

FoSIL at PAN'23: Trigger Detection with a Two Stage Topic Classifier

Notebook for PAN at CLEF 2023

Jenny Felser^{1,*}, Christoph Demus¹, Dirk Labudde¹ and Michael Spranger¹

¹Mittweida University of Applied Sciences, Technikumplatz 17, Mittweida, 09648, Germany

Abstract

Fanfiction platforms become very popular. However, since fan fiction stories can also contain content that can be disturbing to readers, it is important to assign appropriate warnings to them. The automatic assignment of 32 trigger labels to fanfiction works is addressed by the Trigger Detection task of PAN'23 in terms of a multi-label document classification. This paper presents a two-stage approach in which the final multi-label classification was preceded by a pre-classifier that predicted newly formed upper classes of trigger warnings. For both classifiers, features based on fastText word-embeddings and semi-supervised topic modeling were used. Multi-Layer-Perceptron (MLP) was used as classifier in both stages, and its performance was compared with Random Forest (RF) for the first classifier. The applicability of the two-step approach was shown in a comparison with a traditional one-step procedure. The best model achieved a micro F1-score of 0.54.

Keywords

trigger detection, multi-label document classification, semi-supervised topic modeling, Multi-Layer-Perceptron, word embeddings

1. Introduction

Fanfiction web forums, where fans can write and publish stories inspired by existing fictional works such as books, films, TV series or cartoons [1], are becoming increasingly popular. For instance, the fanfiction platform Archive of Our Own (AO3) currently, as of May 2023, has over five million users and hosts over eleven million works in approximately 58,000 fandoms (sub-categories) [2]. One reason for this popularity is that fanfiction platforms allow inexperienced authors to publish stories without the pressure of earning money and being well appreciated by a broad mass of society [3]. However, this also implies that fanfiction stories more often contain scenarios with physical, emotional and sexual violence, even if these were not included in the original work [4], as is the case with many Harry Potter fanfiction works, for example [5].


To protect sensitive readers for whom such content may evoke negative emotions, especially those who have experienced traumatic events, trigger warnings have been suggested [6]. According to Cambridge Dictionary, those are defined as “[...] a statement at the start of a piece


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

*Corresponding author.

✉ jfelser@hs-mittweida.de (J. Felser); cdemus@hs-mittweida.de (C. Demus); labudde@hs-mittweida.de (D. Labudde); spranger@hs-mittweida.de (M. Spranger)

ORCID 0000-0003-4319-3175 (C. Demus); 0000-0002-6780-0841 (M. Spranger)

 © 2023 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

 CEUR Workshop Proceedings (CEUR-WS.org)

of writing, video, etc. warning people that they may find the content very upsetting [...]” [7]. Writers are, on the platform AO3 for example, encouraged to add trigger warnings to their stories themselves, if necessary [8]. However, AO3, also offers the “no warning” option [1], so not all stories come with appropriate warnings. Furthermore, the perception of whether content is emotionally upsetting is often subjective [5], so an author may choose not to warn readers even though some readers may find the story upsetting.

One way to address these problems is to automatically assign trigger warnings to fanfiction stories. This is one mission of the PAN’23 competition [9]. The task called Trigger Detection consists of automatically assigning all appropriate trigger labels from a set of 32 labels to fanfiction works in terms of a multi-label document classification [10].

In this paper the detection of trigger warnings is divided into two stages: First, MLP and RF were compared to predict newly formed superclasses of trigger warnings. Then, the predictions were used as features for the final classification using 32 binary MLPs. In addition, features based on fastText and semi-supervised topic modeling were considered for both classifiers.

The paper is structured as follows: First, related work is presented in Section 2. An overview of the data is given in Section 3, followed by the description of our methodology in Section 4. Then our results are presented and discussed in Section 5. Finally, a brief conclusion and an outlook on future development are given in Section 6.

2. Related Work

To the extent of our knowledge, the only work that has addressed the assignment of trigger warnings to fanfiction or more generally to fictional texts, was presented by Wolska et al. [1]. In this work, the authors demonstrated that a Support Vector Machine (SVM) with n-gram features can already achieve promising results and even outperforms the state-of-the-art model Bidirectional Encoder Representations from Transformers (BERT) [11]. However, they only focused on a single trigger type, i.e. violence, and hence performed a binary classification, whereas for the PAN-2023 competition, trigger detection is designed as a multi-label document classification [10].

