Enhancing the Performance of Spoiler Review Detection by a LLM with Hints

Genta Nishi
Graduate School of Information
Science and Electrical Engineering,
Kyushu University, Japan
nishi.genta.985@s.kyushu-u.ac.jp

Einoshin Suzuki
Faculty of Information
Science and Electrical Engineering,
Kyushu University, Japan
suzuki@inf.kyushu-u.ac.jp

Abstract

We investigate the effects of three hints including an introduction text, a few examples, and prompting techniques to enhance the performance of a Large-Language Model (LLM) in detecting a spoiler review of a movie. Detecting a spoiler review of a movie represents an important Natural Language Processing (NLP) task which resists the Deep Learning (DL) approach due to its highly subjective nature and scarcity in data. The highly subjective nature is also the main reason of the poor performance of LLMs-based methods, which explains their scarcity for the target problem. We address this problem by providing the LLM with an introduction text of the movie and a few reviews with their class labels as well as equipping it with prompts that select and exploit spoiler types with reasoning. Experiments using 400 manually labeled reviews and about 3200 LLM-labeled reviews show that our CAST (Clue And Select Types prompting) outperforms (0.05 higher) or is on par with (only 0.01 lower) cutting-edge LLM-based methods in three out of four movies in ROC-AUC. We believe our study represents an evidence of a target problem in which the knowledge intensive approach outperforms the learning-based approach.

1 Introduction

According to the Oxford Learner's Dictionaries, a spoiler is defined as "information that you are given about what is going to happen in a film, television series, etc. before it is shown to the public"[1], which can hinder or stop consumers' enjoyment of a work (Tsang and Yan, 2009). In this paper, we focus our attention to spoiler reviews of a movie due to their complex nature for NLP and their high influence on our daily life. Manually setting mute words (Golbeck, 2012) , e.g., the true criminal, or spoiler tags, though effective, are expensive due to the necessary human labor. NLP-based automatic detection could be a realistic solution depending on its accuracy and cost.

Since movies are rich in variety and so are their reviews, detecting a spoiler review is a highly subjective task. Moreover, Guo and Ramakrishnan (2010) pointed out that constructing a large-scale dataset with high-quality labels is difficult for spoiler detection. These two reasons rule out DL-based methods from consideration, even if they have been quite successful in various NLP tasks. LLMs could be considered as the state-of-the-art solutions of the knowledge intensive approach due to their high capabilities in various tasks and their low costs in development. However, Zhang et al. (2025b) pointed out that their text classification capabilities are limited and the development has been slow, which we believe the reason for their scarcity in the spoiler detection domain.

In this paper, we investigate three kinds of hints to enhance the performance of spoiler review detection by an LLM. The first hint is an introduction text, which corresponds to domain knowledge in the knowledge intensive approach. The second hint is a few reviews with their binary class labels, i.e., spoiler or not spoiler, which can be regarded as examples for few-shot learning. The third hint is spoiler types with a reasoning strategy, which could be viewed as an inference strategy on subclasses for the LLM. Broadly speaking, exploiting these three kinds of hints belongs to the widely-used prompt engineering, though our motivation is to obtain an evidence which suggests in the long run the characteristics and the conditions of the target problems in which the knowledge intensive approach outperforms the learning-based approach.

Figure 1 shows two working examples of our CAST on the movie "Hulk", in which Bruce transforms himself to a green heroic monster. The first

[1] https://www.oxfordlearnersdictionaries.com/definition/english/spoiler?q=spoiler

Proceedings of the First Workshop on Comparative Performance Evaluation: From Rules to Language Models associated with RANLP 2025, pages 97–112, Varna, Bulgaria, Sep 11, 2025.
https://doi.org/10.26615/978-954-452-102-8-011

SPOILER
input: I thought endowing Banner's father with the Absorbing Man's powers was a brilliant idea, symbolizing what his father indirectly did to Bruce his whole life.
clue: "Absorbing Man's powers", "father indirectly did to Bruce his whole life".
spoiler type: true identity, character features, development of the story, past, problem occurs.

→ **spoiler level**: 0.9999

NOT SPOILER
input: Seeing the green behometh smash up tanks, helicopters etc had me in aweof the amazing folks who created the cgi.
clue: "seeing the green behometh", "amazing folks".
spoiler types: appearance, development of the story, true identity, past, status/power.

→ **spoiler level**: 0.1559

Figure 1: Examples of spoiler and non-spoiler reviews for "Hulk". Spoiler levels are provided by our CAST. The spoiler review mentions the identity of the final villain. The non-spoiler review mentions unimportant details.

review reveals the identity of the final villain, who gave the power to Bruce and is thus a spoiler. CAST correctly estimates its spoiler level to 0.9999 by selecting four spoiler types, of which the red two are correct, with its LLM. CAST also explains two clues in its decision, which demonstrates its comprehensibility to the users. The second review, on the other hand, just explains the widely-known capabilities of Hulk, and is thus not a spoiler. CAST correctly estimates its spoiler level again.

2 Related Work

2.1 Spoiler Detection

Spoiler Detection methods can be classified into classification-based, clues-based, and LLMs-based. The first approach exploits the powerful capabilities of the text classification methods. Wan et al. (2019) proposed SpoilerNet, which inputs review documents and item specificity information to Hierarchical Attention Network (Yang et al., 2016). Chang et al. (2021) proposed SDGNN, which combines a Graph Neural Network (Marcheggiani and Titov, 2017) that recognizes sentence dependencies with a genre aware structure. We consider this approach is inadequate for our target problem due to the lack of large-scale data and the variety in movies and their reviews.

