

# Harnessing Large Language Models for organic spoiler detection and rephrasing in movie reviews

Mattia Angius (3130282), Giacomo Bellodi (3136190), Pierluigi Mancinelli (3120534),  
Federico Marinozzi (3136855), Leonardo Polverari (3122388)

Bocconi University  
Via Sarfatti, 25  
20136 Milan, Italy

## Abstract

Spoilers in reviews can damage the enjoyment of entertainment products. Therefore, review platforms focus on detecting and flagging them. Unfortunately, this practice censors potentially valuable reviews. Large language models (LLMs) may represent a powerful tool to overcome this problem. We implement multiple classifiers, we utilize LLMs to generate paraphrased, spoiler-free reviews and we propose a novel evaluation score, the **S5**, to assess their performance. After testing various models, the DistilBERT classifier emerges as the most effective, achieving a 68% macro F1 score. This performs well compared to state-of-the-art models and surpasses our established baseline. Moreover, the **S5** effectively evaluates the LLMs paraphrases, considering the spoiler removal. This organic approach can improve user interaction with platforms by automatically detecting spoilers and modifying reviews, preventing the reader from being harmed and the writer from feeling censored.

## 1 Introduction

Spoilers in movie reviews are disclosures of key plot elements that can diminish the audience’s viewing experience by revealing significant information prematurely. For instance, being aware of the main character’s death in advance could discourage a viewer from watching the entire movie. Review aggregation platforms prioritize spoiler detection to improve user experience. Unfortunately, this practice can lead to the unintended censorship of potentially valuable reviews, restricting insightful critiques even if containing only minor plot details. In this respect, subjectivity intrinsic in spoiler detection further complicates the picture. Something that a reader considers spoiler might just be an important part to understand a review for another. Large language models such as OpenAI’s GPT, have demonstrated significant capabilities in understanding and generating human-like text. There-

fore, they may represent a powerful tool to overcome this problem. Thanks to their skills, they can effectively identify the spoiler content and rephrase it, preserving its original meaning. While the limited existing literature focuses on the classification task (Wan et al., 2019; Bao et al., 2021), no study explores solutions involving paraphrases with LLMs. To fill this gap, we first implement multiple classifiers, using both transformer-based and Machine Learning algorithms. We train them using plain review texts. We also create custom models to combine text and other features, similar to the work of Zeng et al. (2024). Then, we utilize OpenAI and Meta’s LLMs to generate paraphrased, spoiler-free reviews. Finally, we propose a **novel evaluation score**, the **S5**, to measure their performance. This approach can improve user experience on review platforms by automatically detecting spoilers and modifying reviews, providing a protection for both readers and writers.

## 2 Experiments

We carry out two parallel experiments: spoiler classification and LLM review rephrasing. We test and evaluate both experiments on the same data.

### 2.1 Data Description

Our study utilizes data sourced from Kaggle originally scraped from **IMDb**, an online database containing information about movies, TV shows, and video games. There are two large datasets: the first includes information about 1572 movies, whereas the second comprises 573,906 reviews about these movies, published between 1998 and 2018. Each review’s label indicates whether the review contains spoilers: 74% do not contain spoilers, while 26% do. Finally, we scrape movie titles from IMDb website as they are not included in the original datasets.

Given the large size of the dataset, we employ sampling techniques to reduce the number of re-

views. We discard reviews from 1998 to 2004 because the balance of spoiler to non-spoiler reviews does not match the distribution in the overall dataset. To avoid biases in this sample, we perform t-tests on the continuous and categorical features of the dataset. We find out that even if p-values are significant, the magnitudes of differences are extremely low. Furthermore, we apply a stratified sampling technique, adjusting the number of reviews per movie and per spoiler status. This method ensures that the balance of reviews across both movie titles and spoiler status is maintained (74% non-spoiler, 26% spoiler), effectively reducing the dataset size to **117,216 reviews**.

The data preprocessing consists of expanding contractions, removing non-alphanumeric characters, stripping metadata elements like HTML tags and URLs, and eliminating extra spaces. Moreover, for the baseline model we convert all text to lower-case, we apply stemming to reduce words to their root form and we remove stopwords. For the other models we apply a pretrained tokenizer.

