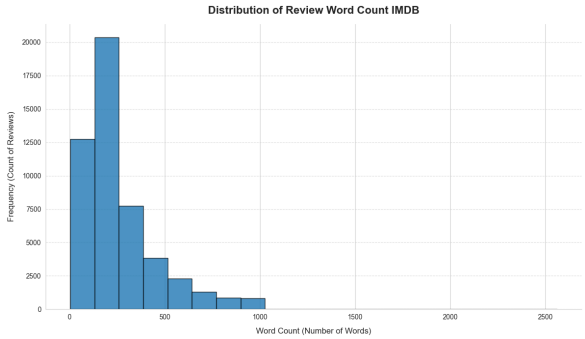

Figure 4: Distribution of review word counts in IMDB dataset.

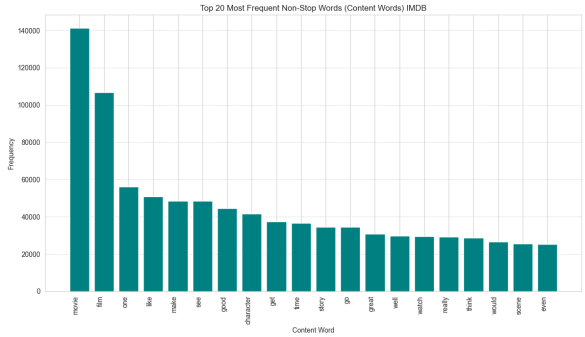

Figure 5: Top 20 most frequent content words found in IMDB reviews.

and peaking at over 1,000 reviews per month by 2017. This growth indicates a big expansion in user activity on the platform during this period.

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 8). 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-conected terminology like ”book”, ”read” and ”character” followed by opinion words such as ”like”, ”love” and ”good” (see Figure 9).

3 Experiments

With composed dataset we proceed to our experiments. We devided this process into two main stages

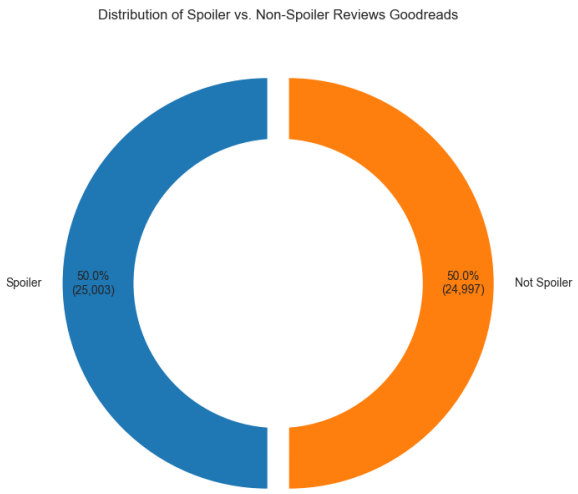

Figure 6: Distribution of spoiler vs. non-spoiler reviews in the balanced Goodreads dataset.

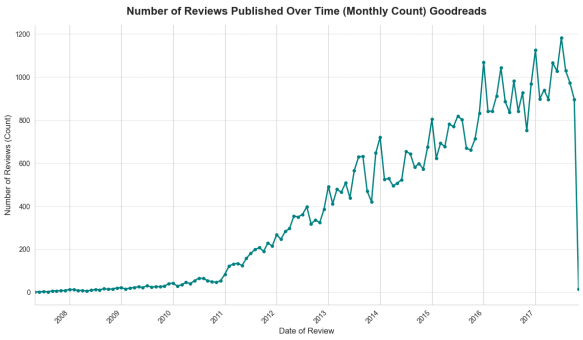

Figure 7: Time-series line graph representing the monthly count of Goodreads reviews published from 2008 to 2017.

1. Models training
2. Implementation and execution of Testing Suite

Details of those stages were described below.

3.1 Our Approach

At this stage of our work, we implemented two classical approaches and one LLM, which are described in the following subsection.

- Tfidf + SVM
  For our classical approach, we used a TF-IDF vectorizer to convert text into numerical features by weighting words according to their importance in the corpus. These features were then used to train a Support Vector Machine (SVM), classifier, which learns an optimal decision boundary for text classifiaction in a high-dimensional feature space.

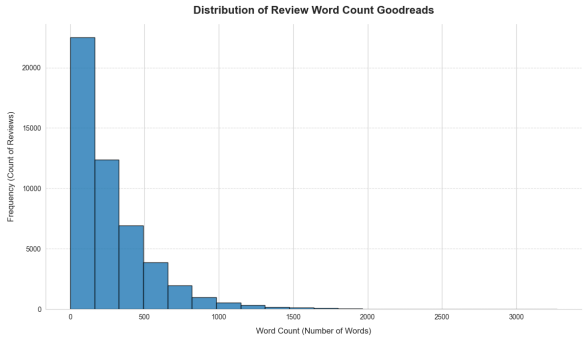

Figure 8: Distribution of review word counts in Goodreads dataset.