The second approach uses multiple frameworks to extract features from various kinds of clues including user data, movie metadata, and reviews. The features could be passed to a Mixture of Experts for each genre (Zeng et al., 2024; Zhang et al., 2025a). This approach is relevant to our CAST, though the former doesn't use an LLM for the main purpose of spoiler detection.

The last approach is rare in spoiler detection, possibly due to the limited capability of LLMs in text classification (Zhang et al., 2025b). As we explained in the previous section, we try to enhance the performance of this approach by using three kinds of hints, which are not limited to the data source. Since LLMs have achieved notable successes in handling semantics (Schaeffer et al., 2025), we believe they are also promising for our target task.

2.2 Text Classification by LLM

Text classification using LLMs can be broadly classified into two approaches, i.e., the approach that relies on fine-tuning and the one on few-shot learning.

As an example of the former, Zhang et al. (2025b) proposed RGPT, of which fine-tuning is based on the idea of Adaptive Boosting (Freund and Schapire, 1997). Their fine-tuning is conducted in multiple rounds, each of which updates the weight distribution over the dataset based on the predictions of the weak learner induced in the round. The final prediction is based on weak learners with their model weights obtained according to the predictions.

As an example of the latter, Sun et al. (2023) proposed Clue and Reasoning Prompting (CARP). They pointed out that LLM-based methods are inferior to fine-tuned models in text classification tasks due to the lack of inference ability and token length limitations in the former. CARP encourages users to find clues such as keywords, tone, semantic relations, and references from the text before reasoning, which strengthen its reasoning ability. They succeeded in conducting few-shot learning by sampling a few examples with the k-nearest neighbor method and developed a voting method among LLMs with various outputs. CARP outperforms a powerful prompt engineering method Zero-shot-Chain-of-Thought[2] in text classification performance (Kojima et al., 2022).

In the datasets used for spoiler detection, the labels are typically collected from review sites[3]. It has been pointed out that their quality is low due to differences in spoiler standards between human labelers and their mistakes, e.g., they occasionally forget to add spoiler tags (Guo and Ramakrishnan, 2010). In other words, models trained on these datasets are likely to exhibit low accuracy. Therefore, we focus our attention to the second approach.

3 Target Problem

As we stated, our target problem is spoiler review detection of a movie. We could have formalized it as a classification problem by setting a binary class label of spoiler and not spoiler as our output. This formalization allows us to use accuracy as the evaluation measure, which is easy to understand, but necessitates a threshold that separates the two classes. Setting an appropriate threshold is possible when the misclassification costs, i.e., the cost of a false positive and that of a false negative, are known (Han et al., 2011), which is not the case for us. We therefore formalized the target problem as an estimation problem of the spoiler level of a movie review from 0 to 1, the higher being more likely to be a spoiler. As we will explain later, ROC-AUC (Han et al., 2011) is adopted as our evaluation measure.

The input to our target problem consists of set $\{R_{i,1}, \ldots, R_{i,n(i)}\}$ of review texts and introduction text I_i for movie i, where $R_{i,j}$ and $n(i)$ represent the j-th review and the number of reviews, respectively. The output of our target problem is spoiler levels $(Y_{i,1}, \ldots, Y_{i,n(i)})$, where $Y_{i,j}$ represents the spoiler level of $R_{i,j}$.

Since the reviews can be sorted in descending order based on their spoiler levels, we can compute ROC-AUC of an output, which we adopt as our evaluation measure (Han et al., 2011). We assume that the class label of $R_{i,j}$ is available in everything in the output. ROC-AUC corresponds to the probability that a positive example, i.e., a spoiler review in our case, is ranked higher than a negative example, i.e., a non spoiler. ROC-AUC is widely adopted in detection problems where the misclassification costs are unknown.

4 Proposed Method: CAST

4.1 Overview

Algorithm 1 CAST

```
Input: set $\{R_{i,1}, \ldots, R_{i,n(i)}\}$ of review documents and introduction text $I_i$ of movie $i$.
Output: spoiler levels $(Y_{i,1}, \ldots, Y_{i,n(i)})$
  for $j = 1$ to $n(i)$ do
    for each sentence $r_{i,j,k}$ in $R_{i,j}$ do
      // Construct $prompt_{i,j,k}$.
      $c_{i,j,k} = CLUE(r_{i,j,k})$
      $t_{i,j,k} = SelectType(r_{i,j,k}, c_{i,j,k})$
      $prompt_{i,j,k}$
        $= BasePrompt(I_i, r_{i,j,k}, c_{i,j,k}, t_{i,j,k})$
      // Estimate $P_{\text{ANSWER}}$ using LLM.
      $P_{\text{SPOILER}} = P(\text{“ SP”}|prompt_{i,j,k})$
      $P_{\text{NOT SPOILER}} = P(\text{“ NOT”}|prompt_{i,j,k})$
      // Compute the spoiler level.
      $y_{i,j,k} = \dfrac{e^{P_{\text{SPOILER}}}}{e^{P_{\text{SPOILER}}} + e^{P_{\text{NOT SPOILER}}}}$
    end for
    $Y_{i,j} = \max_k y_{i,j,k}$
  end for
  $\mathcal{Y}_i = (Y_{i,1}, \ldots, Y_{i,n(i)})$
  return $\mathcal{Y}_i$
```

$BasePrompt(I_i, r_{i,j,k}, c_{i,j,k}, t_{i,j,k})$ is shown below.

