Fine-Grained Spoiler Detection in Book Reviews: An Ensemble Approach Combining Classification and Listwise Ranking

Anay Kulkarni
ankulkarni@ucsd.edu

Abstract

This paper introduces a hybrid computational approach for automatically detecting character references, plot twists, and such elements that might be revealing to the storyline from book reviews. We begin by compiling a representative dataset from the large-scale Goodreads book review collection, incorporating fine-grained sentence-level spoiler annotations alongside book-level metadata. We extract item-level and review-level features informed by prior research through feature engineering. We then approach spoiler detection from two perspectives: an absolute context, where sentences are explicitly classified as spoilers or non-spoilers, and a relative context, where sentences within a review are ranked based on their likelihood of containing spoilers. To further enhance detection performance, we propose an ensemble method that integrates these two strategies, yielding significant improvements over baseline models.

1 Introduction

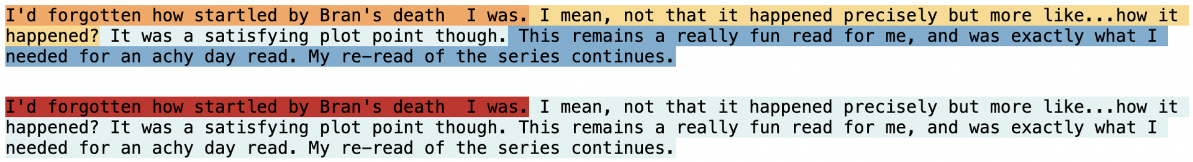
Spoilers in reviews can significantly diminish the enjoyment of media by revealing critical plot details (Loewenstein, 1994). While some review platforms rely on users to self-report spoilers or employ crowdsourcing to flag spoiler content, these approaches face notable challenges, such as low participation rates and scalability limitations (Wan et al., 2019). Prior research has explored machine-learning techniques for spoiler detection to address these shortcomings. Early studies primarily utilized simple topic models, linguistic features, or external metadata, but these approaches often failed to capture sentence dependencies, user- or item-specific spoiler biases, and contextual semantics.

More recently, state-of-the-art systems have leveraged deep neural networks, such as hierarchical attention network (Yang et al., 2016), to improve spoiler detection. However, these systems face challenges due to inconsistent and subjective spoiler tagging standards. Self-reported labels in datasets often include controversial annotations, where even human annotators struggle to reach a consensus (Wan et al., 2019). For instance, a sentence that may be classified as a 'non-spoiler' by the language model is tagged as a spoiler by the review author, likely due to its contextual connection to the preceding spoiler sentence. These inconsistencies highlight the limitations of binary classification approaches and motivate a shift toward treating spoiler detection as a ranking task. By ranking sentences within the same review context based on their likelihood of being spoilers, this approach reduces reliance on absolute thresholds and offers a more robust evaluation framework.

Building on these insights, our study proposes a novel approach that frames spoiler detection as both a classification and ranking task. Leveraging fine-grained sentence-level annotations from the Goodreads dataset, we combine a classifier model with a list-wise ranking model. This ensemble approach provides two benefits - the classifier model captures the absolute sentence-level information that deems a sentence as a spoiler, while the list-wise ranking model captures review-level dependencies between sentences, i.e., contextual connections to a preceding or succeeding spoiler sentence. Our experiments demonstrate the effectiveness of this hybrid approach in addressing the complexities of fine-grained spoiler detection.

Experimental Steps Undertaken:

- Data Collection and Preprocessing
- Data Preparation and Exploratory Analysis
- Feature Engineering
- Model Training and Evaluation



```
I'd forgotten how startled by Bran's death  I was. I mean, not that it happened precisely but more like...how it
happened? It was a satisfying plot point though. This remains a really fun read for me, and was exactly what I
needed for an achy day read. My re-read of the series continues.

I'd forgotten how startled by Bran's death  I was. I mean, not that it happened precisely but more like...how it
happened? It was a satisfying plot point though. This remains a really fun read for me, and was exactly what I
needed for an achy day read. My re-read of the series continues.
```

Figure 1: *Visualizing fine-grained spoiler detection as a ranking task (First) v/s binary classification task (Second). Blue or colder colors represent a lower rank, implying a lower likelihood of containing spoilers. Red or warmer colors represent a higher rank and, therefore, a higher likelihood of spoilers being contained. Here, we can see that LambdaRank correctly ranks the spoiler sentence the highest and captures dependencies between the neighboring sentences, which contain contextual references to the spoiler. The binary classification shown are true labels, annotated by humans.*