## 2.2 Model Description

We first employ a TF-IDF (2-6) characters on review texts, followed by a Logistic regression. This serves as the baseline for our project. Next, we present two alternative approaches: one using only plain text to make predictions, the other combining both text and other features. For the first we deploy a DistilBERT tokenizer and classifier. For the second we propose a new approach, inspired by related work (Zeng et al., 2024). Specifically, the meta-features we create are sentiment polarity and subjectivity, NER count, NER text, word and char length of each review, and the presence of keywords. For this novel application, we compare the two most successful models. One encodes the text with Google’s T5, then combines the resulting embeddings with the meta-features and finally passes the output to a Logistic Regression. The other uses DistilBERT tokenizer on text, then concatenates this output with the meta-features and classifies with a Neural Network. We also deploy and compare other models, such as XGBoost and BiDirectional GRU. See appendix A for their structures and parameters specifications.

For the rephrasing task, we use OpenAI APIs to call ‘gpt-3.5-turbo’ and HuggingFace Transformers to interact with ‘Llama-3B-Instruct’. We give both the same prompt in which we include movie

summary and synopsis as context and the review text as user message. For both we set temperature to 0.001, max\_tokens = 1024 and top\_p = 0.9. These parameters prevent the LLMs from being too creative and force them to answer to our specific prompt (available in Appendix B).

## 2.3 Experimental Setup

For each model we divide the dataset into train, evaluation and test splits (80-10-10). We use the first 2 to fit and evaluate the classifiers and the latter as fresh data to make predictions. For the rephrasing task we ask both LLMs to provide rephrased versions of the original reviews, removing spoiler content but keeping the same wording. We produce 80 paraphrases and we evaluate them using the metrics defined in the next section.

## 2.4 Evaluation Metrics

### 2.4.1 Classification evaluation

To evaluate the performance of the classifiers we employ precision, recall and macro F1 score. We focus on macro F1 to give more importance to the minority class. To calculate this F1, we prioritize recall.

### 2.4.2 Rephrasing evaluation

To assess the performance of rephrased reviews, we use two existing metrics, the METEOR score (Banerjee and Lavie, 2005) and the BERTScore (Zhang et al., 2020). We also modify the latter to create our own evaluation metric, the **Synopsis-Summary Semantic Similarity Score (S5)**.

The METEOR score evaluates text similarity based on unigram matching through stages of exact, stemmed, and synonym matches. It calculates unigram precision and recall, combined into an F1 score prioritizing recall. We compute the METEOR score for each rephrased review against the original to assess word overlap.

BERTScore overcomes the limitations of n-gram matching models by using contextual embeddings from pretrained models such as RoBERTa. It first calculates cosine similarity between token embeddings. Then, it incorporates precision, recall, and F1 score with additional importance weighting through inverse document frequency. BERTScore thus provides a valuable metric to assess whether each rephrased review contains the same fundamental concepts as the original one.

**S5 score** The dataset contains valuable information in the form of movie summaries and synopses.

While summaries describe the plot without spoilers, synopses do include them. We propose the S5 score by combining the contextual evaluation of the BERTScore with the embeddings of both synopses and summaries. For each paraphrased review, we retrieve its BERTScore with respect to the original review. We then compute an Adjustment factor by comparing the cosine similarity between the original and rephrased reviews with both the summaries and synopses.

$$S5 = BERTScore \times (1 + AdjFactor)$$

$$AdjFactor = \left( \frac{Osy - Rsy}{Osy} \right) + \left( \frac{Rsu - Osu}{Osu} \right)$$

In the Adjustment factor<sup>1</sup>, the first normalized difference assesses the deviation of the rephrased review from the synopsis, relative to the original review. A higher difference indicates a lower presence of spoiler-containing plot sections. The second normalized difference assesses the deviation of the rephrased review from the summary, relative to the original review. A higher difference thus indicates better retention of spoiler-free plot elements.

The S5 score adjusts the BERTScore between a rephrased review and its original version. It does so by considering the trade-off between maintaining the coherence of the review and removing spoiler content. This approach ensures that the resulting paraphrases retains the essence of the original review, but also minimizes the risk of revealing spoiler details. Indeed, BERTScore penalizes rephrased reviews due to the removal of spoilers. However, our goal is to eliminate spoiler content. Our score adjusts the BERTScore to account for the removal of spoilers, aligning it with our task for the LLMs.

### 3 Results

For the classification task (Table 1), the baseline model achieves surprisingly positive results. The only model capable of beating the baseline is the **DistilBERT** classifier on text only. This pre-trained model reaches **68% F1 score**, with a 3% gain over the previous. The other models do not achieve similar results, with BiDirectional GRU being the one

<sup>1</sup>*Osy* is the similarity between the original review and the synopsis; *Rsy* is the similarity between the rephrased review and the synopsis; *Rsu* is the similarity between the rephrased review and the summary; *Osu* is the similarity between the original review and the summary.