> This is a Spoiler Detection for input movie reviews.
> “Spoilers” is a description of a significant plot point or other aspect of a movie, which if previously known may spoil a person’s first experience of the work.
> A significant plot point is one that cannot be predicted from the film’s introduction or early developments.
> List CLUES (i.e., keywords, phrases, contextual information, semantic meaning, semantic relationships, tones, references) that support the spoiler

[2] Kojima et al. (2022) succeeded in stabilizing the zero-shot performance of LLMs and improving performance even in the few-shot case by using the CoT (Chain-of-Thought) approach.

[3] IMDb (https://www.imdb.com/) and Goodreads (https://www.goodreads.com/)

detection of the input.
Finally, based on introduction, clues, spoiler types, and the input, categorize the overall ANSWER of input as SPOILER or NOT SPOILER.

introduction: [example introduction 1]
review: [example review 1]
clue: [example clues 1]
spoiler types: [example types 1]
answer: [example answer 1]
...(7 few-shot examples follow.)

introduction: I_i
review: $r_{i.j.k}$
clue: $c_{i,j,k}$
spoiler types: $t_{i,j,k}$
answer:

We propose CAST (Clue And Select Types prompting), a spoiler detection method using an LLM. As shown later in Figure 2, CARP is weak against roundabout expressions, which are common in spoiler detection. We attribute this reason to the fact that such expressions "confuse" the LLM's judgment. Therefore, in CAST, the LLM is dynamically given spoiler types as hints for the judgment.

First, clues $CLUE(r_{i,j,k})$ are extracted from the input review $r_{i,j,k}$ according to Sun et al. (2023), where $r_{i,j,k}$ represents the k-th sentence of $R_{i,j}$. Then, the LLM is given most of $BasePrompt$, from the beginning to the second "clue:" so that it outputs phrases that are clues for spoiler detection. We show the example reviews in Tables 10, 11, and 12. Next, the LLM is given all $BasePrompt$, which includes the output above as $CLUE(r_{i,j,k})$ and the spoiler types obtained with $SelectType(_{i,j,k}, CLUE(r_{i,j,k}))$, which we will explain in the next Sections.

Next, based on the input, clues, and spoiler types, LLM outputs a probability distribution of the following tokens: "SPOILER", "NOT SPOILER", and other words[4]. From the distribution, we calculate the probability P_{SPOILER} that the LLM outputs "SPOILER" and the probability $P_{\text{NOT SPOILER}}$ that it outputs "NOT SPOILER"[5]. Finally, we calculate the spoiler level $y_{i,j,k}$ using the softmax function to eliminate the probability of other words, i.e., the probabilities of "SP" and "NOT" sum up to 1. $Y_{i,j}$ is the maximum value of $y_{i,j,k}$ in terms of k, as we think the sentence that is most likely to be a spoiler determines the spoiler level of the review.

character relationships	true identity
character features	life or death
victory or defeat	purpose
problem occurs	trick
development of the story	past
status/power	appearance

Table 1: Spoiler types defined by Tajima and Nakamura (2015).

To provide diverse input for the LLM, we used eight few-shot examples that covered a range of review types (a direct spoiler review, an indirect spoiler review, an impression-only review, and a review with unimportant content). In addition, these examples were drawn from movies across various genres to develop a method applicable to multiple domains.

4.2 Selecting Spoiler Types

Since only one or a few spoiler types are relevant to a review, inputing all 12 types to the LLM would degrade the performance. We thus propose to select relevant spoiler types using the LLM using the following prompt.

Please select k spoiler types that are most appropriate for the review and its keywords from the following spoiler types.

spoiler type: [all types]
review: [review]
keywords: [clues]
appropriate type:

In CAST, we use the spoiler types classified by Tajima and Nakamura (2015). They collected 1370 spoilers from over 100 students and manually classified them into 12 types without any excess or deficiency. We show them in Table 1.

[4] We can obtain the distribution by setting the variable *logprobs* to True in llama-cpp-python (`https://github.com/abetlen/llama-cpp-python`).

[5] To be precise, we use "SP" and "NOT" instead of "SPOILER" and "NOT SPOILER", respectively, as the last two are not included in the vocabulary of Llama. These replacements are justified because the probabilities of "ILER" and "SPOILER" right after "SP" and "NOT" are almost 1, respectively.

5 Experiments

5.1 Conditions

As datasets, Kaggle (Misra, 2022) and LCS (Wang et al., 2023) are often used in recent spoiler detection studies (Zeng et al., 2024; Zhang et al., 2025a). However, several papers argue that their labels are not accurate due to their human labelers, whose spoiler standards are not uniform (Guo and Ramakrishnan, 2010; Wan et al., 2019). We conduct experiments on the IMDb dataset (Misra, 2022), which one annotator relabeled manually[6]. We also conducted the relabeling with an LLM. In our relabeling, we define a spoiler review as a review that includes an important event shown in Table 7 in the Appendix. The target movies and data sizes that we used in our experiments are shown in Table 2. The introduction texts were taken from the IMDb movie page. We show them in Table 9.

