Identifying Discomforting Content in Lengthy Fan Fiction

Daniel Hebenstreit

Graz University of Technology
daniel.hebenstreit@student.tugraz.at

Abstract

Fan fiction is a form of creative writing created by fans, inspired by existing works of fiction without official permission. As Fan fiction is of an unregulated nature, Fan fiction readers are prone to unexpectedly encounter content that might trigger discomforting or distressing feelings. In this paper, we present our approach to classifying 32 different triggers utilizing the PAN CLEF 2023 trigger detection dataset, which consists of lengthy user-written texts with class-imbalances. We adapt a text summarization method based on feature-importance achieving results on par with the top leaderboard solution of the PAN CLEF 2023 [1] contest while demanding significantly less computational resources. Furthermore, we fine-tune a transformer model and apply max-pooling on text segments, resulting in better F1-Macro and F1-Micro scores when compared to the top leaderboard solution.

Introduction

From the abduction of children in "Rumpelstiltskin" to burning witches in "Hansel and Gretel", harsh content has persisted in literature throughout centuries. While many readers do not find themselves affected by the violence in literature, there are certain subgroups that do feel discomfort or distress. Unlike more conventional art, such as movies or books, which are typically regulated by either age-recommendations or content-warnings, user written content like Fan fiction often lacks these precautions. This absence of warnings leaves subgroups vulnerable to potentially discomforting content. In our effort to enhance the protection of these vulnerable subgroups from emotionally triggering content in Fan fiction, we aim to advance current research in this area.

We employ a dataset of lengthy fan-fiction works alongside respective trigger labels. Our main approach involves fine-tuning a transformer model, extracting the respective text encodings and combining them via max-pooling to feed them in a two-layer neural network. While this approach yields a superior performance compared to other published methods on this dataset, training a transformer on multiple segments of very large texts requires a lot of computational resources. To address this issue, we have adapted the TextGuide algorithm. In this approach we extract important features of a bag-of-words (BoW) model. These important features correspond to key words within the text, and we incorporate their neighboring tokens to form a summary of each text. These summaries are then subsequently used to train a transformer model. This approach yields results that are only slightly inferior while significantly reducing the training duration, approximately being 7 times faster than our segmented approach.

Related Work

Wolska et al. [2] proposed a violence detector for Fan fiction using documents from Archive of Our Own (AO3). Nonetheless, their focus remained limited to a binary classification problem, determining the presence or absence of triggers in a given text. Building upon this work, Wiegmann et al. [3] expanded the scope of the task by creating a Trigger warning corpus consisting of 29 different warnings. They evaluated the efficiency of xgboost and support vector machines trained on tf-idf encodings, as well as transformers for this task. Subsequently, there was an open call for the PAN 2023 trigger detection challenge. In response, Sahin, Kucukkaya, and Toraman [1] presented their top-ranked solution. Their approach involved documents segmentation followed by fine-tuning a RoBERTa model on the segmented data. Afterwards, they extracted the CLS embedding of each segment of a document and fed them into a Long Short-Term Memory (LSTM) model.

Wagh et al. [4] explored various methods and compared their performance on datasets sharing similarities with the trigger detection dataset, such as extended documents lengths.

Dataset

The dataset, published within the context of the PAN CLEF 2023 trigger warning contest, consists of a train-

ing set of 307.102 texts and a validation set of 17.104 texts. Additionally, for each text, there is a corresponding ground truth comprising 32 trigger labels, where each text contains one or multiple labels. This dataset possesses several properties, including class imbalance and extensive document lengths, making it particularly intriguing for analysis.

Length of Documents

While the median document length is 2.124 words, texts can contain up to 13.253 words. Such document lengths surpass the input capacities of state-of-the-art transformers, such as BERT (512 tokens) or BigBird (4096 tokens).

Label Imbalance

As shown in Figure 1, it is evident that this dataset suffers from a significant class imbalance. The most common label is 'pornographic-content', accounting for 238.075 out of 307.102 (roughly 77%) documents. In contrast, labels like 'animal cruelty' are found in only 150 documents, being represented in a mere 0.0004% of the dataset.

Methods

Preprocessing

Our document preprocessing involves the application of the following techniques: (i) Removing HTML tags; (ii) Lowercasing; (iii) Removing special characters, numbers, and multiple spaces; (iv) Tokenization. Additionally, we employ lemmatization and create uni-gram tf-idf encodings in our Bag-of-words methodologies.

Bag-of-words

In our Bag-of-words approach, we employ a Support Vector Machine trained on tf-idf encodings. We choose the LinearSVC algorithm from the Scikit library with a one-versus-all strategy. This algorithm is characterized by low computation times and typically excellent results in text classification tasks [5].