2 Related work

Spoiler detection in text is a relatively underexplored domain. Early works primarily relied on traditional machine learning techniques, incorporating features such as unigrams, frequent verbs, named entities, and metadata like review subjects or genres (Boyd-Graber et al., 2013; Guo and Ramakrishnan, 2010; Jeon et al., 2013; Iwai et al., 2014). These approaches typically used standard classifiers, such as Support Vector Machines, to identify spoilers. More recently, deep learning methods have been applied to this task, with a notable study leveraging genre-specific external information to improve spoiler detection (Chang et al., 2018). However, prior research largely overlooks critical aspects such as inter-sentence dependencies, user and item spoiler biases, and item-specific semantics in spoiler content.

In related fields, neural network approaches—such as CNN-based (Kim, 2014), RNN-based (Yang et al., 2016), and self-attention-based models like BERT (Devlin et al., 2018)—have demonstrated substantial success in sentence and document classification. Inspired by these advancements, spoiler detection can be framed as a domain-specific sentence classification task, necessitating techniques that model contextual and semantic patterns unique to spoilers. This study seeks to address these gaps by leveraging domain-specific insights and large-scale, annotated datasets to advance spoiler detection methods.

A notable contribution to spoiler detection is SpoilerNet (Wan et al., 2019), which frames spoiler sentence identification as a binary classification problem. SpoilerNet builds upon the Hierarchical Attention Network (HAN) (Yang et al., 2016) and incorporates key domain-specific features. It models sentence-level dependencies using bidirectional GRUs and enhances word embeddings with item-specific features, such as document frequency (DF) and inverse item frequency (IIF). SpoilerNet also accounts for user and item biases through learnable parameters in the output layer, which adjusts for disparities in spoiler distributions. The integration of attention mechanisms helps focus on spoiler-relevant terms (e.g., ”kill,” ”die”), and the model achieves improved performance by explicitly considering contextual semantics and dependencies within reviews. This approach highlights the potential of combining deep learning with item-specific and contextual information for nuanced spoiler detection tasks.

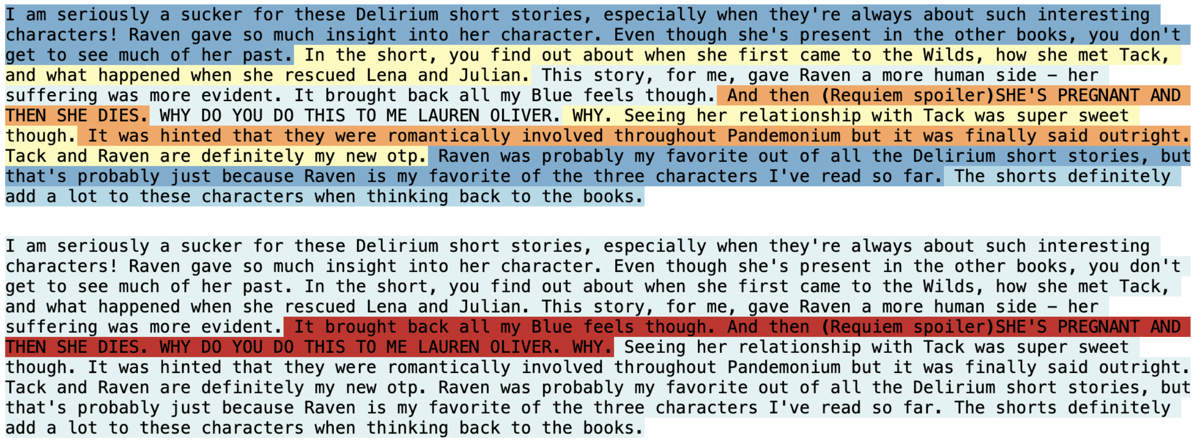
3 Goodreads Book Review Dataset

The Goodreads spoiler reviews dataset (Wan et al., 2019) is a large-scale collection of 1,378,033 English book reviews scraped from Goodreads, covering 25,475 books and 18,892 users, with each book and user having at least one associated spoiler review. The dataset includes 17,672,655 sentences, of which 3.22% are labeled as spoiler sentences, making it the first dataset of this scale to provide fine-grained spoiler annotations.