Table 1: Comparison of Macro F1 Scores on Test Set

Model	Plain Text	Text and Meta Features
TF-IDF Baseline	0.65	-
T5 Logistic	0.63	0.64
T5 XGB	0.58	0.58
DistilBERT	<b>0.68</b>	0.64
GRU	0.62	0.63

that best nears baseline performances. For the meta-features approach instead, none of the models beat the baseline. Finally, results are robust between dev and test set, with a maximum difference of 0.01 between the two.

For the rephrasing task we offer a comparative view of the summary statistics of three selected scores, namely BERTScore F1, METEOR score and our novel S5 score, and our Adjustment factor. We provide two tables, one with review paraphrases made by GPT 3.5 (Table 2) and the other by Llama3 (Table 3).

Table 2: Statistical summary of evaluation metrics for GPT 3.5

GPT 3.5				
	BERT F1	S5	Adj. Factor	METEOR
<b>mean</b>	0.873	0.874	0.001	0.260
<b>min</b>	0.830	0.813	-0.031	0.067
<b>25%</b>	0.850	0.847	-0.003	0.167
<b>50%</b>	0.865	0.870	0.0004	0.225
<b>75%</b>	0.888	0.891	0.005	0.306
<b>max</b>	0.991	0.990	0.043	0.950

Table 3: Statistical summary of evaluation metrics for Llama3

Llama3				
	BERT F1	S5	Adj. Factor	METEOR
<b>mean</b>	0.908	0.908	0.00006	0.521
<b>min</b>	0.794	0.795	-0.022	0.029
<b>25%</b>	0.878	0.875	-0.005	0.343
<b>50%</b>	0.909	0.913	-0.0003	0.493
<b>75%</b>	0.938	0.938	0.004	0.721
<b>max</b>	0.995	0.994	0.026	0.978

For the rephrasing task we offer a comparative view of the BERTScore F1, METEOR score and our novel S5 score on the review paraphrases made by GPT 3.5 (Table 2) and Llama3 (Table 3).

### 4 Discussion

In this section, we analyze the results of the previous two tables. First, we did not expect such a small difference in performance between the baseline and

more advanced models. Despite our DistilBERT model achieving a higher score than those in existing literature (collected by Zeng et al. in 2024), we note that none of the considered studies uses a simple TF-IDF model as a baseline. All models have equal or better results when using meta-features compared to plain text. The only one that decreases in performance is the DistilBERT. We hypothesize that this phenomenon happens because our hardware limitations prevent us from properly training such a complex model.

Given that reviewers provide spoiler flags themselves, we can deduce that subjective definitions of spoilers may lead to inaccurate evaluations. Therefore, a limitation of our classifier lies in the trustworthiness of labels.

For the rephrasing task, results reveal that Llama’s responses achieve a higher average BERTScore than OpenAI’s. However, the METEOR score of the former exceeds that of the latter by more than double on average. This suggests that Llama may be emulating the original reviews, thus failing to effectively remove spoilers. Such results are exemplified by our chosen test case, whose details are available in Appendix C.

Table 4: Test case review - metrics comparison between GPT 3.5 and Llama3

	GPT 3.5	Llama3
<b>BERTScore F1</b>	0.863	0.908
<b>S5</b>	0.868	0.908
<b>Adjustment Factor</b>	0.006	0.0001
<b>METEOR</b>	0.199	0.467
<b>Original Synopsis Score</b>	0.954	0.954
<b>Rephrase Synopsis Score</b>	0.942	0.949
<b>Original Summary Score</b>	0.906	0.906
<b>Rephrase Summary Score</b>	0.900	0.902

Our S5 is favoring OpenAI’s rephrased version. Its Adjustment factor is slightly higher than that of Llama’s. This discrepancy stems from the lower similarity of GPT’s rephrasing with the plot synopsis. Consequently, the S5 score effectively prioritizes spoiler removal and penalizes the LLM’s tendency to copy the original review without removing spoilers. The S5 score is primarily limited by the low magnitude of the Adjustment factor. Attempts to increase it, either linearly or exponentially, lead to adjusted scores surpassing BERTScore’s upper limit. Manual annotation could help us fine-tune a weighting strategy to increase the factor’s tangible effect.

## 5 Related Work

The literature regarding spoiler detection is quite limited. Early studies use topic modeling and n-gram counts through simple classifiers (Guo and Ramakrishnan, 2010; Jordan Boyd-Graber, 2014). More recent approaches involve LSTM or transformers (Bao et al., 2021). In our work, we propose a similar approach, adding text-related features. Other papers focus on using meta-features to enrich the models, adding network information (Zeng et al., 2024) or features related to the distribution of spoilers (Wan et al., 2019). However, we cannot apply this strategy since our task is to classify whole reviews instead of single sentences. As highlighted in this last paper, a current research gap regards the rephrasing of “spoiled” reviews, which is the focus of the second part of our work.

## 6 Conclusion

We propose a novel organic approach to spoiler detection that comprises two parallel experiments. We first implement multiple classifiers, using both transformer-based and Machine Learning algorithms. We train them using plain review texts. We also create custom models to combine text and other features. Secondly, we prompt OpenAI and Meta’s LLMs to rephrase the user reviews to remove spoilers. Finally, we propose a novel evaluation score for these paraphrases, the S5 score. Our results show that the DistilBERT classifier outperforms both our baseline and state-of-the-art models on simple text with 68% macro F1 Score. Contrarily, the novel approach that introduces meta-features is not as powerful and effective. We acknowledge the limitations of this creative approach, but we still believe in its promising aspects. Hence, we do expect the research community to delve into such a promising strategy to spoiler detection.

Moreover, our S5 offers a comprehensive view of the rephrasing task, considering the removal of spoiler-containing phrases. This allows to adjust the BERTScore evaluation, thus compensating for the LLM’s removal of spoiler content. Future research should also focus on using the S5 score to fine-tune Instruct language models. The maximization of such novel score would give more importance to a paraphrases that removes spoilers rather than merely emulating the original review. An improved version of our S5 score could incorporate proxies for the model creativity.



## References

- Satanjeev Banerjee and Alon Lavie. 2005. [Meteor: An automatic metric for mt evaluation with improved correlation with human judgments](#). pages 65–72.
- Allen Bao, Marshall Ho, and Saarthak Sangamnerkar. 2021. [Spoiler alert: Using natural language processing to detect spoilers in book reviews](#).
- Sheng Guo and Naren Ramakrishnan. 2010. Finding the storyteller: Automatic spoiler tagging using linguistic cues. Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010):412–420.
- Jackie Sauter Zajac Jordan Boyd-Graber, Kimberly Glasgow. 2014. [Spoiler alert: Machine learning approaches to detect social media posts with revelatory information](#). *Proceedings of the American Society for Information Science and Technology 5*, arXiv:1503.06733.
- Mengting Wan, Rishabh Misra, Ndapa 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.
- Shuo Yang, Wei-Lin Chiang, Lianmin Zheng, Joseph E. Gonzalez, and Ion Stoica. 2023. [Rethinking benchmark and contamination for language models with rephrased samples](#).
- Zinan Zeng, Sen Ye, Zijian Cai, Heng Wang, Yuhan Liu, Haokai Zhang, and Minnan Luo. 2024. [Mmoe: Robust spoiler detection with multi-modal information and domain-aware mixture-of-experts](#).
- Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. [Bertscore: Evaluating text generation with bert](#).

## A Appendix

Here we provide the parameters’ specifications for each of the models employed:

- TFIDF Baseline:
  - TFIDF: `n_gram = (2, 6)`; `min_df = 0.001`; `max_df = 0.8`; `analyzer = char`
  - Logistic: `class_weight = balanced`, `C = 0.5`
- Text-only Models:
  - T5-Logistic: `class_weight = balanced`, `max_iter = 5000`
  - T5-XGBoost: `n_estimators = 500`; `max_depth = 3`; `learning_rate = 0.1`, `subsample = 0.8`, `eval_metrics = logloss`, `early_stopping = 10`

- DistilBERT: `epochs = 3`, `batch_size_per_device(GPU) = 16`, `weight_decay = 0.01`, `gradient_accumulation_steps = 4`
- GRU: Structure: 1 BiDirectional GRU layer (128 nodes), 1 dropout layer (0.3), 1 simple GRU layer (64); 1 dropout layer (0.2); 1 feed-forward layer (64), `batch_size = 64`; `epochs = 10`; `early_stopping (patience = 2)`

### • Text + meta-features Models:

- T5-Logistic: `class_weight = balanced`, `max_iter = 5000`
- T5-XGBoost: `n_estimators = 500`; `max_depth = 3`; `learning_rate = 0.1`, `subsample = 0.8`, `eval_metrics = logloss`, `early_stopping = 10`
- Custom DistilBERT: `epochs = 5`, `batch_size_per_device(GPU) = 16`, `weight_decay = 0.01`, `gradient_accumulation_steps = 4`.  
**Structure:** 1 DistilBERT\_model layer on text. Then concatenate text with meta-features followed by a Linear(768+ addit\_feature\_count = 8 , 128), ReLU activation, Batch Normalization, dropout layer. To conclude, another Linear(128, 1)
- GRU: `batch_size = 64`; `epochs = 10`; `early_stopping (patience = 2)`

### **Structure:**

- \* Text: 1 embedding layer 1 BiDirectional GRU layer (128 nodes), 1 dropout layer (0.3), 1 simple GRU layer (64); 1 dropout layer (0.2); 1 feed-forward layer (64)
- \* Movie title: 1 embedding layer, 1 maxpooling
- \* Concatenate movie title, text and meta-features passed into fully connected NN with 2 hidden layers with ReLU activation and dropout and a sigmoid output layer.