In the relabeling with an LLM, we adopted the following prompt.

```
This is to determine whether a review contains spoilers.
“Spoilers” is a description of a significant plot point or other aspect of a movie, which if previously known may spoil a person’s first experience of the work.
A significant plot point is one that cannot be predicted from the film’s introduction or early developments.
We will give you the title and significant plots of the movie, so please use that to determine whether the review contains spoilers.

title: [title]
significant plots: [events]
review: [review]
label (True or False):
```

Here, [events] is the same as the event shown in Table 7. In the relabeling, we used Llama3.1-8B (Dubey et al., 2024). Table 3 shows the ratios of the modified labels in our relabeling.

5.2 Baseline Methods

We employed Zero-Plus-Few-shot-Chain-of-Thought (CoT) (Kojima et al., 2022) and CARP (Sun et al., 2023) as the baseline methods. Although CoT is not a method developed for text classification, we use it as a baseline following Sun et al. (Sun et al., 2023). As we have introduced, CARP is a method for text classification by LLM. To keep the setting fair, we did not employ its voting method. We also tested variants of these methods by omitting their reasoning process and/or by employing the introduction text. Prompts of the methods are shown in the Appendix. In comparing CAST with the baseline methods, we used Llama2-13B (Touvron et al., 2023) implemented in llama-cpp-python[7] as the backbone of the LLM. In this experiment, we used Human labels. CAST and CARP were also compared in experiments using LLM labels with Llama2-13B, as well as in experiments using human labels on more recent LLM platform, Llama3.1-8B (Dubey et al., 2024). We adopted +i-r as the condition due to its overall, superior performance in the latter.

5.3 Few-shot Learning

Few-shot leaning is performed to standardize the answer format and improve accuracy. Two reviews (positive and negative) were collected from each of four movies (“Million Dollar Baby”, “The Fast and the Furious”, “Groundhog Day”, and “Match Point”) in the IMDb dataset (Misra, 2022). To be fair, the same examples were used by all methods[8]. The examples are shown in Tables 10, 11 and 12.

5.4 Results

The results are shown in Table 4. We first focus on the results on the datasets relabeled by a human, which are considered more accurate those with the LLM. For “Hulk” and “The Shawshank Redemption”, CAST is the best method. For “Mean Girls”, it is the third best performing method, quite close to the second one. For “Blood Diamond”, it is the second best performing method overall and the best performing method for “+intro -reasoning”. Overall, we conclude that CAST is the best method for few-shot spoiler detection based on human values. We then focus on the results on the LLM relabeled dataset. Compared to CARP, it performs worse on “Hulk” but slightly better on the other three movies. A detailed analysis is provided in Section 6.

[6] We admit the weakness of adopting a single annotator as the quality is affected by his subjectivity.

[7] abetlen/llama-cpp-python. https://github.com/abetlen/llama-cpp-python

[8] The presence or absence of spoiler types or introduction text is adjusted to match the method. The sampling method of CARP was skipped as the examples were given.

	Hulk	The Shawshank Redemption	Mean Girls	Blood Diamond
Human	100	100	100	100
-spoiler	31	43	32	25
-not spoiler	69	57	68	75
LLM	523	1737	445	628
-spoiler	41	148	84	117
-not spoiler	482	1589	361	511

Table 2: Target movies and data size of the dataset.

	Human	LLM
Hulk	36%	83.2%
The Shawshank Redemption	43%	49.6%
Mean Girls	33%	73.7%
Blood Diamond	25%	74.4%

Table 3: Ratios of modified labels in our relabeling.

6 Detailed Analysis

6.1 Case Study

To investigate how CAST detects spoilers, we analyze the example of "Blood Diamond" in comparison with CARP. The following contains spoilers for "Blood Diamond". In the final scene, the main character (Leonardo DiCaprio) dies. This content is clearly a spoiler. The review of Figure 2 includes this content, but describes it in a roundabout way ("I was hoping Leo would not die"). CARP is affected by the roundabout expression and shows poor performance, i.e., the spoiler level of about 0.78 is moderately high. On the other hand, in CAST, we can see that the spoiler types, e.g., "life or death", selected dynamically lead the LLM to output a very high spoiler level of about 0.97.

6.2 Useful Issues

We present several issues that we noticed in the experiments, which could contribute to our future research.

6.2.1 LLM is sensitive to cruel scenes

LLM over-identifies scenes involving injury or death as spoilers. Usually, commenting on an injury of a sub-character, especially toward the beginning of the movie, is not a spoiler. This movie, "Blood Diamond", is set in a war zone and thus contains many violent scenes, which are related to the spoiler type "life or death". Not only CAST but also CARP and CoT are subject to this kind of false positives, as they all employ LLMs. A

CARP
input: I was hoping Leo would not die I really wanted him to get out of Africa, but Zwick isn't about happy endings which i admire.
clue: "Leo", "die", "Zwick isn't about happy endings".

→ **spoiler level**: 0.7886 △

CAST
input: I was hoping Leo would not die I really wanted him to get out of Africa, but Zwick isn't about happy endings which i admire.
clue: "Leo", "die", "Africa".
spoiler types: life or death, true identity, development of the story, victory or defeat, problem occurs.

→ **spoiler level**: 0.9729 ○

Figure 2: Example of spoiler detection in a review of "Blood Diamond".