The task of multi-label classification has so far been addressed, for example, as a hierarchical multi-label classification of blurbs into genres [12] or in the field of detection of offensive language and hate speech [e.g. 13, 14]. The second mentioned task is in some ways similar to trigger detection, since the warnings used to label the fanfiction stories include several sub-forms of discrimination, such as racism, sexism, and transphobia, as well as abuse, including verbal abuse [8]. Approaches to multi-label classification in this domain included using classifiers such as neural networks that can be directly applied to the multi-label problem [13, 14], transforming the problem into multiple binary classification problems [15, 16, 17], or using classifier chains [18], first described in [19, 20]. The latter has shown promising results, for example in the multi-label classification of abusive language in German tweets [18], but is accompanied by the difficulty of having to determine the order of the classifiers in the chain [21]. Besides, a hierarchical approach has been proposed by some authors [17, 22, 23] for multi-label classification, where the final classifier is preceded by a binary classifier. For example, Prabowo et al. [22] proposed five different hierarchies for multi-label classification of hate speech in Indonesian, in most

of which a distinction was first made between hate speech and abusive language, and then different subtypes of hate speech were classified. Although this work is similar to our approach in terms of hierarchical approach, two major differences exist: While in the work by Prabowo et al. [22] as well as furthermore by Joshi et al. [17] a binary classifier was used in the first stage, here a multi-label classifier was applied instead. Furthermore, Prabowo et al. [22] as well as Joshi et al. [17] trained the second classifier exclusively on documents that had been classified as hate speech in the first stage. In contrast, in the work described here, the second classifier was trained on the entire dataset and the predictions of the first classifier were used as features. This ensured that incorrect results in the first stage had less impact on the final result.

Furthermore, in addition to the approach to multi-label classification, the features used are also of high importance. In this work, the focus was on the use of word embeddings and topic modeling for feature engineering, so the following description of similar work will also focus on these two features. Word embedding based features have again been applied for the detection of hate speech respectively offensive speech [e.g. 24, 25, 26, 27]. They have an advantage over common Bag of Words (BoW) features in that they take into account semantic and syntactic meaning of words [28]. Consequently, features based on fastText word embeddings were shown to outperform traditional features such as word n-grams weighted by TF-IDF in binary [24] and fine granular offensive language classification of German tweets [18]. Some previous work [e.g. 26, 29] has used word embeddings trained on large external corpora consisting of, for example, Wikipedia articles [30]. However, as Lai et al. [31] demonstrated in a comprehensive study, training word embeddings on a dataset of a domain that differs considerably from the training corpus for the classification task can have a negative impact on text classification performance. Since the training dataset in this work consisted of fictional texts [8], as opposed to the underlying datasets of most of the pre-trained word embeddings, the word embeddings used here were trained analogously to the trigger warning classifiers based on the fanfiction documents provided by the PAN'23 organisers, so we used an approach similar to, e.g. Niemann [18].

Topics have been widely used as features for automatic classification, including detection of mental illness and psychosocial risk in texts such as diary entries and posts on social media [e.g. 32, 33, 34]. Research in this area in particular is also relevant to the task of trigger detection, as it shares similarities with the identification of trigger warnings such as suicide, self-harm, mental illness, and eating disorders [8]. Most previous work used topic probabilities of documents as features [e.g. 33, 34, 35] obtained by applying topic modeling algorithms such as Latent Dirichlet Allocation (LDA) [e.g. 33, 34], originally proposed by Blei et al. [36] and Latent Semantic Analysis (LSA) [e.g. 34, 35] as described by Deerwester et al. [37]. Topic features were found to be particularly successful with respect to detecting depression [34, 38], suicide risk [32, 39], as well as predicting re-hospitalization for mental illnesses [35]. A limitation of these approaches, however, is that they all used unsupervised topic modeling algorithms for feature extraction. Thus, Churchill et al. [40] points out that unsupervised topic models do not necessarily produce the desired topics. Rather, for the purpose of text classification, it would be desirable if the generated topics were related to the known categories of the dataset. Addressing this problem, an important contribution of this work is to use semi-supervised topic modeling, as proposed by Lu et al. [41], to generate the topics related to the categories.

Table 1

Manual Grouping of the trigger warnings into six upper classes by its meanings.

Coarse topic	Trigger warnings
birth/ pregnancy discrimination	childbirth, abortion, pregnancy, miscarriages racism, sexism, transphobia, fat-phobia, ableism, classism, misogyny, homophobia
impairment violence	mental-illness, eating-disorders, body-hatred violence, dissection, abduction, animal-cruelty, abuse, child-abuse, self-harm, blood, kidnapping
death	death, animal-death, dying, suicide
sex	sexual-assault, incest, underage, pornographic-content

3. Data Description

The Webis Trigger Warning Corpus 2022 (Webis-Trigger-22), created by Wiegmann et al. [8], constituted the underlying data foundation for the Trigger Detection task. The data consists of a training set (307,102 documents), validation set (17,104 documents) and test set (17,040 documents), where the documents have lengths from 50 to 6,000 words. The documents are labelled with 32 possible trigger warnings. It is a multi-label task, which means that each document is provided with zero, one or more labels. The label “pornographic-content” is with 238,075 occurrences in the training set much more dominating in the corpus than all other labels. The second most frequent label is “sexual-assault” with 31,320 documents. Twelve labels occur less than 1,000 times in the training set, with the “animal-cruelty” label being the rarest with 150 occurrences.

A manual review of the 32 labels showed that many of them are very similar having just slight differences in its meanings. Therefore, the labels were manually summarized into six super classes (Table 1) which are later used as additional features in different ways. Thereby every label was assigned to exactly one of the coarse topics.