As depicted in Table 1, our SVM achieves a F1-Macro score of 0.22 and F1-Micro score of 0.71. It's important to note that introducing n-grams beyond uni-grams tends to diminish the algorithm's performance. To address label-imbalances, we experiment with balancing the SVM by using the inverse label frequencies as weights. This results in a tradeoff: we achieve a higher F1-Macro score of 0.27 but a lower F1-Micro score of 0.56 compared to our unbalanced SVM.

Furthermore, we try to incorporate label hierarchies in our approach by using a Hierarchical Mixture of Experts approach, which is based on the principle of "divide and conquer". A large problem is divided into many smaller, easier to solve problems, whose combined solutions yield a solution to the complex problem [6]. As seen in Figure 2, we categorize labels into broader, coarse-grained categories and train an SVM for each category. The mapping of fine-grained labels to their respective coarse-grained categories is established based on the proposed categorization already presented by Wiegmann et al. [3]. Subsequently, for each subgroup, we train additional SVM "experts" whose sole responsibility is to classify text within the respective coarse-grained category. The combined outputs of all experts forms the final prediction. This approach is superior to be previous SVM configurations, as we manage to achieve scores of 0.27 F1-Macro and 0.72 F1-Micro. In each metric, it is at least as effective as, if not better than, our previous approaches.

Transformers

Training models on long text is a non-trivial task as they are limited by their input size. Despite state-of-the-art models like BigBird with large input sizes of 4096 tokens, it is often not feasible in practice to train them due to the large computational demands. Fine-tuning BigBird for this specific task would haven taken $\approx$ 27 days for one epoch on a NVIDIA Geforce 4060. Consequently, we explore three different transformer-based approaches, each constrained to an input size of 512 tokens.

Distribution Balanced Loss Function Re-sampling and Re-weighting are common methods to handle long-tailed class distribution. However, they often fall short in taking label dependencies into account. In our approach, we employ a distribution balanced loss across all of our transformers. By using this loss, we reduce redundant information of label co-occurence and explicitly assign lower weight on "easy-to-classify" negative instances [7]. This loss has already demonstrated its effectiveness on diverse long-tailed, textual multiclassification problems [8].

Truncated BERT In our initial approach, we fine-tune a pretrained BERT model on this multilabel classification task. We straightforwardly truncate documents longer than the maximum input size of 512 tokens. We achieve a score of 0.16 F1-Macro and 0.68 F1-Micro.

TextGuide In our second approach, we try to summarize documents based on important keywords. We adapt the TextGuide algorithm [9]. This entails the