		Hulk	The Shawshank Redemption	Mean Girls	Blood Diamond
			Human Relabel		
Llama2-13B	CoT	0.7447	0.7209	0.8162	0.7072
	+i	0.7176	0,7187	0.8580	**0.7211**
	-r	0.6397	0.7340	0.7762	0.6450
	+i -r	0.7218	0.7546	**0.8736**	0.5691
	CARP	0.7433	0.7623	0.8350	0.6651
	-f	0.6840	**0.8209**	0.6719	0.6347
	+i	0.7087	0.7325	0.8244	0.6705
	-r	0.7555	0.7475	0.7992	0.6373
	+i -r	0.7129	0.7823	0.8534	0.5968
	CAST	0.7685	0.7638	0.8208	0.7056
	-f	0.7162	0.7825	0.7849	0.5696
	+i	**0.7761**	0.7813	0.8603	0.6863
Llama3.1-8B	CARP +i -r	0.7232	0.7987	0.8470	0.7109
	CAST +i	**0.7377**	**0.8184**	**0.8732**	**0.7701**
			LLM Relabel		
Llama2-13B	CARP +i -r	**0.8505**	0.8149	0.8272	0.7283
	CAST +i	0.8219	**0.8467**	**0.8273**	**0.7395**

Table 4: ROC-AUC of the spoiler levels of each methods for four movies from the IMDb dataset (Misra, 2022). “-f” represents a case that the prompt contains no few-shot example. “+i” represents a case that the prompt contains an introduction of the movie. “-r” represents a case without reasoning, which corresponds to our CAST. “+r” represents a case with reasoning, of which details will be explained in Section 7.2. The highest value for each film is highlighted in bold fonts, the second highest in underlined.

	Hulk	The Shawshank Redemption	Mean Girls	Blood Diamond
AllType	0.7602	0.7772	0.8695	0.6768
Embedding	0.7662	0.7764	0.8355	0.6645
LLM($k = 1$)	0.7017	0.7919	0.8125	0.6864
LLM($k = 3$)	0.7639	0.7597	0.8566	0.6704
LLM($k = 5$)	0.7761	0.7813	0.8603	0.6863

Table 5: ROC-AUC for each spoiler type selection method.

	Hulk	The Shawshank Redemption	Mean Girls	Blood Diamond
CARP +reasoning	0.7087	0.7325	0.8244	0.6705
CARP -reasoning	0.7120	0.7746	0.8621	0.6277
CAST +reasoning	0.7139	0.7195	0.8235	0.7072
CAST -reasoning	0.7761	0.7813	0.8603	0.6863

Table 6: ROC-AUC for each method with and without reasoning.

input: The rebels make a speech and then cut some kids arm off, then there ready to do the same to Solomon, but the rebel leader decides to spare him and take him as a prisoner and use him as a worker, then the movie continues on from there.
clue: "cut some kids arm off", "spare him and take him as a prisoner".
spoiler type: life or death, problem occurs, development of the story, past, status/power.

→ **spoiler level**: 0.9997

Figure 3: Example which shows LLM is sensitive to cruel scenes.

possible solution would be to strengthen the movie introduction to discourage LLM from reacting to the early scenes, or to use the synopsis included in the IMDb dataset (Misra, 2022) to convince LLM that the scenes are not important.

6.2.2 Spoiler type can increase false positive

Although spoiler types provide evidence of spoilers and contribute to lowering the false negative rate (Figure 2), they may also help to judge a non spoiler example as a spoiler. We show an example in Figure 4, which includes known descriptions of the main character in "Hulk". Unlike CARP (-reasoning) which appropriately gives a low spoiler level (0.1556), our CAST gave a high spoiler level due to the spoiler types "past" and "character features". Possible solutions include explaining in the prompt that some reviews may not be spoilers even if they match the spoiler type, or setting types also to the class of not spoiler. We suspect that there are about three such types but believe we need more evidence for further investigation.

input: Young Bruce grows up in an adopted family, never knowing what happened to his birth parents nor that he may be carrying abnormal genes as a result of his father's work.
clue: "Young Bruce", "never knowing", "abnormal genes".
spoiler type: past, character features, true identity, development of the story, problem occurs.

→ **spoiler level**: 0.9430

Figure 4: Example which shows spoiler types lead to an excessive spoiler level.

7 Ablation Study

7.1 Methods for Selecting Types

We evaluate the effect of our spoiler type selection in Section 4.1, which we call LLM. Here, we set the number of choices $k = 1, 3, 5$ and use Llama2 (Touvron et al., 2023) as our LLM.

As an alternative, we introduce another method which we call Embedding. Following the Dense Passage Retriever (DPR) (Karpukhin et al., 2020), we selected the spoiler type by the cosine similarity between the embedding vectors of the clues and the spoiler type. The embedding model is RoBERTa (Liu et al., 2019), which is fine-tuned on the spoiler domain dataset (Wan et al., 2019) and its paraphrases by Llama2-13B. The loss function is the same as DPR.

As a baseline method, we also compare AllType that does not select spoiler types and uses all of them. The experimental settings are based on Section 5. We use the human-relabeled datasets. The results are shown in Table 5. Overall, we conclude that the results of LLM ($k = 5$) is the best. Furthermore, an example is shown in Figure 5. This example is about "the death of a person", but Embedding cannot select "life or death", resulting in a false negative. In contrast, LLM is able to cor-

Embedding
input: I was hoping Leo would not die I really wanted him to get out of Africa, but Zwick isn' t about happy endings which i admire.
clue: "Leo", "die", "Africa".
spoiler type: problem occurs, past, trick, character features, appearance, character relationships.

→ **spoiler level**: 0.5457

LLM ($k = 5$)
...
spoiler type: life or death, true identity, development of the story, victory or defeat, problem occurs.