4. Methodologies

To solve the task of multi-label classification of trigger warnings, two different approaches were compared. In both cases, fastText word embeddings [42] and semi-supervised topic modeling [41] served as features. The first approach formed our baseline. Here, 32 MLPs were each trained as a binary classifier for one of the 32 trigger warnings using only the previously mentioned features (MLP_S). In contrast, the second approach implemented a two-stage process that results in runs MLP_E1 and MLP_E2. During the first stage, we calculate the probability of each text belonging to each of the superclasses described in Section 3. Here we compared to methods, namely MLP and RF. In the second stage, the predictions of the first stage were considered as additional features for the final multi-label classification using the 32 MLPs. The next subsections describe the methods in more detail. The entire workflow that led to the submitted results is outlined in Figure 1.

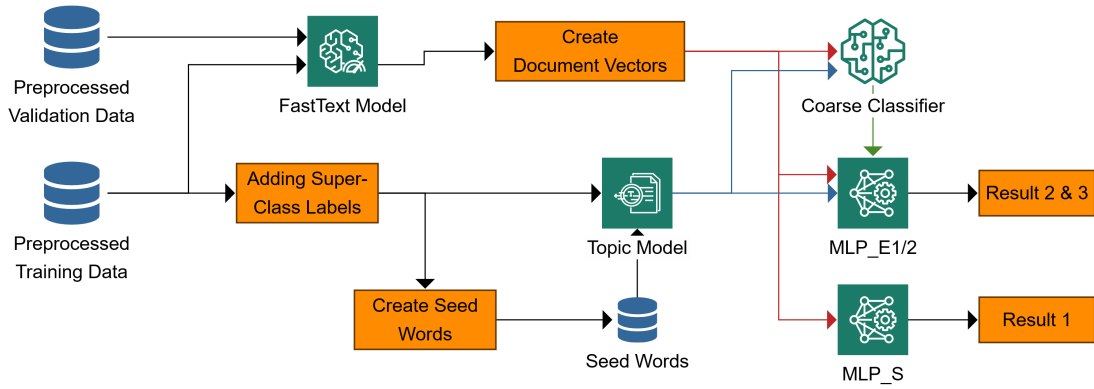


Figure 1: Workflow of the proposed approaches that led to the three submitted runs.

4.1. Data Preprocessing

First, several pre-processing steps were performed on the given datasets. These included normalising the texts by expanding contractions and converting all words to lower case. Furthermore, HTML tags, escaped characters, URLs, punctuation marks and stopwords were removed. Words were lemmatized to be able to represent inflected forms that have the same meaning as a single unit and to reduce sparsity [43]. Numbers were replaced by the word “number”, as specific numbers might not be relevant but the knowledge of an occurrence of a number in general could point to different trigger warnings.

4.2. Creation of Document Vectors

As a basic feature representation document vectors were determined from fastText word-embeddings [44]. Therefore, a fastText skipgram-model with a window size of five and minimum word count of three was trained on the preprocessed training- and validation data to represent the words as 200-dimensional word vectors. fastText was used as it has the advantage to be able to also represent out-of-vocabulary-words by using character n-grams in the background [44]. From the word-embeddings, document vectors d were computed by summing up the word vectors v multiplied with its term frequencies tf of the preprocessed documents[45] (see Equation (1)):

$$d_i = \sum_{j=1}^n (tf_j \cdot v_j) \quad (1)$$

Finally, the document vectors were normalized by scaling each vector to unit norm.

4.3. Guided Topic Modelling

To generate additional features to the fastText word-embeddings a semi-supervised topic model was trained with the goal to separate superclasses described in Section 3. A semi-supervised topic model was used to guide the model which topics to separate. It was trained based on

Table 2

Extracted bi-grams used as seed words for semi-supervised topic modeling.

Coarse topic	Seed words
birth/ pregnancy	baby shoe, pregnancy now, male carrier, may pregnant, baby gender, maternity pant
discrimination impairment	officer bower, faggot word, mook knife, nash look, want fag, cecil belly, blood doll spectrum disorder, know bipolar, binder feel, trauma holder, know autistic, purge night
violence death	macaroni fuck, bubble bass, bitter root, psycho delic, robber girl quirkless useless, funeral want, baby assassin, drooly loser, concept death, stealth shadow
sex	leg spread, now want, wrap around, around cock, finger inside, wrap lip, swirl tongue, mouth kiss, woundfucker number

word-bi-grams as they contain more information and can better capture the meaning of the texts compared to unigrams. The semi-supervised bi-gram topic model requires only few seed bi-grams describing each topic as input. Therefore, a seed bi-gram list for every superclass had to be created. For simplicity, the word bi-grams will be referred to as words in the following, even though word bi-grams are meant. In the following, first the extraction of seed bi-grams is described before details about the topic model are introduced.

The main challenge in this part was the imbalance of the dataset and in particular the dominance of the label “pornographic-content”, to which more than 75% of the texts belong. In fact, there is no trigger warning that did not occur at least once at the same time as this one. This leads to the assumption that there is a fraction of the vocabulary shared by “pornographic content” with every other trigger warning. To address this problem, a background corpus containing all pre-processed documents labeled only “pornographic-content” was created which was later used to remove noise in other documents caused by pornographic content related vocabulary. In a second step, six folds of documents were created - one fold for each of the superclasses. Each fold contained all documents, that were labeled at least with one label contained in the superclass. Documents being labeled with trigger warnings from different superclasses were assigned to each respective fold. To extract meaningful and class separating seed words for each of the six superclasses from the respective bag of documents, words appearing frequently in the pornographic background corpus should be ranked low for the seed word extraction. To achieve this, the relative term frequencies of the word bi-grams of the coarse class bags were divided by each words relative frequency in the background corpus. As a result, words being frequent in a upper class bag of documents and rare in the background corpus are ranked high. Therefore, high ranked words are significant for the respective upper class. To omit zero divisions in this process, terms not occurring in the background corpus were set to a very small frequency (1^{-20}). Finally, from the 50 highest ranked words in each upper class document bag, meaningful five to nine word bi-grams were manually selected from each topic. The resulting seed words for each topic are listed in Table 2.

Subsequently, the extracted seed words were used as input for Seeded LDA, a semi-supervised extension of LDA originally proposed by Lu et al. [41] and further improved by Watanabe and

Baturo [46]. In this approach, pseudo-counts were added to the seed words before fitting the topic model to bias the model towards extracting topics associated with the seed words [41, 46]. The use of semi-supervised topic modeling instead of an unsupervised approach made it possible to establish a direct correspondence between the created superclasses of trigger warnings and the extracted topics. Furthermore, an additional advantage of semi-supervised topic modeling is that it is better able to detect rare topics in the data [47], which is particularly important in this work because, as described in Section 3, topics other than "pornographic content" were only present to a small extent in the dataset.

In order to create a topic model based on bi-grams, the cleaned documents were tokenized into bi-grams as an additional pre-processing step. In the following, the topic model was trained over 500 iterations, where the hyperparameters alpha and beta affecting the concentration of topic probabilities in documents and word probabilities in topics [46] were set to 0,5 and 0,1. The pseudo-counts of the seed words were determined in proportion to the size of the vocabulary, as by Lu et al. [41], and set to approximately 221.000, which is 2 % of the total number of bi-grams in the dataset. Training the topic model was the most time-consuming step for building our classification models and took about 15 hours using 32 Intel(R) Xeon(R) Gold 6132 CPUs with 100 GB RAM. In the end the probabilities of the six topics in the documents were considered as features.

4.4. Coarse Classifier

For the two-stage approach based on the features introduced in the last sections, a coarse classifier was trained to classify the six superclasses and output another set of topic probabilities. For this purpose, an MLP with six neurons in the output layer (one for each topic) was trained. Its architecture contained one hidden layer with 100 neurons and ReLU as activation function. It should be noted that unlike the fine-grained classifier, only a single multi-label MLP was used to reduce the training time [48]. Furthermore, the results were compared with an adapted version of RF for multi-label problems, as described by [49], using 100 trees. As the results of the MLP outperformed the RF, the experiments were continued using the results of the MLP in the fine-grained second stage classifier. The training of the MLP took 20 minutes, while the Random Forest took only two minutes, with both models utilising an AMD Ryzen 9 3900X 12-core processor equipped with 64 GB RAM.

4.5. Fine-Grained Classifier

For the final classification of the labels an ensemble of 32 binary MLPs - one for each class - was used. The decision to use 32 binary MLPs has been made because results of first experiments using a single multi-label MLP were not promising. Each MLP was built with three dense layers of size 128, 100 and 32 where a dropout of 0.1 was applied to the first two layers. After each layer a ReLU-activation function was applied. As the loss function the cross-entropy-loss was used with varying label weights w_0 and w_1 to compensate the strong label imbalance in the dataset. w_0 is the penalty factor for negative classification (label absence) and w_1 is the penalty factor for positive classification (label present). The weights were computed for each of the 32

Table 3

Short description of submitted runs. λ is a parameter to control weighting of labels in the loss function of the MLP.

Run	Short description
MLP_S	One stage approach using fastText-embeddings and topic model probabilities as input features and $\lambda = 1$.
MLP_E1	Two stage approach of a coarse classifier (MLP) for the upper classes followed by an ensemble of 32 MLPs (one for each label). Label weights in the loss function were higher than in MLP_E2 ($\lambda = 1$).
MLP_E2	Two stage approach of a coarse classifier (MLP) for the upper classes followed by an ensemble of 32 MLPs (one for each label). Label weights in the loss function were lower than in MLP_E1 ($\lambda = 2$ except for “pornographic-content” $\lambda = 1$ was kept).

MLPs with Equation (2):

$$(w_0, w_1) = \left(\frac{N}{N-l}, \frac{N}{\lambda l} \right), \quad (2)$$

where N is the total number of training examples, l is the number of occurrences of the respective label in the training data and λ is a parameter to control the weights. With $\lambda \geq 1$ missing a label (false negative) is penalized stronger than a false positive which is necessary in imbalanced datasets to prevent the network only predict the majority class and therefore to improve the recall.

For the submitted runs three different settings were tested (Table 3). In MLP_S the document embeddings concatenated with the upper class topic probabilities of the topic model were used as input for the binary MLPs. Parameter lambda was set to one. In contrast, MLP_E1 and MLP_E2 were two staged approaches: on a first stage a coarse classifier was used to predict topic probabilities on top of the output of the topic model. The input of the final binary MLPs was a concatenation of the document embeddings, the topic model probabilities and the output of the coarse classifier. In MLP_E1 lambda was set to one. In MLP_E2 lambda was set to 2 to soften the penalization except for the label “pornographic-content” for which lambda was kept one, as it is much more frequent compared to the other labels.

The final classifiers were trained on the whole training set and evaluated using the validation set. The training of the fine-grained classifier took 50 minutes using an NVIDIA Tesla T4 with 15 GB RAM.

5. Results and Discussion

5.1. Coarse Classifier

The results of the compared coarse classifiers (Table 4) show clearly that the MLP outperformed RF. In particular the difference in the macro scores is high, whereas the micro scores are in the same range. Comparing precision and recall, the results show that the main problem was to identify all topics while not missing some. This worked better in the MLP which was the

Table 4

Comparison of the results of MLP and RF as Coarse Classifier.

	Macro-Scores			Micro-Scores		
	Pr	Re	F1	Pr	Re	F1
MLP	0.7927	0.3727	0.4502	0.9123	0.7531	0.8251
RF	0.5723	0.2277	0.2628	0.9105	0.6931	0.7871

Table 5

Results of submitted runs on the validation data.

	Macro-Scores			Micro-Scores			Acc
	Pr	Re	F1	Pr	Re	F1	
MLP_S	0.0822	0.4530	0.1147	0.2751	0.7350	0.4003	0.2149
MLP_E1	0.1135	0.6511	0.1622	0.2655	0.8214	0.4013	0.2602
MLP_E2	0.1209	0.3925	0.1568	0.4451	0.6947	0.5426	0.4523

reason, why the MLP was used to generate the additional topic probabilities for the fine-grained classifier.

5.2. Fine-Grained (Final) Classifier

The results on the validation set of the three submitted systems (Table 5) with macro F1-Scores between 0.11 and 0.16 are unexpected low. However the micro F1-Scores with values of 0.40 for MLP_S and MLP_E1 and 0.54 for the MLP_E2 are much higher. The big difference between the micro and macro scores is due to multiple labels were never predicted. In detail, 15, 5 and 13 labels were never predicted in the respective runs MLP_S, MLP_E1 and MLP_E2. The lower number of never predicted labels in MLP_E1 coincides with the highest recall in this run. Nevertheless, the high recall also resulted in a low precision. This problem was addressed in the MLP_E2 run using different weights (Section 4.5) compared to MLP_E1. By thereby lowering the penalty factor for false-negatives, the gap between precision and recall could be reduced, resulting in a higher micro F1-Score. The macro F1-Score is slightly lower as in MLP_E1 as the macro recall decreased by approx. 0.26, but the macro precision did only slightly increase (0.0074).

The comparison between the MLP_S run not using the results of the coarse topic classifier and the MLP_E1 and MLP_E2 using the upstream topic classifier shows, that the results, in particular the macro scores and accuracy could be improved in the latter systems. This is an interesting observation as the coarse MLP-classifier used the same features (document embeddings plus topic model probabilities) that were also put in the final fine-grained classifiers. This means, the additional system outputting another set of topic weights generates additional features with a positive impact on the final results. This also leads to the conclusion, that the introduction of those manually defined coarse topics is helpful for the classification.

The two best performing systems, namely MLP_E1 and MLP_E2, were evaluated on a hidden test set via the TIRA environment [50]. Table 6 summarises the results on this dataset. The

Table 6

Results of submitted runs on the test data.

	Macro-Scores			Micro-Scores			Acc
	Pr	Re	F1	Pr	Re	F1	
MLP_E1	0.113	0.631	0.161	0.266	0.818	0.402	0.268
MLP_E2	0.119	0.381	0.152	0.444	0.69	0.54	0.456

highest macro F1 score, which was chosen by the organisers as the primary ranking criterion [10], was again achieved by MLP_E1 and was approx. 0.16. With this result we have reached the 6th place of the Trigger Detection task of PAN'23.

6. Future Outlook

Even though the numerical results are low, the tested approaches have potential to be further improved. By now, the coarse topics were manually created but it is not clear, if those topics are optimal or if another choice could improve the results. Furthermore, the improvement of the performance due to the variation of weights in MLP_E2 shows that parameter fine-tuning has a big influence in this case. Instead of setting λ in the weight-function (Equation (2)) globally for all 32 fine grained classifiers, it would likely improve the result if λ would be optimized for each classifier separately. In particular for the less represented classes an undersampling of the negative classes could also help to address the imbalance problem.

Another idea is to use a classifier chain for the final classifiers as done in Bellmann et al. [45]. It is not completely clear but there are likely inter-dependencies between the trigger warnings. A classifier chain would make use of those dependencies by arranging in a row (chain) taking outputs of previous binary classifiers as additional inputs for the current classifier. However, this would require an analysis of dependencies between the classes to find the optimal order of the classifiers in the chain.

References

- [1] M. Wolska, C. Schröder, O. Borchardt, B. Stein, M. Potthast, Trigger Warnings: Bootstrapping a Violence Detector for FanFiction, 2022. [arXiv:2209.04409](https://arxiv.org/abs/2209.04409).
- [2] Organization for Transformative Works (OTW), Home | Archive of Our Own, <https://archiveofourown.org/>, 2023.
- [3] J. Rogers, Authentic Representation and Author Identity: Exploring Mental Illness in The Hobbit Fanfiction, *Canadian Journal of Disability Studies* 8 (2019) 127–146. doi:10.15353/cjds.v8i2.494.
- [4] C. V. Bruns, Stinging or Soothing: Trigger Warnings, Fanfiction, and Reading Violent Texts, *Journal of Aesthetic Education* 55 (2021) 15–32.
- [5] F. Göbel, The dark arts: Violence, incest and rape in Harry Potter fan fictions, in: M. Gymnich, H. Birk, D. Burkhard (Eds.), "Harry - yer a wizard": Exploring J.K. Rowling's Harry Potter Universe, number 6 in *Wissenschaftliche Beiträge aus dem Tectum-Verlag*.

- Reihe Anglistik, Tectum Verlag, in der Nomos Verlagsgesellschaft, Baden-Baden, 2017, pp. 215–223.
- [6] E. J. M. Knox, *Trigger Warnings: History, Theory, Context*, Rowman & Littlefield, Lanham, Maryland, USA, 2017.
- [7] K. Woodford, *Trigger warning*, <https://archiveofourown.org/>, 2023.
- [8] 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.
- [9] J. Bevendorff, I. Borrego-Obrador, M. Chinea-Ríos, M. Franco-Salvador, M. Fröbe, A. Heini, K. Kredens, M. Mayerl, P. Pezik, 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. Tsirikika, 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.
- [10] 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.
- [11] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*, in: *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, volume 1, Association for Computational Linguistics, Minneapolis, Minnesota, 2019, pp. 4171–4186. doi:10.18653/v1/N19-1423.
- [12] S. Remus, R. Aly, C. Biemann, *Germeval 2019 task 1: Hierarchical classification of blurbs*, in: *Proceedings of the 15th Conference on Natural Language Processing (KONVENS 2019)*, German Society for Computational Linguistics & Language Technology, Erlangen, Germany, 2019, pp. 280–292.
- [13] N. B. Defersha, J. Abawajy, K. Kekeba, *Deep Learning based Multilabel Hateful Speech Text Comments Recognition and Classification Model for Resource Scarce Ethiopian Language: The case of Afaan Oromo*, in: *International Conference on Current Development in Engineering and Technology (CCET)*, IEEE Xplore, 2022, pp. 1–11. doi:10.1109/CCET56606.2022.10080837.
- [14] J. M. Pérez, F. M. Luque, D. Zayat, M. Kondratzky, A. Moro, P. S. Serrati, J. Zajac, P. Miguel, N. Debandi, A. Gravano, V. Cotik, *Assessing the Impact of Contextual Information in Hate Speech Detection*, *IEEE Access* 11 (2023) 30575–30590. doi:10.1109/ACCESS.2023.3258973.
- [15] E. Utami, R. Rini, A. F. Iskandar, S. Raharjo, *Multi-Label Classification of Indonesian Hate Speech Detection Using One-vs-All Method*, in: *Proceedings of the International Conference on Information Technology, Information Systems and Electrical Engineering (ICITISEE)*, IEEE Xplore, Purwokerto, Indonesia, 2021, pp. 78–82. doi:10.1109/ICITISEE53823.2021.9655883.