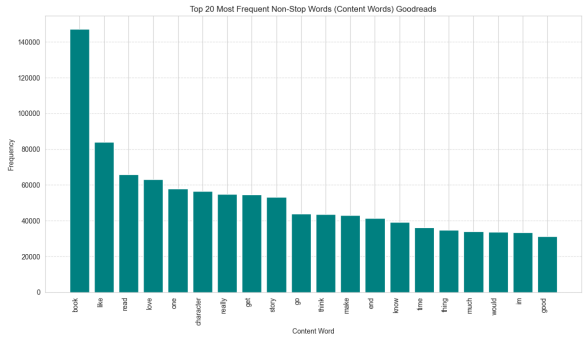

Figure 9: Top 20 most frequent content words found in Goodreads reviews.

- BoW + Logistic Regression
  Our second classical approach uses a Bag-of-Words (BoW) vectorizer to represent text as frequency-based feature vectors with unigrams and bigrams. These features were then used to train a Logistic Regression classifier, which models the probability of class membership for the text classification taks.

- BERT
  For LLM model we used BERT (Bidirectional Encoder Representations from Transformers), which is a pretrained transformer-based language model that learns contextual word representations by processing text bidirectionally. In our work, we used the bert-base-uncased model in a fine-tuning setup. The pretrained BERT encoder was frozen, and only the final pooling layers were trained.

3.2 Testing Suite

Crucial part of our solution is Testing Suite. It provides two main functionality:

3.2.1 Tested Model interface

Wrapper for developed models, giving standardized outputs of predictions. This way our analysis can be further extended with new models.

3.2.2 ModelTestingSuite

The primary Test class evaluates specific models against a defined dataset. It supports evaluation based on chosen metrics or measures the total inference latency. The interface also supports batch processing, including random dataset splitting and the generation of result collections for each batch.

4 Initial Results

To demonstrate proof-of-concept, we utilized our Testing Suite to compare model accuracies (see Figure 10) and inference times (see Figure 11).

While these results align with our initial intuition, they serve as preliminary indicators rather than definitive proof.

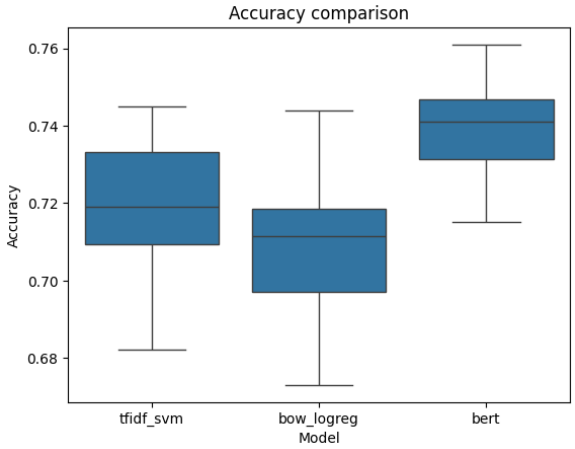

Figure 10: Initial comparion of accuracies

5 Future Steps

In the next phase of this research, we aim to expand our modeling ensemble by incorporating RoBERTa, alongside two classical machine learning baselines. To ensure a comprehensive evaluation, we will extend our testing suite with complementary metrics and conduct in-depth diagnostic analyses. These steps are designed to provide the empirical depth necessary to validate our core hypotheses.

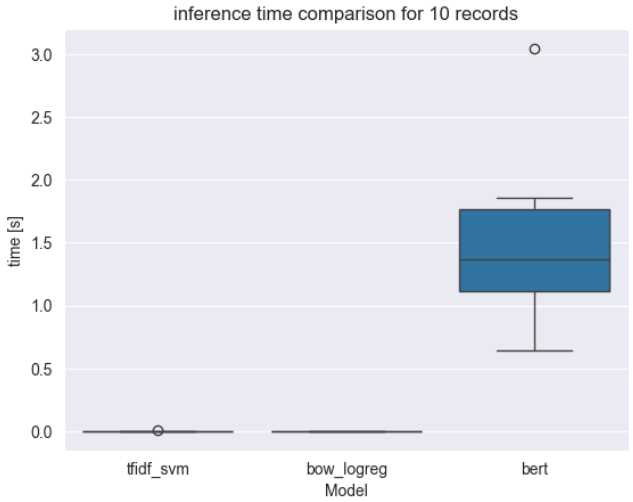

Figure 11: Initial comparison of inference time

References

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.

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*.

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.

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*.