→ **spoiler level**: 0.9729

Figure 5: Comparison of Select type methods: Embedding and LLM ($k = 5$).

rectly select the spoiler type, and the output is also correct. We used in the main experiments LLM ($k = 5$) as our selection method.

7.2 Effect of Reasoning

Though reasoning is said to enhance the performance of LLM (Wei et al., 2022; Kojima et al., 2022; Sun et al., 2023), several researchers argue against it. Chen et al. (2024) point out several cases in which reasoning increases the probability of an incorrect output in text classification. Therefore, we investigate the effect of reasoning in CARP and CAST. We write "+reasoning" and "-reasoning" for with and without reasoning, respectively. We give their prompts at the end of the Appendix. The experimental settings are based on Section 5. We use the human-relabeled datasets.

The results are shown in Table 6. In both CARP and CAST, "-reasoning" performs better. In fact, there are almost no case where the correct answer is obtained through reasoning. We conclude that reasoning is unnecessary for our spoiler detection.

8 Conclusion

We show that in the field of spoiler detection, where there is a lack of high-quality datasets, adding three kinds of hints improves the performance of LLM-based spoiler review detection of a movie. It is no wonder that the introduction text and the few examples for few-shot learning are useful, as they represent typical domain knowledge and representative cases. The types that we used represent subclasses of the positive class. Our prompts instructs their effective selection and usage, which could be also useful in other text classification problems.

Our future research includes defining better spoiler types as well as setting non-spoiler types. Such types or sub-classes could be set dynamically according to the given reviews, the introduction text, and the few examples for few-shot learning. Utilizing other kinds of additional data such as synopses would deepen our understanding on the target domain and the prompt engineering.

References

Buru Chang, Inggeol Lee, Hyunjae Kim, and Jaewoo Kang. 2021. "Killing Me" Is Not a Spoiler: Spoiler Detection Model using Graph Neural Networks with Dependency Relation-Aware Attention Mechanism. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics*, pages 3613–3617.

Zhikai Chen, Haitao Mao, Hang Li, Wei Jin, Hongzhi Wen, Xiaochi Wei, Shuaiqiang Wang, Dawei Yin, Wenqi Fan, Hui Liu, et al. 2024. Exploring the Potential of Large Language Models (LLMs) in Learning on Graphs. *ACM SIGKDD Explorations Newsletter*, 25(2):42–61.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv e-prints*, pages arXiv–2407.

Yoav Freund and Robert E Schapire. 1997. A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting. *Journal of Computer and System Sciences*, 55(1):119–139.

Jennifer Golbeck. 2012. The Twitter Mute Button: a Web Filtering Challenge. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 2755–2758.

Sheng Guo and Naren Ramakrishnan. 2010. Finding the Storyteller: Automatic Spoiler Tagging using Linguistic Cues. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 412–420.

Jiawei Han, Micheline Kamber, and Jian Pei. 2011. *Data Mining: Concepts and Techniques, 3rd ed.* Morgan Kaufmann.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick SH Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-Tau Yih. 2020. Dense Passage Retrieval for

Open-Domain Question Answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP 2020)*, pages 6769–6781.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large Language Models are Zero-Shot Reasoners. *Advances in Neural Information Processing Systems*, 35:22199–22213.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. arXiv: 2403.05265.

Diego Marcheggiani and Ivan Titov. 2017. Encoding Sentences with Graph Convolutional Networks for Semantic Role Labeling. arXiv: 1703.04826.

Rishabh Misra. 2022. IMDb Spoiler Dataset. arXiv: 2212.06034.

Rylan Schaeffer, Punit Singh Koura, Binh Tang, Ranjan Subramanian, Aaditya K Singh, Todor Mihaylov, Prajjwal Bhargava, Lovish Madaan, Niladri S Chatterji, Vedanuj Goswami, et al. 2025. Correlating and Predicting Human Evaluations of Language Models from Natural Language Processing Benchmarks. arXiv: 2502.18339.

Xiaofei Sun, Xiaoya Li, Jiwei Li, Fei Wu, Shangwei Guo, Tianwei Zhang, and Guoyin Wang. 2023. Text Classification via Large Language Models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 8990–9005.

Kazuki Tajima and Satoshi Nakamura. 2015. A Study on Story Spoilers and Considering the Possibility to Detect Spoilers. *IPSJ SIG Technical Report on Groupware and Network Services (GN)*, 2015(7):1–6. (in Japanese).

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open Foundation and Fine-Tuned Chat Models. arXiv: 2307.09288.

Alex S. L. Tsang and Dengfeng Yan. 2009. Reducing the spoiler effect in experiential consumption. pages 708–709.

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.

Heng Wang, Wenqian Zhang, Yuyang Bai, Zhaoxuan Tan, Shangbin Feng, Qinghua Zheng, and Minnan Luo. 2023. Detecting Spoilers in Movie Reviews with External Movie Knowledge and User Networks. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 16035–16050.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. *Advances in Neural Information Processing Systems*, 35:24824–24837.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical Attention Networks for Document Classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1480–1489.

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. arXiv: 2403.05265.

Haokai Zhang, Shengtao Zhang, Zijian Cai, Heng Wang, Ruixuan Zhu, Zinan Zeng, and Minnan Luo. 2025a. Unveiling the Hidden: Movie Genre and User Bias in Spoiler Detection. arXiv: 2504.17834.