- [16] R. A. Ilma, S. Hadi, A. Helen, Twitter's Hate Speech Multi-label Classification Using Bidirectional Long Short-term Memory (BiLSTM) Method, in: *Proceeding of the International Conference on Artificial Intelligence and Big Data Analytics*, IEEE Xplore, Bandung, Indonesia, 2021, pp. 93–99. doi:10.1109/ICAIBDA53487.2021.9689767.
- [17] R. Joshi, R. Karnavat, K. Jirapure, R. Joshi, Evaluation of Deep Learning Models for Hostility Detection in Hindi Text, in: *Proceedings of the 6th International Conference for Convergence in Technology (I2CT)*, IEEE Xplore, Maharashtra, India, 2021, pp. 1–5. doi:10.1109/I2CT51068.2021.9418073.
- [18] M. Niemann, Abusiveness is Non-Binary: Five Shades of Gray in German Online News-Comments, in: *Proceedings of the 21st Conference on Business Informatics (CBI)*, volume 1, IEEE Computer Society, Moscow, Russia, 2019, pp. 11–20. doi:10.1109/CBI.2019.00009.
- [19] J. Read, B. Pfahringer, G. Holmes, E. Frank, Classifier chains for multi-label classification, in: *Machine Learning and Knowledge Discovery in Databases: Proceedings of the European Conference (ECML PKDD)*, volume 2, Springer, Bled, Slovenia, 2009, pp. 254–269.
- [20] J. Read, B. Pfahringer, G. Holmes, E. Frank, Classifier chains for multi-label classification, *Machine learning* 85 (2011) 333–359. doi:10.1007/s10994-011-5256-5.
- [21] M. Kulesa, E. Loza Mencía, Dynamic Classifier Chain with Random Decision Trees, in: L. Soldatova, J. Vanschoren, G. Papadopoulos, M. Ceci (Eds.), *Discovery Science: Proceedings of the 21st International Conference*, Lecture Notes in Computer Science, Springer Nature, Limassol, Cyprus, 2018, pp. 33–50. doi:10.1007/978-3-030-01771-2.
- [22] F. A. Prabowo, M. O. Ibrohim, I. Budi, Hierarchical Multi-label Classification to Identify Hate Speech and Abusive Language on Indonesian Twitter, in: *Proceeding of the 6th International Conference on Information Technology, Computer and Electrical Engineering (ICITACEE)*, IEEE Xplore, Semarang, Indonesia, 2019, pp. 1–5. doi:10.1109/ICITACEE.2019.8904425.
- [23] D. Purwitasari, D. A. Navastara, Y. Findawati, K. A. Pramana, A. Budi Raharjo, Feature Extraction in Hierarchical Multi-Label Classification for Dangerous Speech Identification on Twitter Texts, in: *Proceedings of the International Conference on Computer Science, Information Technology and Engineering (ICCoSITE)*, IEEE Xplore, Jakarta, Indonesia, 2023, pp. 78–83. doi:10.1109/ICCoSITE57641.2023.10127774.
- [24] F. Schmid, J. Thielemann, A. Mantwill, J. Xi, D. Labudde, M. Spranger, FoSIL -Offensive language classification of German tweets combining SVMs and deep learning techniques, in: *Proceedings of the 15th Conference on Natural Language Processing (KONVENS 2019)*, Erlangen, Germany, 2019, pp. 382–386.
- [25] N. Sevani, I. A. Soenandi, Adianto, J. Wijaya, Detection of Hate Speech by Employing Support Vector Machine with Word2Vec Model, in: *2021 7th International Conference on Electrical, Electronics and Information Engineering (ICEEIE)*, 2021, pp. 1–5. doi:10.1109/ICEEIE52663.2021.9616721.
- [26] A. G. D'Sa, I. Illina, D. Fohr, BERT and fastText Embeddings for Automatic Detection of Toxic Speech, in: *Proceedings of the International Multi-Conference on: "Organization of Knowledge and Advanced Technologies" (OCTA)*, IEEE Xplore, Tunis, Tunisia, 2020, pp. 1–5. doi:10.1109/OCTA49274.2020.9151853.
- [27] R. Lumbantoruan, R. U. Siregar, I. Manik, N. Tambunan, H. Simanjuntak, Analysis Com-

- parison of FastText and Word2vec for Detecting Offensive Language, in: Proceedings of the IEEE International Conference of Computer Science and Information Technology (ICOSNIKOM), IEEE Xplore, Laguboti, North Sumatra, Indonesia, 2022, pp. 1–8. doi:10.1109/ICOSNIKOM56551.2022.10034886.
- [28] Y. Shao, S. Taylor, N. Marshall, C. Morioka, Q. Zeng-Treitler, Clinical Text Classification with Word Embedding Features vs. Bag-of-Words Features, in: Proceedings of the IEEE International Conference on Big Data (Big Data), IEEE Xplore, Seattle, WA, USA, 2018, pp. 2874–2878. doi:10.1109/BigData.2018.8622345.
- [29] G. B. Herwanto, A. M. Ningtyas, K. E. Nugraha, N. P. Trisna, Hate Speech and Abusive Language Classification using fastText, in: Proceedings of the International Seminar on Research of Information Technology and Intelligent Systems (ISRITI), IEEE Xplore, Yogyakarta, Indonesia, 2019, pp. 69–72. doi:10.1109/ISRITI48646.2019.9034560.
- [30] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, J. Dean, Distributed Representations of Words and Phrases and their Compositionality, in: Proceedings of the 26th International Conference on Neural Information Processing Systems, volume 2, Curran Associates Inc., Lake Tahoe, Nevada, USA, 2013, pp. 1–9.
- [31] S. Lai, K. Liu, S. He, J. Zhao, How to Generate a Good Word Embedding, IEEE Intelligent Systems 31 (2016) 5–14. doi:10.1109/MIS.2016.45.
- [32] S. J. Fodeh, T. Li, K. S. Menczynski, T. M. Burgette, A. K. Harris, G. I. Ilita, S. K. Rao, J. F. Gemmell, D. S. Raicu, Using Machine Learning Algorithms to Detect Suicide Risk Factors on Twitter, in: Proceeding of International Conference on Data Mining Workshops (ICDMW), IEEE Xplore, Beijing, China, 2019, pp. 941–948. doi:10.1109/ICDMW.2019.00137.
- [33] K. Shidara, H. Tanaka, R. Asada, K. Higashiyama, H. Adachi, D. Kanayama, Y. Sakagami, T. Kudo, S. Nakamura, Linguistic Features of Clients and Counselors for Early Detection of Mental Health Issues in Online Text-based Counseling, in: Proceedings of the 44th Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC), IEEE Xplore, Scottish Event Campus, Glasgow, UK, 2022, pp. 2668–2671. doi:10.1109/EMBC48229.2022.9871408.
- [34] K. Allen, A. L. Davis, T. Krishnamurti, Indirect Identification of Perinatal Psychosocial Risks from Natural Language, IEEE Transactions on Affective Computing (2021) 1–1. doi:10.1109/TAFFC.2021.3079282.
- [35] Y. Cheng, Y. Shao, S. Gottipati, Q. Zeng, Comparison of Structured and Free-text Based Features for Rehospitalization Prediction among Patients with Severe Mental Illness, in: Proceedings of the International Conference on Electrical, Computer and Energy Technologies (ICECET), IEEE Xplore, Cape Town, South Africa, 2021, pp. 1–6. doi:10.1109/ICECET52533.2021.9698504.
- [36] D. M. Blei, A. Y. Ng, M. I. Jordan, Latent Dirichlet Allocation, Journal of Machine Learning Research (2003) 993–1022.
- [37] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, R. Harshman, Indexing by latent semantic analysis, Journal of the American Society for Information Science 41 (1990) 391–407. doi:10.1002/(SICI)1097-4571(199009)41:6<391::AID-ASI1>3.0.CO;2-9.
- [38] T. Nguyen, D. Phung, B. Dao, S. Venkatesh, M. Berk, Affective and Content Analysis of Online Depression Communities, IEEE Transactions on Affective Computing 5 (2014)

- 217–226. doi:10.1109/TAFPC.2014.2315623.
- [39] B. Desmet, V. Hoste, Online suicide prevention through optimised text classification, *Information Sciences* 439–440 (2018) 61–78.
- [40] R. Churchill, L. Singh, R. Ryan, P. Davis-Kean, A Guided Topic-Noise Model for Short Texts, in: *Proceedings of the ACM Web Conference 2022, WWW '22*, Association for Computing Machinery (ACM), New York, NY, USA, 2022, pp. 2870–2878. doi:10.1145/3485447.3512007.
- [41] B. Lu, M. Ott, C. Cardie, B. K. Tsou, Multi-aspect Sentiment Analysis with Topic Models, in: *Proceedings of the 11th International Conference on Data Mining Workshops, ICDMW '11*, IEEE Computer Society, USA, 2011, pp. 81–88. doi:10.1109/ICDMW.2011.125.
- [42] A. Joulin, E. Grave, P. Bojanowski, T. Mikolov, Bag of Tricks for Efficient Text Classification, 2016. doi:10.48550/arXiv.1607.01759. arXiv:1607.01759.
- [43] L. Hickman, S. Thapa, L. Tay, M. Cao, P. Srinivasan, Text Preprocessing for Text Mining in Organizational Research: Review and Recommendations In-press at *Organizational Research Methods*, *Organizational Research Methods* in press (2020). doi:10.1177/1094428120971683.
- [44] P. Bojanowski, E. Grave, A. Joulin, T. Mikolov, Enriching word vectors with subword information (2017). doi:https://doi.org/10.48550/arXiv.1607.04606. arXiv:1607.04606 [cs].
- [45] F. Bellmann, L. Bunzel, C. Demus, L. Fellendorf, O. Gräupner, Q. Hu, T. Lange, A. Stuhr, J. Xi, D. Labudde, M. Spranger, Multi-label classification of blurbs with svm classifier chains, in: *Proceedings of the 15th Conference on Natural Language Processing (KONVENS 2019)*, German Society for Computational Linguistics & Language Technology, Erlangen, Germany, 2019, pp. 293–299. URL: https://konvens.org/proceedings/2019/https://konvens.org/proceedings/2019/papers/germeval/Germeval_Task1_paper_1.pdf.
- [46] K. Watanabe, A. Baturu, Seeded Sequential LDA: A Semi-supervised Algorithm for Topic-specific Analysis of Sentences, *Social Science Computer Review* (2023) 08944393231178605.
- [47] J. Jagarlamudi, H. Daumé III, R. Udupa, Incorporating Lexical Priors into Topic Models, in: *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, Association for Computational Linguistics, Avignon, France, 2012, pp. 204–213.
- [48] J. Bogatinovski, L. Todorovski, S. Džeroski, D. Kocev, Comprehensive comparative study of multi-label classification methods, *Expert Systems with Applications* 203 (2022). doi:10.1016/j.eswa.2022.117215.
- [49] M. Dumont, R. Marée, L. Wehenkel, P. Geurts, Fast Multi-class Image Annotation with Random Subwindows and Multiple Output Randomized Trees., in: *Proceedings of the Fourth International Conference on Computer Vision Theory and Applications*, volume 2, Lisboa, Portugal, 2009, p. 203.
- [50] 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.