A book metadata dataset is included, providing complementary information such as book titles, genres, and other attributes, enabling more nuanced analysis and modeling. This dataset is publicly available at https://mengtingwan.github.io/data/goodreads.html

3.1 Data Analysis and Preprocessing

We utilize the Goodreads book review spoiler dataset, which comprises approximately 1.3 million reviews (17 million review sentences). Due



I am seriously a sucker for these Delirium short stories, especially when they're always about such interesting characters! Raven gave so much insight into her character. Even though she's present in the other books, you don't get to see much of her past. In the short, you find out about when she first came to the Wilds, how she met Tack, and what happened when she rescued Lena and Julian. This story, for me, gave Raven a more human side - her suffering was more evident. It brought back all my Blue feels though. And then (Requiem spoiler)SHE'S PREGNANT AND THEN SHE DIES. WHY DO YOU DO THIS TO ME LAUREN OLIVER. WHY. Seeing her relationship with Tack was super sweet though. It was hinted that they were romantically involved throughout Pandemonium but it was finally said outright. Tack and Raven are definitely my new otp. Raven was probably my favorite out of all the Delirium short stories, but that's probably just because Raven is my favorite of the three characters I've read so far. The shorts definitely add a lot to these characters when thinking back to the books.

I am seriously a sucker for these Delirium short stories, especially when they're always about such interesting characters! Raven gave so much insight into her character. Even though she's present in the other books, you don't get to see much of her past. In the short, you find out about when she first came to the Wilds, how she met Tack, and what happened when she rescued Lena and Julian. This story, for me, gave Raven a more human side - her suffering was more evident. It brought back all my Blue feels though. And then (Requiem spoiler)SHE'S PREGNANT AND THEN SHE DIES. WHY DO YOU DO THIS TO ME LAUREN OLIVER. WHY. Seeing her relationship with Tack was super sweet though. It was hinted that they were romantically involved throughout Pandemonium but it was finally said outright. Tack and Raven are definitely my new otp. Raven was probably my favorite out of all the Delirium short stories, but that's probably just because Raven is my favorite of the three characters I've read so far. The shorts definitely add a lot to these characters when thinking back to the books.

Figure 2: *The first sentence classified as a spoiler according to the true annotations seems controversial as it contains no revelatory information. Note that such an example is difficult even for human annotators to justify. LambdaRank does not make the mistake of assigning high scores to such sentences and only targets the sentences containing potential spoilers. Furthermore, we notice that some sentences that contain spoilers, which the human annotators missed, were also successfully captured by our model.*

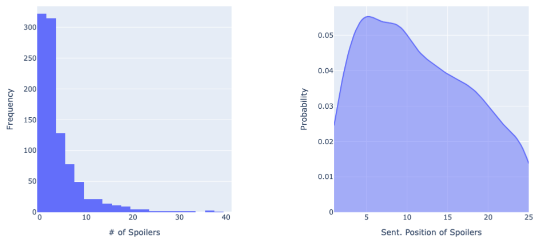


Figure 3: *(a) Frequency Distribution of the number of spoiler sentences in reviews, (b) Probability density function of positions of spoiler sentences within reviews*

to computational constraints, we limit our analysis to 1,000 carefully curated reviews, ensuring each review contains at least one spoiler sentence. From these reviews, we extract 18,969 sentences with sentence-level spoiler annotations. Additionally, book titles are compiled from the associated metadata to enrich the feature space.

Statistic	**Value**
# reviews	1,000
# unique items (books)	859
# review sentences	18,969
Avg. len of reviews (in sent.)	18.969
Avg. # of spoiler sent	4.806
Avg. position	20.934
Median position	15.0

Table 1: *Preliminary Analysis of the Dataset*

Table 1 shows the preliminary statistics of our sampled dataset. Our sampled dataset comprises 1000 reviews of 859 unique books, with a total of 18,969 sentences analyzed. On average, each review contains approximately 18.97 sentences, out of which 4.81 are spoilers. Spoiler sentences tend to appear later in the reviews, with their average position being 20.93 and the median position at 15.0. These insights suggest that spoiler sentences are distributed non-uniformly, often occurring towards the latter half of the reviews.

Figure 3 shows the frequency and probability density distributions of spoiler sentences. This observation may provide useful signals for spoiler detection models. See `EDA.ipynb`

3.2 Data Preparation

The dataset is split into 70% training, 20% validation, and 10% testing sets to facilitate model training and evaluation.