Yazhou Zhang, Mengyao Wang, Qiuchi Li, Prayag Tiwari, and Jing Qin. 2025b. Pushing the limit of LLM capacity for text classification. In *Companion Proceedings of the ACM on Web Conference 2025*, pages 1524–1528.

A Details of the Experiments

We show important events of each movie in Table 7. Each sentence is taken from plot_synopsis of IMDb_movie_details in the IMDb Dataset (Misra, 2022). We show the prompts of baseline methods in Table 8. We show the introduction text of each movie in Table 9. Each text is taken from the movie's IMDb page.

We show the reviews, the clues, the spoiler types, the reasonings, and the answers used in the few-shot learning in Tables 10 and 11.

We also show the introduction texts in the few-shot learning in Table 12.

We show the prompts with and without reasoning for CARP and CAST.

CARP

prompt of + reasoning

> This is a Spoiler Detection for input movie reviews.
> List CLUES (i.e., keywords, phrases, contextual information, semantic meaning, semantic relationships, tones, references) that support the spoiler detection of the input.
> Next, deduce the diagnostic REA-

Hulk	There he proceeds to wreak havoc in the city until Betty arrives and calms him down. David taps into a powerline and becomes living electricity. Bruce transforms into the Hulk and the two men battle.
The Shawshank Redemption	Red believes Andy intends to use the hammer to engineer his escape in the future but when the tool arrives and he sees how small it is, Red puts aside the thought that Andy could ever use it to dig his way out of prison. He goes to a halfway house but finds it impossible to adjust to life outside the prison. He eventually commits suicide.
Mean Girls	In her efforts to get revenge on Regina, Cady gradually loses her individual personality and remakes herself in the image of Regina. She soon becomes as spiteful as Regina, abandoning Janis and Damien and focusing more on her image. Regina storms out, pursued by an apologetic Cady, and gets hit by a school bus in her haste. At the Spring Fling dance, Cady is elected Spring Fling Queen, but in her acceptance speech, she declares her victory is meaningless: they are all wonderful in their own way and thus the victory belongs to everyone.
Blood Diamond	Dia is conscripted into the rebel forces, the brainwashing eventually turning him into a hardened killer. Archer holds off the soldiers chasing them while Solomon and Dia flee, and then makes a final phone call to Bowen, asking her to help Solomon as a last favor before looking out over the beautiful landscape of Africa once more and dying peacefully.

Table 7: Important events of each movie.

CoT (Kojima et al., 2022)	You are detecting “Spoilers” in movie reviews. “Spoilers” is a description of a significant plot point or other aspect of a movie, which if previously known may spoil a person’s first experience of the work. A significant plot point is one that cannot be predicted from the filmś introduction or early developments. Based on introduction, does the following review contain spoilers? introduction: [intro] review: [review] reasoning: Let’s think step by step. [reasoning]
CARP (Sun et al., 2023)	This is a Spoiler Detection for input movie reviews. List CLUES (i.e., keywords, phrases, contextual information, semantic meaning, semantic relationships, tones, references) that support the spoiler detection of the input. Next, deduce the diagnostic REASONING process from premises (i.e., introduction, clues, input) that support the spoiler detection. Finally, based on the introduction, the clues, the reasoning and the input, categorize the overall ANSWER of input as SPOILER or NOT SPOILER. introduction: [intro] review: [review] clues: [clue] reasoning: [reasoning]

Table 8: Prompts of the baseline methods.

Hulk	Bruce Banner, a genetics researcher with a tragic past, suffers a lab accident that makes him transform into a raging, giant green monster when angered, making him a target of forces seeking to abuse his power.
The Shawshank Redemption	A banker convicted of uxoricide forms a friendship over a quarter century with a hardened convict, while maintaining his innocence and trying to remain hopeful through simple compassion.
Mean Girls	Cady Heron is a hit with The Plastics, the A-list girl clique at her new school, until she makes the mistake of falling for Aaron Samuels, the ex-boyfriend of alpha Plastic Regina George.
Blood Diamond	A fisherman, a smuggler, and a syndicate of businessmen match wits over the possession of a priceless diamond.

Table 9: Introduction texts of the movies.

Million Dollar Baby	input: I can’t find any reason for not loving this movie as much as possible. clues: “I can’t find”, “loving”. spoiler type: life or death, character features, development of the story, appearance. reasoning: This review is just an opinion like “can’t find” and “loving” and does not touch on the content of the movie. Therefore, it does not match the spoiler type. answer: NOT SPOILER.
	input: When Maggie finally gets her title fight, an illegal punch by her monster-like opponent sends her to the mat, landing head-first on her corner stool- an event which in real life would disqualify her opponent and possibly concuss Maggie instead wins her opponent the fight and renders Maggie paralyzed, bedridden and ventilator-dependent for the rest of her miserable life. clues: “an illegal punch”, “Maggie paralyzed, bedridden”, “the rest of her miserable life”. spoiler type: problem occurs, life or death, character features, past, character relationships, development of the story, appearance. reasoning: This review is about Maggie suffering a concussion and becoming bedridden, an event that changes her life and is a key plot of the movie. Therefore, this review matches “problem occurs” and “development of the story”. answer: SPOILER.
The Fast and the Furious	input: It is plot is plain and predictable but because it’s unique itself and is origin of all illegal street racing movies so this makes the meaning of the plot inconsequential. clues: “plot is plain and predictable”. spoiler type: past, trick, appearance. reasoning: This review criticizes the storyline but does not reveal any specifics. Therefore, it does not match the spoiler type. answer: NOT SPOILER.
	input: It’s beautiful to see Brian and Dom at the end: Brian betrayed him and should arrest him but instead, they do the 10 second-race and don’t know what to think about each other. clues: “It’s beautiful to see Brian and Dom”, “Brian betrayed him”, “should arrest him”, “the 10 second-race”. spoiler type: life or death, true identity, character features, trick, past, character relationships, appearance, development of the story. reasoning: This review is about the last scene of the movie and my thoughts on that scene. Although it contains thoughts, this review contains the important plot of the last scene of the movie. Therefore, this review matches “character relationships” and “development of the story”. answer: SPOILER.

