ARC-NLP at PAN 2023: Hierarchical Long Text Classification for Trigger Detection

Umitcan Sahin[1], Izzet Emre Kucukkaya[1] and Cagri Toraman[1]

[1] *Aselsan Research Center, 06378, Ankara, Turkey.*

Abstract

Fanfiction, a popular form of creative writing set within established fictional universes, has gained a substantial online following. However, ensuring the well-being and safety of participants has become a critical concern in this community. The detection of triggering content, material that may cause emotional distress or trauma to readers, poses a significant challenge. In this paper, we describe our approach for the Trigger Detection shared task at PAN CLEF 2023, where we want to detect multiple triggering content in a given Fanfiction document. For this, we build a hierarchical model that uses recurrence over Transformer-based language models. In our approach, we first split long documents into smaller sized segments and use them to fine-tune a Transformer model. Then, we extract feature embeddings from the fine-tuned Transformer model, which are used as input in the training of multiple LSTM models for trigger detection in a multi-label setting. Our model achieves an F1-macro score of 0.372 and F1-micro score of 0.736 on the validation set, which are higher than the baseline results shared at PAN CLEF 2023.

Keywords

Trigger detection, Fanfiction, Transformer-based language models, Long text classification, Multi-label classification

1. Introduction

Fanfiction has been incredibly popular in recent years, especially among online communities. It is a lively and imaginative form of literary expression. It entails the development of fresh storylines, characters, and situations based on pre-existing fictional worlds, giving fans the chance to develop their favorite stories and pursue original, inventive concepts. As the Fanfiction community expands, there is an increasing need to address crucial issues related to participant safety and well-being.

The existence of triggering content is a major issue in the world of Fanfiction. Content that can cause extreme negative emotional reactions or the traumatization of people is referred to as triggering content. It might touch on issues like abuse, violence, mental health, or other upsetting topics. It is essential to create systems for properly identifying and managing triggering content given the diverse and frequently fragile character of Fanfiction readers and writers.

In this study, as Aselsan Research Center - Natural Language Processing team (ARC-NLP), we propose a method for trigger detection in long text documents using natural language

CLEF 2023 – Conference and Labs of the Evaluation Forum, September 18–21, 2023, Thessaloniki, Greece

ucsahin@aselsan.com.tr (U. Sahin); ekucukkaya@aselsan.com.tr (I. E. Kucukkaya); ctoraman@aselsan.com.tr (C. Toraman)

0000-0001-9594-3148 (U. Sahin); 0009-0006-2877-8713 (I. E. Kucukkaya); 0000-0001-6976-3258 (C. Toraman)

processing (NLP) and machine learning techniques. We seek to train a classification algorithm capable of precisely recognizing multiple triggering contents by using the concept of hierarchical recurrence over Transformer-based language models [1, 2, 3]. In our method, we first split long Fanfiction documents into smaller sized segments with an overlap between each consecutive segment. We use these segments to fine-tune a Transformer-based language model. Then, we extract each segment's feature embedding from the fine-tuned Transformer model, which are used in the training of multiple LSTM models. Finally, we combine the predictions of the trained LSTM models to generate trigger labels for multi-class and multi-label classification. To the best of our knowledge, we are first to use the techniques and methods described in this paper in the context of trigger detection in Fanfiction.

2. Task Description

In the context of trigger detection at PAN CLEF 2023 [4, 5], our objective is to assign warning labels to Fanfiction documents that may contain content capable of causing discomfort or distress (known as triggering content) [5, 6, 7]. Specifically, trigger detection is posed as a multi-label document classification task, aiming to assign the appropriate trigger warnings to each document without exceeding the necessary labels. It is important to note that all trigger warnings are determined from the perspective of the document's author, meaning that the author decides which specific trigger(s) the document contains. There are 32 distinct trigger labels including *pornographic-content*, *violence*, *death* and *sexual-assault* etc[1]. Each document can contain more than one trigger label, which leads to the need to adopt a multi-class and multi-label approach.

3. Dataset

Table 1
Trigger detection dataset shared at PAN CLEF 2023. The number of documents, average number of words in each document and presence of labels for *training*, *validation*, and *test* sets are given.

	#Document	#Avg. Words	Labels
Train	307,102	2,350	✓
Validation	17,104	2,336	✓
Test	17,040	2,338	-

The dataset for trigger detection at PAN CLEF 2023 comprises of fanfiction pieces sourced from archiveofourown.org (AO3) [6, 7]. Each document falls within the range of 50 to 6,000 words and is accompanied by one or more trigger labels. Table 1 shows the number of documents, average number of words, and presence of labels for training, validation, and test sets. As mentioned before, the label set encompasses 32 distinct trigger warnings, exhibiting a distribution where certain labels are frequently encountered while the majority of labels are relatively uncommon.

[1] https://pan.webis.de/clef23/pan23-web/trigger-detection.html

Table 2 shows the class distribution ratios with respect to 32 trigger labels in the training and validation sets. As seen, the classes are greatly imbalanced. Furthermore, we also note that the class distributions between the training and validation sets are very similar to each other, which suggests that they come from the same distribution.

Table 2
Class distribution ratios with respect to 32 trigger labels in the training and validation sets.

	Class	Train	Validation
1	*pornographic*	77.52%	77.33%
2	*violence*	9.48%	9.46%
3	*death*	6.77%	6.75%
4	*sexual-assault*	10.20%	10.18%
5	*abuse*	7.22%	7.21%
6	*blood*	4.92%	4.90%
7	*suicide*	2.67%	2.67%
8	*pregnancy*	4.44%	4.44%
9	*child-abuse*	2.34%	2.34%
10	*incest*	4.39%	4.38%
11	*underage*	2.90%	2.89%
12	*homophobia*	1.61%	1.61%
13	*self-harm*	1.71%	1.71%
14	*dying*	2.44%	2.44%
15	*kidnapping*	1.46%	1.45%
16	*mental-illness*	1.36%	1.36%
17	*dissection*	0.56%	0.55%
18	*eating-disorder*	0.39%	0.40%
19	*abduction*	0.35%	0.34%
20	*body-hatred*	0.44%	0.44%
21	*childbirth*	0.28%	0.28%
22	*racism*	0.13%	0.13%
23	*sexism*	0.17%	0.17%
24	*miscarriage*	0.16%	0.17%
25	*transphobia*	0.12%	0.12%
26	*abortion*	0.11%	0.12%
27	*fat-phobia*	0.24%	0.24%
28	*animal-death*	0.06%	0.07%
29	*ableism*	0.09%	0.09%
30	*classism*	0.06%	0.06%
31	*misogyny*	0.07%	0.08%
32	*animal-cruelty*	0.05%	0.05%

4. Proposed Method: Hierarchical Recurrence over Transformer-based Language Model

In this section, we describe our method of hierarchical recurrence over Transformer-based model for long text classification of multi-label trigger detection. The diagram of the model is shown in Figure 1. As shown in the figure, we divide our methodology into four parts: 1) Segmentation, 2) Tokenization, 3) Feature extraction, and 4) Model training, which are explained in detail below.

4.1. Segmentation

According to the information provided in Table 1, the average word count in each document within the training set exceeds 2000. Traditional Transformer-based language models such as BERT [8] employ tokenizers with a maximum length of 512 tokens and typically truncate the remaining text. However, this approach is inadequate for accurately classifying long documents since crucial information may be omitted through truncation, resulting in subpar classification performance. Therefore, we follow a similar approach to [1, 2, 3, 9] in our segmentation method. We first apply text processing to the documents by

- removing HTML tags,
- removing URLs,
- removing English stop words [10], and
- lower-casing all text.

Then, we split the processed document into segments (i.e., text chunks) of 200 words (i.e., $w_0, ..., w_{200}$) with an overlap of 50 words between each consecutive segment as shown in Figure 1. We assign the original document label to each segment. In other words, we represent each document in the training set with a variable-length sequence of 200-word segments where each segment is assigned the original document label.

4.2. Tokenization

For tokenization, we fine-tune a Transformer-based RoBERTa model [11] using the sequence of segments obtained after our segmentation method. We use the base-version of the model at HuggingFace[2] with a learning rate of $1e-5$, epoch number 3, and training batch size of 8. We also use the corresponding RoBERTa model's tokenizer with a maximum length of 256 tokens. After the tokenization step, we represent each 200-word segment by the corresponding tokens with size 256 (i.e., $t_0, ..., t_{255}$ in Figure 1).

4.3. Feature Extraction

For this method, we feed forward the segment tokens obtained after our tokenization method to the fine-tuned RoBERTa model and construct segment embeddings. We extract the embedding of the CLS token (i.e., a special classification token used for classification tasks) from the last

[2] https://huggingface.co/roberta-base

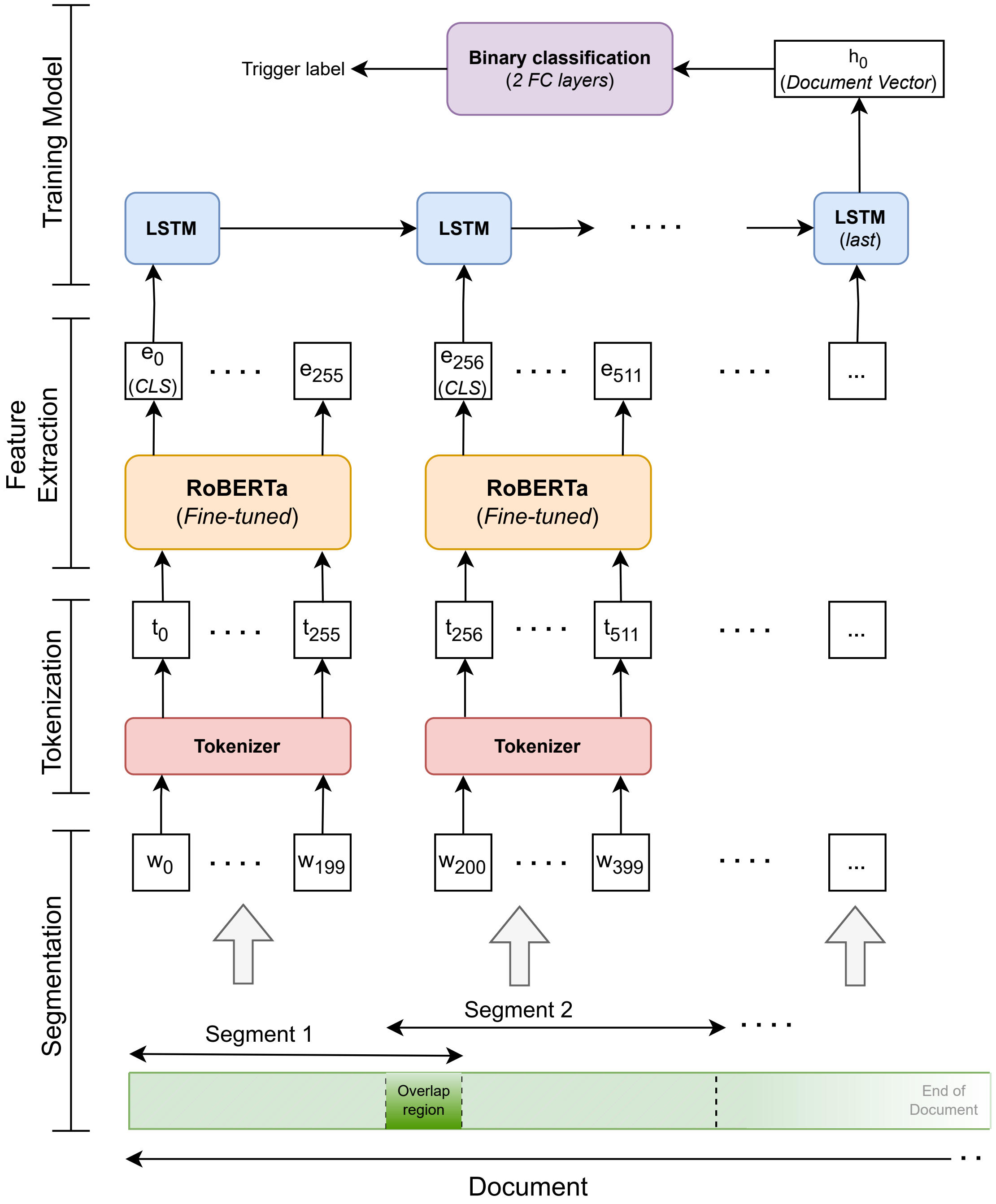

Figure 1: Hierarchical recurrence over Transformer-based model diagram for trigger detection. Each document is split into 200-word segments with a 50-word overlap between the consecutive segments. ROBERTa [11] along with its tokenizer (256 tokens) is used as the Transformer model and fine-tuned using all the segmented documents in the training set. Transformer model and its tokenizer are shared throughout the network. CLS embeddings are extracted for each segment of a given document, which are then used as input feature vectors for the training of the LSTM model. For the classifier, two fully-connected linear layers with ReLU activation and binary cross-entropy (BCE) loss are used.

hidden layer of the fine-tuned RoBERTa model for each segment and use them as our feature vectors, which are input to the LSTM model.

4.4. Model Training

After feature extraction, we train a single layer LSTM network with a hidden unit size of 100 and a batch size of 8. We use Pytorch's implementation of the Stochastic Gradient Descent (SGD) optimizer with learning rate of 0.01 [12]. In the classification step, we use two fully connected linear layers with sizes 100 and ReLU [13] as the activation function between them. Furthermore, we broke down the multi-label trigger detection problem into a multiple binary classification problem. Therefore, we train the aforementioned LSTM model for each trigger class (32 in total) in a one-vs-all classification setting. Finally, we use Binary Cross Entropy (BCE) as our loss function.

As seen in Table 2, the trigger classes are greatly imbalanced. For instance, while the 77.52% of the training data includes *pornographic* content, only the 0.05% of it consists of *animal cruelty*. There are many methods such as oversampling the underrepresented classes, that are proposed to solve the class imbalance problem for deep neural networks [14]. In this study, we solve this problem by changing the weights of the underrepresented classes during the back-propagation updates. We do this by incorporating a weight of positive examples for the loss function. For example, if a dataset contains 20 positive and 1000 negative examples of a single class, then weight for the positive class is assigned as $1000/20 = 50$ in the loss function. In this way, the loss would act as if the dataset contains $20 \times 50 = 1000$ positive examples. In our final model, we assign positive class weights to the loss function for the trigger classes from 15 (i.e., *kidnapping*) to 32 (i.e., *animal-cruelty*). For the first 14 trigger classes, we do not assign any positive class weights to BCE loss function.

Finally, for each trigger class, we train the LSTM model up to 10 epochs and save the best performing model with respect to the highest F1 scores achieved on the validation set. We then combine the predictions of the 32 trained LSTM models to produce our final multi-label trigger labels.

5. Experiments

In this section, we describe the baseline methods used for multi-label trigger detection in Fanfiction, and share the classification performance of our model on the validation set.

5.1. Baselines

- **BERT**: We use HuggingFace's implementation of the Transformer-based BERT model[3] and fine-tune this model with the training set for trigger detection. We set learning rate to $1e-5$, number of training epochs to 5, and batch size to 8. We use multi-label classification layer with 32 classes on top of the BERT model. We use the corresponding BERT tokenizer with maximum length of 512 tokens, truncating the rest of the document in the training set.

[3] https://huggingface.co/bert-base-uncased

Table 3
F1-macro and F1-micro scores of our model and baseline methods. The scores are computed on the validation set provided by PAN CLEF 2023.

Model	F1-macro	F1-micro
BERT	0.0471	0.4607
RoBERTa-Segment	0.1869	0.6958
TF-IDF + XGBoost (task baseline)	0.2575	0.7274
Hierarchical LSTM + RoBERTa (ours)	**0.3720**	**0.7360**

- **RoBERTa-Segment**: We use the same RoBERTa model that is fine-tuned with the segmented tokens described in Section 4.
- **TFIDF+XGBoost**: This baseline is shared by the organizers at PAN CLEF 2023 for trigger detection [7][4]. It uses Gradient Boosted Trees [15] based on a TF-IDF [16] document vectors.

5.2. Validation Results

Multi-label F1-macro and F1-micro scores serve as the primary performance metrics for trigger detection at PAN CLEF 2023[5]. To assess the effectiveness of our model and the baselines, we compute these scores on the validation set, and the results are presented in Table 3. Notably, our model outperforms all others in terms of multi-label F1-macro and F1-micro scores.

The limitations of the BERT model become evident as it struggles due to the constrained token size of 512. Truncating long documents reduces its ability to capture triggering content effectively. While our RoBERTa-Segment baseline improves upon BERT, it still falls short in performance since it lacks the crucial hierarchical recurrence concept integrated into its core architecture.

Among the baselines, the TFIDF+XGBoost approach achieves the highest F1-macro and F1-micro scores. Despite the absence of contextual information, XGBoost compensates by leveraging TF-IDF vector representations, enabling comprehensive coverage of tokens throughout the entire document.

These results underscore the existence of ample room for improvement in the non-trivial task of multi-label trigger detection in Fanfiction. It is evident that further improvements are necessary to enhance the effectiveness of trigger detection methods and address the challenges associated with this task.

5.3. Class-based Validation Results

Table 4 shows the binary classification performances of the trained models for each trigger class on the validation set. It is worth noting that by incorporating a positive class weight into the loss function for the classes from 15 to 32, where positive instances are particularly scarce, the

[4] https://github.com/pan-webis-de/pan-code/tree/master/clef23/trigger-detection/baselines
[5] https://pan.webis.de/clef23/pan23-web/trigger-detection.html

model's capability to predict positive class instances is significantly enhanced. Consequently, this leads to an anticipated improvement in the overall performance of multi-label classification.

Table 4
Various binary classification scores of the proposed method (32 hierarchical LSTM models over Transformer-based RoBERTa) on the validation set for trigger detection. Pos. Ratio indicates the true positive class ratio in the validation set. Pos. Pred. Ratio indicates the positive class ratio predicted by the proposed method. The classification performance computed in terms of macro and micro F1, Precision (P), and Recall (R) scores. Overall multi-label classification performances are given at the bottom.

Class	Pos. Ratio	F1-macro	P-macro	R-macro	F1-micro	P-micro	R-micro	Pos. Pred. Ratio
pornographic	0.773	0.901	0.906	0.897	0.932	0.932	0.932	0.781
violence	0.095	0.715	0.744	0.694	0.912	0.912	0.912	0.074
death	0.068	0.781	0.802	0.763	0.948	0.948	0.948	0.058
sexual-assault	0.102	0.748	0.784	0.722	0.918	0.918	0.918	0.078
abuse	0.072	0.727	0.768	0.699	0.936	0.936	0.936	0.052
blood	0.049	0.758	0.791	0.733	0.959	0.959	0.959	0.039
suicide	0.027	0.797	0.841	0.764	0.981	0.981	0.981	0.021
pregnancy	0.044	0.882	0.883	0.881	0.980	0.980	0.980	0.044
child-abuse	0.023	0.726	0.751	0.705	0.977	0.977	0.977	0.019
incest	0.044	0.837	0.835	0.839	0.973	0.973	0.973	0.044
underage	0.029	0.681	0.744	0.645	0.971	0.971	0.971	0.017
homophobia	0.016	0.711	0.750	0.682	0.984	0.984	0.984	0.012
self-harm	0.017	0.795	0.871	0.745	0.989	0.989	0.989	0.011
dying	0.024	0.678	0.735	0.644	0.975	0.975	0.975	0.015
kidnapping	0.015	0.618	0.791	0.576	0.986	0.986	0.986	0.004
mental-illness	0.014	0.598	0.566	0.787	0.942	0.942	0.942	0.062
dissection	0.006	0.583	0.552	0.771	0.971	0.971	0.971	0.029
eating-disorder	0.004	0.756	0.700	0.858	0.995	0.995	0.995	0.007
abduction	0.003	0.576	0.548	0.697	0.985	0.985	0.985	0.014
body-hatred	0.004	0.639	0.595	0.765	0.988	0.988	0.988	0.012
childbirth	0.003	0.683	0.621	0.882	0.993	0.993	0.993	0.009
racism	0.001	0.605	0.633	0.587	0.998	0.998	0.998	0.001
sexism	0.002	0.577	0.563	0.599	0.996	0.996	0.996	0.003
miscarriage	0.002	0.694	0.662	0.741	0.997	0.997	0.997	0.003
transphobia	0.001	0.722	0.682	0.785	0.998	0.998	0.998	0.002
abortion	0.001	0.531	0.523	0.549	0.997	0.997	0.997	0.002
fat-phobia	0.002	0.780	0.780	0.780	0.998	0.998	0.998	0.002
animal-death	0.001	0.558	0.545	0.583	0.998	0.998	0.998	0.001
ableism	0.001	0.535	0.538	0.533	0.999	0.999	0.999	0.001
classism	0.001	0.500	0.500	0.500	0.999	0.999	0.999	0.000
misogyny	0.001	0.501	0.503	0.604	0.976	0.976	0.976	0.024
animal-cruelty	0.001	0.505	0.505	0.658	0.982	0.982	0.982	0.018
Multi-label	-	**0.3720**	0.3920	0.4330	**0.7360**	0.7330	0.7400	-

5.4. Test Results

We submitted our model as a dockerized image to the TIRA system [17]. The test was conducted on a hardware configuration consisting of a single CPU Core, 10GB of RAM, and a single Nvidia GTX 1080 with 8GB. The test completion time was approximately 150 minutes. The final test results of all the participants for PAN CLEF 2023 Trigger Detection are presented in Table 5.

Table 5
The leaderboard in terms of the F1-macro and F1-micro test scores of all participants at PAN CLEF 2023 Trigger Detection. Our team's name is ***pan23-transformers***. *trigger-detection-baseline* is the TF-IDF+XGBoost model explained in Section 5.

Team	F1-macro	F1-micro
pan23-transformers (Ours)	**0.352**	0.737
pan23-supergirl	0.350	**0.753**
trigger-detection-baseline	0.301	0.689
pan23-jojo-no-kimyou-na-bouken	0.228	0.557
pan23-marvel-cinematic-universe	0.225	0.616
pan23-sherlock	0.161	0.402
pan23-game-of-thrones	0.048	0.625

trigger-detection-baseline is the TFIDF+XGBoost model explained in Section 5. Furthermore, only the submission with the highest F1-macro score was included for teams with multiple submissions. At the end, our team, named ***pan23-transformers***, achieved first place in terms of the multi-label F1-macro score and second place in terms of the multi-label F1-micro score in the leaderboard with our hierarchical recurrence over Transformer-based language model.

6. Conclusion

This study presents an approach for detecting triggers in Fanfiction by employing natural language processing (NLP) and machine learning techniques. Our objective is to train a classification algorithm capable of accurately identifying multiple instances of triggering content. In our method, we initially break down lengthy Fanfiction documents into smaller segments, ensuring an overlap between consecutive segments. These segments are then used to fine-tune a Transformer-based language model. From this fine-tuned model, we extract feature embeddings for each segment, which serve as inputs for training multiple LSTM models. Subsequently, the predictions of these trained LSTM models are combined to generate trigger labels for multi-class and multi-label classification. We show that our method that is based on hierarchical recurrence over Transform-based model achieves better classification performance than the baselines used for multi-label trigger detection in Fanfiction. Our model ranks first in terms of the multi-label F1-macro score and second in terms of the multi-label F1-micro score on the test set for PAN CLEF 2023 Trigger Detection. Furthermore, Our experimental findings strongly indicate that conventional NLP techniques, such as TF-IDF document vectorization and Transformer-based models with standard tokenization limits (typically set at a maximum length of 512 tokens), exhibit limited performance in the context of multi-class and multi-label classification tasks, particularly when dealing with lengthy documents. These techniques often struggle to effectively handle the complexities associated with the simultaneous prediction of multiple trigger labels in scenarios where extensive text is involved.

References

[1] R. Pappagari, P. Zelasko, J. Villalba, Y. Carmiel, N. Dehak, Hierarchical transformers for long document classification, in: 2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU), IEEE, 2019, pp. 838–844.

[2] X. Dai, I. Chalkidis, S. Darkner, D. Elliott, Revisiting transformer-based models for long document classification, in: Findings of the Association for Computational Linguistics: EMNLP 2022, Association for Computational Linguistics, Abu Dhabi, United Arab Emirates, 2022, pp. 7212–7230.

[3] H. Park, Y. Vyas, K. Shah, Efficient classification of long documents using transformers, in: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), Association for Computational Linguistics, Dublin, Ireland, 2022, pp. 702–709. doi:10.18653/v1/2022.acl-short.79.

[4] J. Bevendorff, I. Borrego-Obrador, M. Chinea-Ríos, M. Franco-Salvador, M. Fröbe, A. Heini, K. Kredens, M. Mayerl, P. Pęzik, M. Potthast, F. Rangel, P. Rosso, E. Stamatatos, B. Stein, M. Wiegmann, M. Wolska, , E. Zangerle, Overview of PAN 2023: Authorship Verification, Multi-Author Writing Style Analysis, Profiling Cryptocurrency Influencers, and Trigger Detection, in: A. Arampatzis, E. Kanoulas, T. Tsikrika, A. G. Stefanos Vrochidis, D. Li, M. Aliannejadi, M. Vlachos, G. Faggioli, N. Ferro (Eds.), Experimental IR Meets Multilinguality, Multimodality, and Interaction. Proceedings of the Fourteenth International Conference of the CLEF Association (CLEF 2023), Lecture Notes in Computer Science, Springer, 2023.

[5] M. Wiegmann, M. Wolska, M. Potthast, B. Stein, Overview of the Trigger Detection Task at PAN 2023, in: M. Aliannejadi, G. Faggioli, N. Ferro, M. Vlachos (Eds.), Working Notes of CLEF 2023 - Conference and Labs of the Evaluation Forum, CEUR-WS, 2023.

[6] M. Wolska, C. Schröder, O. Borchardt, B. Stein, M. Potthast, Trigger warnings: Bootstrapping a violence detector for fanfiction, arXiv preprint arXiv:2209.04409 (2022).

[7] M. Wiegmann, M. Wolska, C. Schröder, O. Borchardt, B. Stein, M. Potthast, Trigger Warning Assignment as a Multi-Label Document Classification Problem, in: Proceedings of the 61th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Toronto, Canada, 2023.

[8] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, arXiv preprint arXiv:1810.04805 (2018).

[9] O. Ozcelik, C. Toraman, Named entity recognition in Turkish: A comparative study with detailed error analysis, Information Processing Management 59 (2022) 103065. doi:https://doi.org/10.1016/j.ipm.2022.103065.

[10] S. Bird, E. Klein, E. Loper, Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit, " O'Reilly Media, Inc.", 2009.

[11] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, V. Stoyanov, Roberta: A robustly optimized BERT pretraining approach, CoRR abs/1907.11692 (2019). arXiv:1907.11692.

[12] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, A. Lerer, Automatic differentiation in pytorch, in: NIPS-W, 2017.

[13] A. F. Agarap, Deep learning using rectified linear units (relu), CoRR abs/1803.08375 (2018).

arXiv:1803.08375.

[14] M. Buda, A. Maki, M. A. Mazurowski, A systematic study of the class imbalance problem in convolutional neural networks, Neural networks 106 (2018) 249–259.

[15] T. Chen, C. Guestrin, Xgboost: A scalable tree boosting system, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016, pp. 785–794.

[16] G. Salton, M. McGill, Introduction to Modern Information Retrieval, McGraw-Hill Book Company, 1984.

[17] M. Fröbe, M. Wiegmann, N. Kolyada, B. Grahm, T. Elstner, F. Loebe, M. Hagen, B. Stein, M. Potthast, Continuous Integration for Reproducible Shared Tasks with TIRA.io, in: J. Kamps, L. Goeuriot, F. Crestani, M. Maistro, H. Joho, B. Davis, C. Gurrin, U. Kruschwitz, A. Caputo (Eds.), Advances in Information Retrieval. 45th European Conference on IR Research (ECIR 2023), Lecture Notes in Computer Science, Springer, Berlin Heidelberg New York, 2023, pp. 236–241.