4 Baselines

Baseline Model: We start with a simple logistic regression binary classifier trained on sentence embeddings. This provides an initial benchmark. Using BERT's pre-trained embeddings, we generate dense 768-dimensional feature vectors. To reduce computational complexity and mitigate the curse of dimensionality, we apply Principal Component Analysis (PCA) to reduce the vector dimensions to 128.

5 The Proposed Approach: Combining Classification & Listwise Ranking

To implement this approach, we employ LightGBM's LambdaRank algorithm, designed for listwise ranking. LambdaRank optimizes a ranking-specific loss function that directly incorporates the order of sentences within a review, ensuring that sentences with higher probabilities of being spoilers are ranked above those less likely to be spoilers. This approach captures dependencies between sentences, enabling a contextual understanding of their spoiler likelihood. The listwise ranking loss considers the relative importance of ranked pairs, making it well-suited for our task.

The gradients for LambdaRank are computed as, $\lambda_{ij} = -\Delta NDCG_{ij} \cdot \sigma \cdot (1 - P_{ij})$ where:

- $\Delta NDCG_{ij} = |D(r_i) - D(r_j)|$
- $D(r) = \frac{1}{\log_2(r+1)}$ is the NDCG discount factor for rank r.

The total listwise ranking loss combines the contributions from all pairs (i, j) of sentences in a given review:

$$L = \sum_{i<j} \Delta NDCG_{ij} \cdot \log\left(1 + e^{-(f(i)-f(j))}\right)$$

To further enhance the performance, we propose an ensemble method that combines the outputs of a logistic regression classifier with the LambdaRank model. First, we train a binary classifier to predict the likelihood of each sentence being a spoiler, producing a probability score for each sentence. These probabilities are then fed into the LambdaRank model as features, allowing the ensemble model to leverage both the absolute predictions from the classifier and the relative rankings generated by LambdaRank. This hybrid framework balances sentence-level predictions with review-level dependencies, improving the granularity of spoiler detection.

We evaluate our models using a combination of classification and ranking metrics. Accuracy and ROC-AUC are primary metrics to assess the classifier's ability to distinguish spoilers from non-spoilers. For the ranking model, we incorporate Normalized Discounted Cumulative Gain (NDCG) to measure the effectiveness of ranked outputs. Note that NDCG is used for fine-tuning LambdaRank but not for comparing models. The combination of these metrics provides a comprehensive evaluation of both sentence-level classification performance and the quality of ranked outputs, demonstrating the robustness of our proposed approach.

	Goodreads	
	AUC	Accuracy
Logistic (Baseline)	0.704	82.67
+ item-spec, position	↑ 0.712	↑ 83.03
LambdaRank		
+ position.	↑ 0.743	↑ 82.84
+ item-spec	↑ 0.769	↑ 83.15
Ensemble		
+ neighbours, proba, TFIDF	↑ 0.791	↑ 84.18

Table 2: *Spoiler sentence detection results on Goodreads, where arrows indicate the performance boost (↑) compared with the base model in each group. The best results are highlighted.*

6 Experiments

We describe the models used for this experiment.

1. **Baseline Model**: This model employs a simple logistic classifier trained on 128-dimensional sentence embeddings. See `Baseline-Classifier.ipynb`

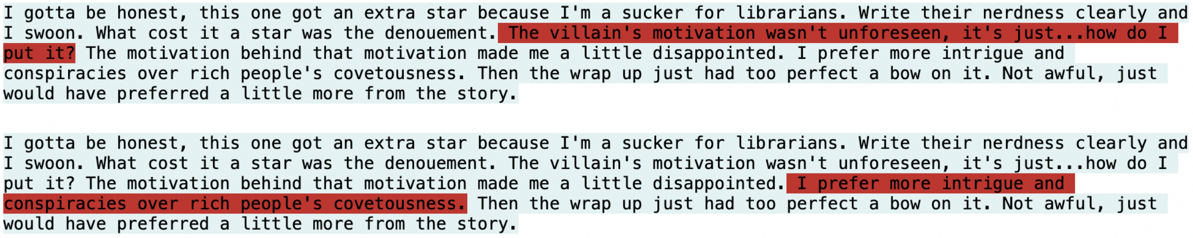
2. **Improved Baseline**: We enhance the baseline by incorporating additional item-specific features in this model. As book-specific terms such as locations or characters' names could be informative to reveal plot information (Jeon et al., 2013), we develop an effective method to identify the specificity of tokens regarding each item (i.e., each book). This is inspired from (Wan et al., 2019) and is computed as follows:

   - **(Popularity)** For word w, item i, we calculate the item-wise document frequency (DF) as

   $$DF_{w,i} = \frac{|D_{w,i}|}{|D_i|};$$

   - **(Uniqueness)** For each word w, we calculate its inverse item frequency (IIF) as

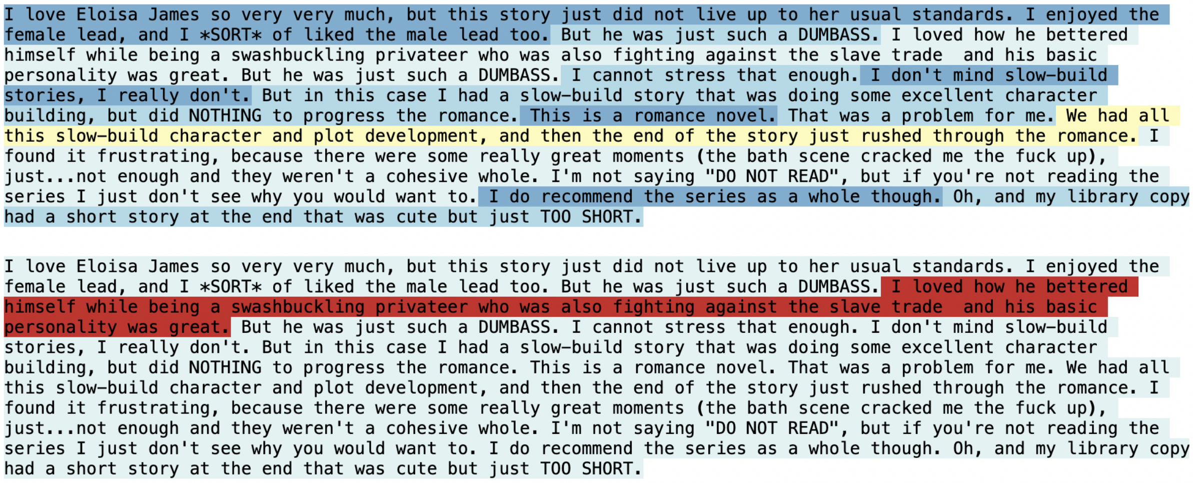
   $$IIF_w = \log \frac{|I| + \epsilon}{|I_w| + \epsilon};$$



I gotta be honest, this one got an extra star because I'm a sucker for librarians. Write their nerdness clearly and I swoon. What cost it a star was the denouement. The villain's motivation wasn't unforeseen, it's just...how do I put it? The motivation behind that motivation made me a little disappointed. I prefer more intrigue and conspiracies over rich people's covetousness. Then the wrap up just had too perfect a bow on it. Not awful, just would have preferred a little more from the story.

I gotta be honest, this one got an extra star because I'm a sucker for librarians. Write their nerdness clearly and I swoon. What cost it a star was the denouement. The villain's motivation wasn't unforeseen, it's just...how do I put it? The motivation behind that motivation made me a little disappointed. I prefer more intrigue and conspiracies over rich people's covetousness. Then the wrap up just had too perfect a bow on it. Not awful, just would have preferred a little more from the story.

Figure 4: *Baseline model struggles to identify spoilers embedded in subjective statements*



I love Eloisa James so very very much, but this story just did not live up to her usual standards. I enjoyed the female lead, and I *SORT* of liked the male lead too. But he was just such a DUMBASS. I loved how he bettered himself while being a swashbuckling privateer who was also fighting against the slave trade and his basic personality was great. But he was just such a DUMBASS. I cannot stress that enough. I don't mind slow-build stories, I really don't. But in this case I had a slow-build story that was doing some excellent character building, but did NOTHING to progress the romance. This is a romance novel. That was a problem for me. We had all this slow-build character and plot development, and then the end of the story just rushed through the romance. I found it frustrating, because there were some really great moments (the bath scene cracked me the fuck up), just...not enough and they weren't a cohesive whole. I'm not saying "DO NOT READ", but if you're not reading the series I just don't see why you would want to. I do recommend the series as a whole though. Oh, and my library copy had a short story at the end that was cute but just TOO SHORT.

I love Eloisa James so very very much, but this story just did not live up to her usual standards. I enjoyed the female lead, and I *SORT* of liked the male lead too. But he was just such a DUMBASS. I loved how he bettered himself while being a swashbuckling privateer who was also fighting against the slave trade and his basic personality was great. But he was just such a DUMBASS. I cannot stress that enough. I don't mind slow-build stories, I really don't. But in this case I had a slow-build story that was doing some excellent character building, but did NOTHING to progress the romance. This is a romance novel. That was a problem for me. We had all this slow-build character and plot development, and then the end of the story just rushed through the romance. I found it frustrating, because there were some really great moments (the bath scene cracked me the fuck up), just...not enough and they weren't a cohesive whole. I'm not saying "DO NOT READ", but if you're not reading the series I just don't see why you would want to. I do recommend the series as a whole though. Oh, and my library copy had a short story at the end that was cute but just TOO SHORT.

Figure 5: *Ensemble model is distracted by non-spoiler neighbours*

- Then for each term w, item i, we are able to obtain the DF-IIF as

$$DF_{w,i} \times IIF_w.$$

These scores are then averaged across all terms in the sentence. Additionally, we include the sentence's position within the review based on prior research suggesting that spoilers are more likely to appear later in the text. See `itemspecificity.py`

3. **Listwise Ranking Model**: Using LightGBM's LambdaRank, we develop a listwise ranking model incorporating sentence embeddings and positional information, capturing review-level dependencies. See `ListwiseRanking-Base.ipynb`

4. **Improved Listwise Ranking Model**: To improve performance, we introduce additional features such as item-specificity scores and TF-IDF values, further refining the model's ability to rank sentences effectively. See `ListwiseRanking-Improved.ipynb` **hyperparameters**:'objective':'lambdarank', 'metric': 'ndcg', 'ndcg_eval_at': [1, 3, 5], 'num_leaves': 31, 'max_depth': -1, 'learning_rate': 0.036, 'boosting_type': 'gbdt', 'min_data_in_leaf': 21

5. **Ensemble Model**: This model combines the predicted probabilities from the improved baseline as inputs to the enhanced listwise ranking model. In addition to BERT embeddings, we introduce TFIDF scores for each sentence. We also introduce a new feature that captures the context of neighboring sentences, specifically whether they are predicted as spoilers or non-spoilers. This is based on the observation that spoilers often appear together in reviews; thus, if neighboring sentences are spoilers, the likelihood of the current sentence being a spoiler increases. We use the same hyperparameters as before. See `Ensemble.ipynb`

These models can be trained on local machines and do not require GPU support. Models can be trained and evaluated in less than a minute. However, generating BERT embeddings might take longer.

Results. **Table 2** presents the results of spoiler sentence detection on the Goodreads dataset, highlighting the performance of different models in terms of AUC and accuracy. The logistic baseline achieves an AUC of 0.704 and an accuracy of 82.67%. Incorporating item-specific and positional features slightly improves its performance, with the AUC rising to 0.712 and accuracy to 83.03%. The LambdaRank model outperforms the baseline, with the inclusion of positional features improving the AUC to 0.743 and accuracy to 82.84%. Adding item-specific features further enhances its performance, achieving an AUC of 0.769 and an accuracy of 83.15%. The Ensemble model, which leverages neighboring sentence information and classification probabilities, achieves the best results with an AUC of 0.791 and an accuracy of 84.18%. These results indicate that the Ensemble model benefits significantly from contextual and probabilistic cues, making it the most effective approach. Arrows in the table denote performance improvements over the respective baselines.

7 Error analysis

We discuss two main issues with the proposed method.

Subjectivity. We first analyze the performance of the baseline classification model. **Figure 4** highlights a recurring challenge: the model struggles to identify spoilers embedded within subjective or indirect statements. For example, when a review includes a statement like, ”The villain’s motivation was...”—directly referencing a key storyline element—the model can discern its spoiler-like nature compared to a more indirect revelation through an opinion such as, ”I prefer seeing more of...”. This inability to capture contextual subtleties underscores the limitations of the current sentence embeddings used by the model.

Improving the embeddings could be a critical step forward. Fine-tuning pre-trained embeddings, such as those from BERT, on this specific dataset might help the model better capture domain-specific nuances and relationships. Additionally, employing advanced encoder architectures, such as RoBERTa or Sentence-T5, could generate richer, context-aware vector representations, potentially addressing these gaps.

The LambdaRank model, while not immune to similar challenges—mainly because it relies on the same embeddings—shows a modest performance improvement. This can be attributed to its ability to assess spoilers at the review level rather than restricting itself to isolated sentences. By considering sentence relationships within the broader review context, the LambdaRank model leverages structural information that helps mitigate sentence-level prediction limitations. This highlights the potential of ranking-based approaches to handle the intricacies of spoiler detection more effectively.

Distraction. The Ensemble model leverages the critical observation that spoiler sentences often appear in clusters within reviews. By combining classification probabilities with contextual information from neighboring sentences, the model demonstrates an overall improvement in performance. This approach allows the model to better capture the dependencies between sentences, enhancing its ability to detect spoilers embedded in a larger narrative. However, this strength can also become a limitation in certain cases, as shown in **Figure 5**. Specifically, we observe instances where a spoiler sentence is incorrectly classified as a non-spoiler due to the overwhelming influence of surrounding sentences that are strongly identified as non-spoilers. This phenomenon suggests that the contextual influence of neighboring sentences may sometimes override the intrinsic properties of the target sentence.

To address this limitation, we propose refining the way neighboring sentence probabilities are incorporated. A potential solution could involve using a weighted sum of the probabilities of surrounding sentences, with weights decreasing as the distance from the target sentence increases. This strategy would ensure that sentences farther away have a proportionally smaller impact on the target sentence’s classification, thereby reducing the undue influence of distant, strongly non-spoiler sentences.

Interestingly, our analysis reveals that this influence is particularly skewed toward the positive class (spoilers). In other words, strong non-spoiler sentences around a potential spoiler sentence have a more pronounced impact on misclassifying it as a non-spoiler. Conversely, spoilers around potential non-spoiler sentences exert a weaker influence on their classification. This asymmetry highlights a bias in how the ensemble model processes contextual information. It suggests that additional adjustments, such as incorporating class-specific weights, may further enhance its robustness in handling such edge cases.

8 Conclusion and Future Work

Overall, the ensemble approach to combining binary classification with listwise ranking proves to be promising. There are a few apparent directions for future work, one of which is scaling the model to incorporate the larger dataset. In these experiments, we limited our scope to only a subset of 1000 reviews compared to the 1.3 million reviews available in the larger dataset. Another direction would be to use classification probabilities from SpoilerNet as inputs to the LambdaRank model rather than using the Logistic classifier baseline. This could potentially improve SpoilerNet's SOTA performance. Other classifiers like XGBoost could also prove to be good contenders as they belong to the same gradient-boosting family of algorithms as LambdaRank.

9 Acknowledgements

- Some of the LaTex formatting for this paper was prepared using ChatGPT with minor changes
- Content was proof-read and revised using Grammarly with minor changes

References

Mengting Wan, Rishabh Misra, Ndapa Nakashole, and Julian J. McAuley. 2019. Fine-Grained Spoiler Detection from Large-Scale Review Corpora *CoRR, vol. abs/1905.13416, 2019.*

Jordan L. Boyd-Graber, Kimberly Glasgow, and Jackie Sauter Zajac. 2013. Spoiler alert: Machine learning approaches to detect social media posts with revelatory information. In *ASIS&T Annual Meeting*.

Buru Chang, Hyunjae Kim, Raehyun Kim, Deahan Kim, and Jaewoo Kang. 2018. A deep neural spoiler detection model using a genre-aware attention mechanism. In *PAKDD*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*.

Sheng Guo and Naren Ramakrishnan. 2010. Finding the storyteller: Automatic spoiler tagging using linguistic cues. In *COLING*.

Hidenari Iwai, Yoshinori Hijikata, Kaori Ikeda, and Shogo Nishida. 2014. Sentence-based plot classification for online review comments. In *WI-IAT*.

Sungho Jeon, Sungchul Kim, and Hwanjo Yu. 2013. Don't be spoiled by your friends: Spoiler detection in TV program tweets. In *ICWSM*.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *EMNLP*.

George Loewenstein. 1994. The psychology of curiosity: A review and reinterpretation. *Psychological bulletin*, 116(1):75.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alexander J. Smola, and Eduard H. Hovy. 2016. Hierarchical attention networks for document classification. In *NAACL*.