Table 10: Reviews, the clues, spoiler types, the reasonings, and the answers used in few-shot learning, part 1. The information to be used is determined according to the conditions of each method (e.g., “reasoning” is omitted for methods that do not perform reasoning).

Groundhog Day	input: This pattern happens over and over again until he realizes he cannot escape Groundhog Day. clues: “happens over and over again”, “cannot escape Groundhog Day”. spoiler type: life or death, character relationships, character features, appearance. reasoning: This review talks about the film repeating the same day over and over again, which is the premise of the film and is also used in the film’s introduction. Therefore, it does not match the spoiler type. answer: NOT SPOILER.
	input: As, most notably, is the way that Andie MacDowell’s Rita can so magically change her opinion of Phil the second that she finds out that he plays an instrument. clues: “Rita”, “magically change her opinion”. soiler type: problem occurs, character relationship, life or death, character features, trick, appearance. reasoning: The review notes that Rita eventually develops feelings for Connors, which is a key plot point in the film’s final conclusion. Therefore, this review matches “tricks” and “character relationship”. answer: SPOILER.
Match Point	input: Sensing an opportunity to climb the social ladder he starts seeing her just as he meets Nola Rice (Scarlett Johanssen), an aspiring American actress, whom he openly flirts with until he realizes she’s Tom’s girlfriend, but an outsider in the Wilton household. clues: “starts seeing her”, “he openly flirts”. spoiler type: past, character features, trick. reasoning: The review notes that Chris begins an affair, but this is just an introduction to the film and not a major plot point. Therefore, it does not match the spoiler type. answer: NOT SPOILER.
	input: Then she starts to get clingy and so he kills her. clues: “starts to get clingy”, “he kills her”. spoiler type: problem occurs, life or death, true identity, character features, appearance. reasoning: The review is about a woman who becomes annoyed with a man and ends up killing her. Therefore, this review matches “life or death” and “problem occurs”. answer: SPOILER.

Table 11: Reviews, the clues, spoiler types, the reasonings, and the answers used in few-shot learning, part 2.

Million Dollar Baby	Frankie, an ill-tempered old coach, reluctantly agrees to train aspiring boxer Maggie. Impressed with her determination and talent, he helps her become the best and the two soon form a close bond.
The Fast and the Furious	Los Angeles police officer Brian O'Conner must decide where his loyalty really lies when he becomes enamored with the street racing world he has been sent undercover to end it.
Groundhog Day	A narcissistic, self-centered weatherman finds himself in a time loop on Groundhog Day.
Match Point	At a turning point in his life, a former tennis pro falls for an actress who happens to be dating his friend and soon-to-be brother-in-law.

Table 12: Introduction texts of the movies in the few-shot learning.

SONING process from premises (i.e., introduction, clues, input) that support the spoiler detection.

Finally, based on the introduction, the clues, the reasoning and the input, categorize the overall ANSWER of input as SPOILER or NOT SPOILER.

introduction: [intro]

review: [review]

clues: [clue]

reasoning: [reasoning]

answer:

prompt of – reasoning

This is a Spoiler Detection for input movie reviews.

List CLUES (i.e., keywords, phrases, contextual information, semantic meaning, semantic relationships, tones, references) that support the spoiler detection of the input.

Finally, based on the introduction, the clues and the input, categorize the overall ANSWER of input as SPOILER or NOT SPOILER.

introduction: [intro]

review: [review]

clues: [clue]

answer:

CAST

prompt of + reasoning

This is a Spoiler Detection for input movie reviews.

"Spoilers" is a description of a significant plot point or other aspect of a movie, which if previously known may spoil person's first experience of the work.

A significant plot point is one that cannot be predicted from the film's introduction or early developments.

List CLUES (i.e., keywords, phrases, contextual information, semantic meaning, semantic relationships, tones, references) that support the spoiler detection of the input.

Next, deduce the diagnostic REASONING process from premises (i.e., introduction, clues, input) that support the spoiler detection.

Finally, based on introduction, clues, spoiler types, the reasoning and the input, categorize the overall ANSWER of input as SPOILER or NOT SPOILER.

introduction: [intro]

review: [review]

clues: [clue]

spoiler type: [types]

reasoning: [reasoning]

answer:

prompt of - reasoning

This is a Spoiler Detection for input movie reviews.

"Spoilers" is a description of a significant plot point or other aspect of a movie, which if previously known may spoil a person's first experience of the work.

A significant plot point is one that cannot be predicted from the film's introduction or early developments.

List CLUES (i.e., keywords, phrases,

contextual information, semantic meaning, semantic relationships, tones, references) that support the spoiler detection of the input.

Finally, based on introduction, clues, spoiler types, and the input, categorize the overall ANSWER of input as SPOILER or NOT SPOILER.

introduction: [intro]

review: [review]

clues: [clue]

spoiler type: [types]

answer: