Spoiler detection and extraction
POC for NLP Course, Winter 2022

M. Kierznowski, Ł. Pancer, P. Wesołowski
Warsaw University of Technology

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

Abstract

This document presents exploratory data analysis, describes current experiments, and provides preliminary results for the first NLP course project. The project addresses the spoiler detection task. The main goal is to evaluate the performance of the spoiler classification models through an interpretability-based technique, i.e., compare their critical phrases with the ones annotated. So far, we have focused on creating machine-learning models. The next step will be to use XAI tools on them.

1 Introduction

As an introduction, let us remind that we aim to analyze models dealing with the classification of entire reviews as containing a spoiler or not. Once the relevant models have been created, we will attempt to compare the sets of words that influence the decision to classify as a spoiler and annotated segments. Moreover, we hope to identify other interesting properties of the models.

We have focused on three datasets, namely:

- Goodreads - 1.3M documents, 17M sentences, 570k spoiler sentences (Wan et al., 2019),
- TV Tropes Books - 340k documents, 670k sentences, 110k spoiler sentences (Wróblewska et al., 2021),
- IMDB reviews - 5.5M documents, 1.1M spoiler (Biswas, 2021)

Their properties are further examined in section 2.

2 Exploratory data analysis

2.1 Goodreads

This data set may be found in two forms: balanced, which is used in provided research, and standard. We have selected the balanced alternative in order to provide research on at least one balanced dataset. Moreover, it contains sentence-level annotation, indicating which sentences carry unwanted information. For these sentences (assigned to accommodate spoilers), the frequency distribution of words is provided in figure 1. Note that before calculations, stop words were deleted (downloaded from the nltk package) as well as punctuation marks such as "," or ".".

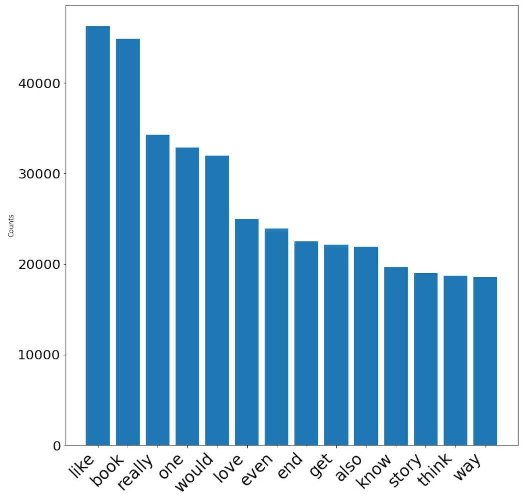

Figure 1: Frequency Distribution of sentences containing spoiler for Goodreads dataset

Analyzing provided figure, we can presume that words such as *love*, *know*, or *story* may hold undesirable content.

Finally, one important issue is worth elaborating on. We focus on the whole review classification. Moreover, our models are BERT-based, which means they can handle up to 512 input tokens. The reviews where the first spoiler occurs after the first 512 tokens may be problematic. We investigated this topic further and estimated that

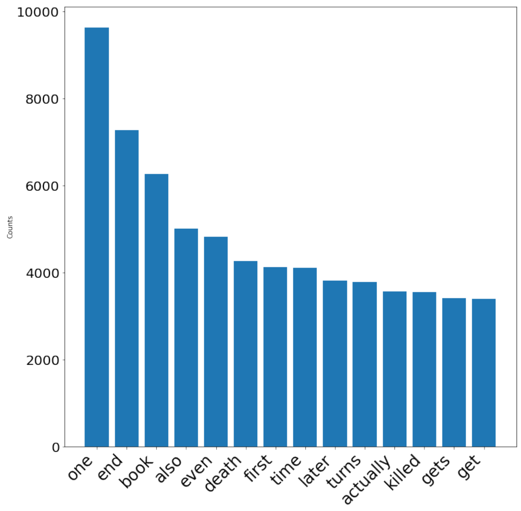

Figure 2: Frequency Distribution of sentences containing spoiler for TV Tropes Books dataset

about 90% of spoiler reviews contain at least one spoiler within the first 512 tokens. Therefore, we decided to leave it as it is. Besides, who knows if the model cannot learn if the review contains spoilers on the basis of only part of it? (but, e.g., the author's writing style, etc.).

2.2 TV Tropes Books

This dataset provides more detailed information about spoilers. It also contains word-based information. However, it also lays out which specific words are spoilers.

Applying frequency distribution of words among spoilers sentences provides compelling information about specific words, as shown in figure 2.

Figure 2 reveals words that may strongly contribute to spoilers, such as *death* or *killed*. Looking closer at those trigger words may contribute to finding some specific pattern.