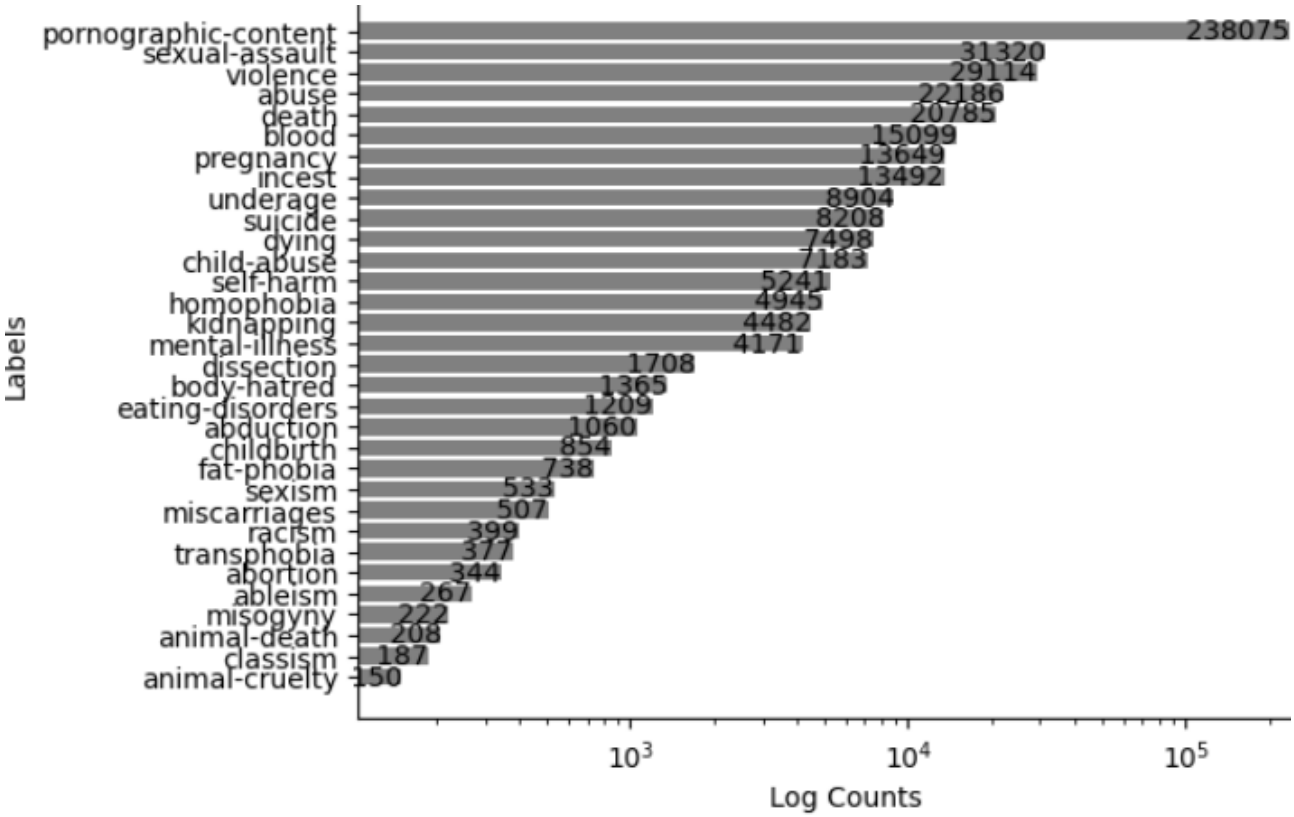

Figure 1: *Logarithmized plot of the label distribution in the training set. We can observe class imbalance, with the majority of texts being categorized as 'pornographic content'.*

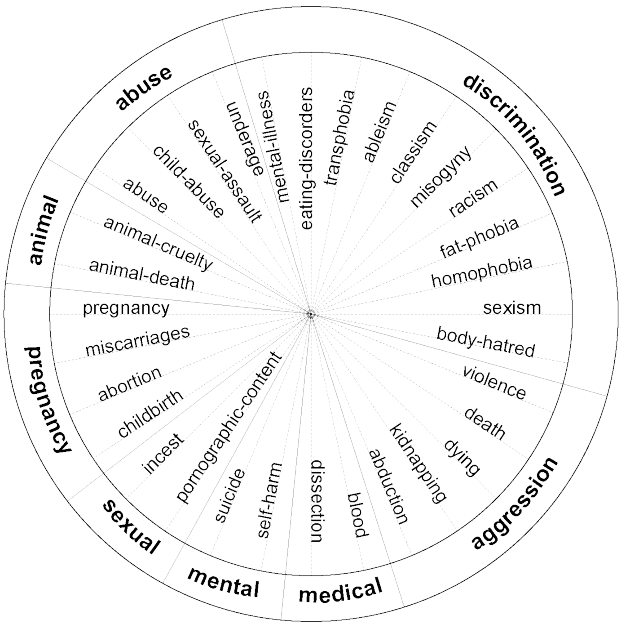

Figure 2: *Coarse- and fine-grained labels.*

extraction of features associated with the highes coefficients of our previously trained Support Vector Machine. These features correspond to important words in our text. Given our one-versus-all setting, we extract the top 250 words from each of the 32 support vector machines. To illustrate, for the label "homophobia", the top 5 words are gay, faggot, homophobic, fag, queer. If a text has less or equal 512 tokens, we keep the text as it is. However, if texts exceed this length, we extract the last 100 tokens from the end of the text. Subsequently, we iterate through our important words list, starting with the most important word of each class. Whenever we encounter a word in our list, we extract this word and 5 Token Neighbours and include them in our document summary. This continues until we either reach 512 tokens or we have exhausted our important words list. Afterwards, we fine-tune Bert on the summaries, yielding in a superior performance compared to the truncated version. Our TextGuide method achieves a remarkable F1-Macro score of 0.34 and a F1-Micro score of 0.75.

Max-Pooled DistillBert Encodings In our final approach, we implement a hierarchical strategy. We split each document into 512 token segments with an overlap of 32 tokens. Subsequently, we fine-tune DistillBert on each of the segments with the labels

Table 1: *Validation Scores of the Applied Approaches.*

Approach	F1-Macro	F1-Micro
PAN 2023 Baseline	0.2575	0.7274
SVM	0.2256	0.7150
SVM Balanced	0.2768	0.5644
Hierarchical SVM	0.2657	0.7201
Bert	0.1586	0.6754
TextGuide	0.3366	0.7465
PAN 2023 Top Solution	0.3720	0.7360
Max-Pooled Encodings	**0.4198**	**0.7613**

of respective document as target. After fine-tuning, we conduct an additional forward pass for all texts, during which we extract the embeddings of the class token from the last hidden layer. These embeddings serve as a summarization of the segment. In order to account for inhomogenious input sizes, we apply max-pooling on all embeddings belonging to the same document. Afterwards, we feed these max-pooled encodings into a 2-layer neural network, with 128 and 64 neurons in the first and second hidden layers, respectively. We achieve our final performance of 0.42 F1-Macro score and 0.76 F1-Micro score.

Conclusion

The final results are summarized in Table 1. It is evident that our SVM trained on tf-idf encodings already delivers competitive results. Including label hierarchies and employing Mixtures of Experts improves the performance even further. While our bag-of-words approach suffers from the loss of contextual information, it still outperforms our truncated transformer approach, as too much information is lost in the truncation. TextGuide overcomes this problem partially by focusing on important terms, leading to superior results compared to all of these approaches.

Finally, we use a hierarchical approach by classifying max-pooled segment encodings of a fine-tuned transformer. Although this approach is computationally more expensive, as we fine-tune on each segment of a text, we achieve results that, to the best of our knowledge, surpass all previously applied methods on this dataset.

Future Work

In forthcoming research, we would like to explore if the performance of our final model could be improved through the incorporation of hierarchical label structures. We argue that this could empower the model to exploit the inherent hierarchical nature of the data, which has already demonstrated success in support vector machines on this dataset, as well as in neural networks, as shown by Ruiz and Srinivasan [6].

Additionally, we intend to explore different configurations and settings of the TextGuide algorithm. Investigation of different numbers of Token Neighbours allows to assess the impact of different contextual scopes on the model's performance. Moreover, we plan to adapt the approach even more by incorporating complete sentences instead of only Token Neighbours. Although incorporating complete sentences would allow for less keyword extraction, as sentences typically contain more tokens than our number of Token Neighbours, this approach might contain less noise from splitted words and a more contextually rich representation.

Furthermore, given that sufficient hardware resources become available, we seek to perform a performance comparison with a large-scale model featuring increased input sizes, such as BigBird. The comparison will give us further insights on the potential advantages and trade-offs associated with scaling-up model complexity on this dataset.

References

[1] Umitcan Sahin, Izzet Emre Kucukkaya, and Cagri Toraman. *ARC-NLP at PAN 2023: Hierarchical Long Text Classification for Trigger Detection*. 2023. arXiv: 2307.14912 [cs.CL].

[2] Magdalena Wolska et al. *Trigger Warnings: Bootstrapping a Violence Detector for FanFiction*. 2022. arXiv: 2209.04409 [cs.CL].

[3] Matti Wiegmann et al. "Trigger Warning Assignment as a Multi-Label Document Classification Problem". In: Jan. 2023, pp. 12113–12134. DOI: 10.18653/v1/2023.acl-long.676.

[4] Vedangi Wagh et al. "Comparative Study of Long Document Classification". In: *TENCON 2021 - 2021 IEEE Region 10 Conference (TENCON)*. IEEE, Dec. 2021. DOI: 10.1109/tencon54134.2021.9707465. URL: https://doi.org/10.1109%2Ftencon54134.2021.9707465.

[5] Yasmen Wahba, Nazim Madhavji, and John Steinbacher. *A Comparison of SVM against Pre-trained Language Models (PLMs) for Text Classification Tasks*. 2022. arXiv: 2211.02563 [cs.CL].

[6] Miguel Ruiz and Padmini Srinivasan. "Hierarchical Text Categorization Using Neural Networks". In: *Information Retrieval* 5 (Jan. 2002), pp. 87–118. DOI: 10.1023/A:1012782908347.

[7] Yi Huang et al. *Balancing Methods for Multi-label Text Classification with Long-Tailed Class Distribution*. 2021. arXiv: 2109.04712 [cs.CL].

[8] Tong Wu et al. *Distribution-Balanced Loss for Multi-Label Classification in Long-Tailed Datasets*. 2021. arXiv: 2007.09654 [cs.CV].

[9] Krzysztof Fiok et al. "Text Guide: Improving the Quality of Long Text Classification by a Text Selection Method Based on Feature Importance". In: *IEEE Access* 9 (2021), pp. 105439–105450. DOI: 10.1109/access.2021.3099758. URL: https://doi.org/10.1109%2Faccess.2021.3099758.