## B Appendix

Guidelines on how to write an effective prompts were followed from the paper published by Yang et al. (2023)

**Prompt:**

*You are an assistant specializing in spoiler detection for reviews. A spoiler is any information about the plot of a TV series, film, or book that could ruin the enjoyment for someone who has not yet seen or read it. You have this task: Rephrase the review to remove any spoilers while maintaining the original wording. To identify spoilers, consider any plot elements mentioned in the plot synopsis but not in the summary. Plot summary: {summary}, Plot synopsis: {synopsis}*

## C Appendix

Here we provide the original review, as well as the two rephrased versions, for the test case mentioned above:

**Original review text:** Popular, but frustrated high school civics teacher Jim McAllister (an excellent performance by Matthew Broderick) locks horns with ruthlessly driven and ambitious over-achiever Tracy Flick (Reese Witherspoon in peak aggressively obnoxious form), who's determined to become president of Carver High's student body. McAllister convinces amiable dumb jock Paul Metzler (played with endearingly goofy charm by Chris Klein) to run for class president. Complications ensue when Paul's sassy lesbian sister Tammy (the adorable Jessica Campbell) decides to join the presidential race. Plus McAllister's personal life is starting to unravel. Director/co-writer Alexander Payne and co-writer Jim Taylor concoct a hilariously savage, cynical and unsentimental no-holds-barred satire on American society as a general whole which ingeniously uses high school as an apt microcosm of the world at large: we've got fiercely barbed commentary on such worthy topics as morals, politics, teen sexuality, marital infidelity, the lonely price of success, how all actions beget consequences that can change the rest of your life, jealousy, and blind ambition run totally amok. Kudos are also in order for the uniformly fine acting from a sound cast: Broderick and Witherspoon spar with each other beautifully well; they receive top-notch support from Klein, Campbell, Phil Reeves as ramrod principal Walt Hendricks, Molly Hagan as Jim's loving wife Diane, Delaney Driscoll as Jim's sweet and attractive lady best friend Linda Novotny, Mark Harelik as disgraced ex-teacher Dave Novotny, Colleen Camp as Tracy's supportive mother Judith R. Flick, and Frankie Ingrassia as

the saucy Lisa Flannagan. James Glennon's slick, agile cinematography makes neat occasional use of freeze frames and superimpositions. Rolfe Kent's jaunty full-bore orchestral score likewise does the trick. An absolute hoot.

**GPT 3.5 rephrased version:** In this sharp and witty high school comedy, dedicated civics teacher Jim McAllister clashes with the determined and ambitious student Tracy Flick, played brilliantly by Reese Witherspoon. McAllister persuades affable athlete Paul Metzler to run for class president, leading to unexpected complications when Paul's spirited sister Tammy also enters the race. Director Alexander Payne and writer Jim Taylor craft a biting satire that cleverly mirrors broader societal issues through the lens of high school dynamics. The film skillfully tackles themes like ethics, politics, relationships, and the consequences of one's actions. The stellar cast, including standout performances from Matthew Broderick, Chris Klein, and Jessica Campbell, delivers a solid portrayal of the characters. With engaging cinematography and a lively score, this film is a delightful and thought-provoking watch

**Llama3 rephrased version:** Popular, but frustrated high school civics teacher Jim McAllister (an excellent performance by Matthew Broderick) clashes with a driven and ambitious student, Tracy Flick (Reese Witherspoon in peak form), who's determined to become president of Carver High's student body. McAllister convinces a likable student, Paul Metzler (played with endearingly goofy charm by Chris Klein), to run for class president. Complications arise when Paul's sister Tammy (the adorable Jessica Campbell) decides to join the presidential race. Meanwhile, McAllister's personal life begins to unravel. Director/co-writer Alexander Payne and co-writer Jim Taylor craft a hilariously savage, cynical, and unsentimental satire on American society, using high school as a microcosm of the world at large. The film tackles a range of topics, including morals, politics, teen sexuality, and the consequences of one's actions. The cast delivers uniformly fine performances, with Broderick and Witherspoon engaging in a delightful sparring match. The film features slick cinematography, a jaunty score, and is an absolute hoot.