2.3 IMDB reviews

Contrary to the other two datasets, the IMDB reviews set contains only document-level annotations. Therefore, we cannot extract as much information from this dataset as in previous cases.

Two sample records from this dataset are shown in table 1.

The dataset is highly imbalanced, as shown in figure 3. It features roughly 1.2M non-spoiler reviews and 4.4M spoiler-tagged reviews.

Regarding review lengths, there are 32 outliers containing more than 200 sentences. There are 625 reviews which consist of more than 100 sentences. Figure 4 shows a histogram of review length in terms of the number of sentences. Note that reviews featuring more than 100 sentences are not included. One can see that the vast majority of sentences are shorter ones, with less than 20 sentences.

Due to the very high volume of this dataset, we decided to remove randomly excessive reviews without spoilers to obtain a balanced dataset. We are aware that more sophisticated techniques may be used to deal with this problem, including class weighting or data augmentation. Since the dataset size results in a very long learning time, reducing both was desirable for us.

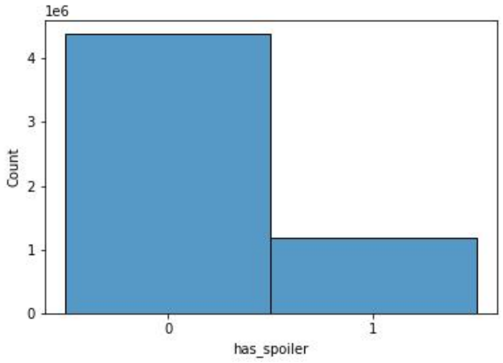

Figure 3: Class distribution in the IMDB dataset

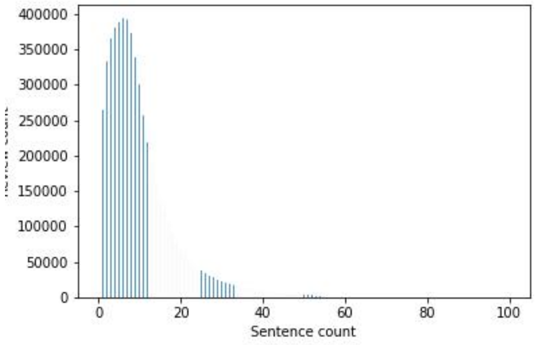

Figure 4: Histogram of review lengths w/o reviews with more than 100 sentences are not included

3 Preliminary machine learning models

In line with the approach proposed by Wróblewska (2021), we have focused on state-of-the-art solutions based on transformer

review_id	reviewer	movie	rating	review_summary	review_date	spoiler_tag	review_detail	helpful
rw0099142	ed.wenn	Le Samouraï (1967)	9.0	Cooler Than Cool	26 November 2000	0	Surely one of the suavest movies ever made. Cr...	[8, 21]
rw0099143	the red duchess	Le Samouraï (1967)	10.0	Along with 'The Wizard of Oz', the supreme fil...	7 December 2000	1	To see how beautiful, moving, exciting and ast...	[219, 284]

Table 1: Sample of IMDB reviews dataset

architecture. Namely, our current classifiers are built on BERT-based architectures. Apart from exploring the standard BERT Base-uncased model, we decided to try smaller DistilBERT architecture. Due to our computational limits, we felt it would be better to go this way rather than larger models such as the ELECTRA. According to the original paper, DistilBERT has 40% fewer parameters than BERT Base-uncased and offers 60% speedup. Table 2 shows a more exact comparison.

Model	#Parameters
DistilBERT	66M
BERT	110M

Table 2: Comparison of used BERT-based base architectures

On top of the base architecture, we use a single fully-connected linear layer preceded by a dropout. Pooled BERT outputs serve as input for the dropout layer and final classification head.

Our initial experiments were conducted without the penultimate dropout layer, but we observed overfitting, so that's why we included it in further models. Additionally, we used pre-trained weights for BERT-based models. We were utilizing the Transformers library (Wolf et al., 2020).

4 Current experiments

4.1 Goodreads

For the Goodreads dataset, we tested both DistilBERT as well as BERT models. The dataset was already split into train, test, and validation datasets. Each part was tokenized and converted to the TensorFlow Dataset class. It turned out that the training process took a long. Therefore, we have managed to train DistilBERT and BERT classifiers only for 3 epochs. The final dropout was set to 0.1. The initial learning rate was set to 2e-5 and followed by linear decay with Adam optimizer. Both models were trained using the balanced dataset and evaluated on both the balanced and entire datasets. The two datasets are disjoint, so we treat the former one as an enlarger train set.

Table 3 contains results for DistillBERT and BERT. The lighter model with a significantly smaller number of parameters achieved only slightly worse results. The most notable difference occurs in the results for the entire imbalanced dataset, as shown in table 4. We will consider focusing solely on DistillBERT in the future to be able to conduct a larger number of experiments.

Classifier architecture	Accuracy	Balanced accuracy	ROC AUC
BERT	0.8194	0.8199	0.9047
DistilBERT	0.8157	0.8162	0.8994

Table 3: Results achieved for the Goodread reviews test set

Classifier architecture	Accuracy	ROC AUC
BERT	0.8211	0.9022
DistilBERT	0.7778	0.8952

Table 4: Results achieved for the entire Goodreads reviews set

Note that in the presented experiments, all model weights were optimized. We also tried out freezing all pre-trained layers (except for the classification head), but it led to visibly worse results.

4.2 TV Tropes Books

Due to plans of combining the datasets to create a more general model, we decided on classifying the entire documents of TV Tropes as opposed to classifying individual sentences to either contain a spoiler or not. Such an approach slightly improves the balance of the dataset from 16% of spoiler-tagged reviews to 22%, which still makes a greatly imbalanced dataset. Not having access to a balanced version of it, we decided to account for the difference using class weights, which proved difficult due to the implementation of selected models and was ultimately postponed to a subsequent

Classifier architecture	Accuracy	Balanced accuracy	ROC AUC
BERT	0.8470	0.7418	0.8670

Table 5: Results achieved for the TV Tropes Books reviews test set

phase of the project. Augmentation of TV Tropes is another solution for the problem of imbalance to consider. Nevertheless, we conducted a training on TV Tropes Books without accounting for imbalance, using the same configuration as for the previous dataset.

The results shown in table 5 exceeded our modest expectations and encouraged us to further investigate proposed improvements.

4.3 IMDB

As mentioned before, we carried out experiments using a balanced IMDB dataset. It contained roughly 2.4M reviews, with half of it marked as spoilers. We split this balanced version into train, validation, and test parts in proportions of 0.8, 0.1, and 0.1. Due to recurring training time issues, particularly pronounced in the case of IMDB, we used the same configuration of 3 epochs, final dropout set to 0.1, and Adam optimizer with decaying learning rate, starting from 2e-5. We saved the weights after the second epoch because the validation results were better compared to the end of the training.

Table 6 shows the results achieved on the test set. Here, we measured only the accuracy and balanced accuracy. Note that we didn't repeat the experiments, i.e., it was conducted only once. That's due to the learning time.

Classifier architecture	Accuracy	Balanced accuracy
BERT	0.7782	0.7783

Table 6: Results achieved for the IMDB reviews test set

The results do not seem satisfactory. However, it was not our goal to build a powerful IMDB review classifier. Our aim is to utilize this task-related pre-trained model on other datasets, hoping for an improvement.

5 Addressing weak points

In the project proposal, we stated that the IMDB dataset, which we use, is rather rarely utilized in NLP works. We received a comment that this may not be true. We would therefore like to stress that the dataset we are using is different from a very popular IMDB reviews dataset introduced by Maas (2011). Note that the IMDB dataset which we explore contains about 5.5M reviews, while the second one features only 50K reviews. We agree that the latter dataset is widely known and often used for sentiment analysis, especially in some introductory materials.

Still, we don't know whether a reviewer was aware of which specific dataset we decided to use. We do not deny that we may be wrong, and this large data set may often be used. We just want to make sure that the reader distinguishes between these two datasets.

In terms of risks regarding the datasets, we are optimistic, as the preliminary results show that while the training does, in fact, take a long time with prudently created experiments, it is manageable to use them.

Furthermore, we received a few comments about the presentation itself (both positive and negative). We will try our best to think through the presentation structure, manage our time, and provide illustrative examples to make the presentation more approachable and appealing.

Some weak points we couldn't understand, e.g., ”Models are not transformers.” During the presentation, we stated that we plan to use transformer-based models, and indeed they are transformer-based. Despite the fact whether the comment was on the basis of content or technicalities, we hope that the improvements to the presentation will resolve that.

6 Summary

In this document, we have shown our current progress. The next step is to incorporate XAI techniques into the developed models.

References

Enam Biswas. 2021. Imdb review dataset - ebd. https://www.kaggle.com/dsv/1836923.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA, June. Association for Computational Linguistics.

Mengting Wan, Rishabh Misra, Ndapa Nakashole, and Julian McAuley. 2019. Fine-grained spoiler detection from large-scale review corpora. *arXiv preprint arXiv:1905.13416*.

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

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