Towards Automated Moderation: Enabling Toxic Language Detection with Transfer Learning and Attention-Based Models

Matthew Caron
Paderborn University
matthew.caron@upb.de

Frederik S. Bäumer
Bielefeld University of Applied Sciences
frederik.baeumer@fh-bielefeld.de

Oliver Müller
Paderborn University
oliver.mueller@upb.de

Abstract

Our world is more connected than ever before. Sadly, however, this highly connected world has made it easier to bully, insult, and propagate hate speech on the cyberspace. Even though researchers and companies alike have started investigating this real-world problem, the question remains as to why users are increasingly being exposed to hate and discrimination online. In fact, the noticeable and persistent increase in harmful language on social media platforms indicates that the situation is, actually, only getting worse. Hence, in this work, we show that contemporary ML methods can help tackle this challenge in an accurate and cost-effective manner. Our experiments demonstrate that a universal approach combining transfer learning methods and state-of-the-art Transformer architectures can trigger the efficient development of toxic language detection models. Consequently, with this universal approach, we provide platform providers with a simplistic approach capable of enabling the automated moderation of user-generated content, and as a result, hope to contribute to making the web a safer place.

1. Introduction

With the rise of the smartphone, the widespread access to the internet, and the surge of social media networks, our world is more connected than ever before. These technological advancements have not only revolutionized the way we communicate with one another, but they have also forever changed the way we share ideas, consume our daily news, or even do business. However, notwithstanding the uncountable benefits that our modern society has gained from these developments, our highly connected world has made it easier for social media users to bully, insult, and propagate hate speech on the cyberspace [1]. Genuinely, alongside the dramatic increase in user-generated content and interactions on social and sharing platforms, the web has also witnessed a rise in abusive, discriminative, hateful, offensive, and racist material [2], which, for the lack of a universal definition, is henceforth referred to as toxic language. In truth, the non-restrictive and pseudo-anonymous nature of the internet provides users with malicious intents with the perfect environment to express hatred and despise.

The fact that some users "misuse the [web] to promote offensive and hateful language, which mars experience of regular users, affects the business of online companies, and may even have severe real-life consequences" [3] has forced providers to look for effective ways to eradicate toxic language from sharing platforms [4]. Accordingly, community-driven moderation approaches, such as encouraging users to flag inappropriate content, can partly solve the problem but can also lead to silencing supposedly unpopular opinions. On the other hand, the ever-increasing amount of available user-generated content makes the labor-intensive process of manually moderating and policing these platforms solely inconceivable [4, 5]. As a result, the need for automated solutions capable of detecting toxic language and, therefore, support the moderation process is proliferating [5].

In the last decade, researchers and companies alike have started investigating this real-world problem. Even though detecting the thin line between acceptable opinions and hate speech is, in some cases, still extremely challenging [4], advancements in machine learning (ML), natural language processing (NLP), and text classification techniques have made the detection of offensive language more accurate and more accessible [6]. Nevertheless, the question remains as to why users are increasingly being exposed to hate and discrimination online. As reported by various news outlets and academic publications, the noticeable and persistent increase in harmful language on social media and sharing platforms indicates that the situation is only getting worse [7, 8, 9, 10]; thus, confirming that this problem is far from being resolved. In truth, even the ongoing COVID-19 global pandemic triggered a dramatic rise in cultural, political, and religious hatred

URI: https://hdl.handle.net/10125/79428
978-0-9981331-5-7

on the internet [11, 12, 13]. Hence, by simply browsing the web, one can witness the adverse effects of harmful language, and that, without even actually looking for it. Furthermore, while it seems that some platforms allow basically every form of speech, other providers, such as the German National News Service *Tagesschau*, have decided, due to a lack of resources[1], to deactivate, in part, the comment function. In either case, we firmly believe that users, providers, and freedom of speech suffer from this situation.

Given the apparent advancements in ML techniques and the present accessibility of such methods, it is, from an outside perspective, difficult to understand the exact reason(s) why some platform providers are not, manually or automatically, moderating toxic content effectively. This endeavor may seem technically daunting for some or even costly for others. The so-called cold-start problem, the lack of readily available datasets, and the need for data scientists or expensive hardware are all factors that, perhaps, weigh in the balance. Even though such tools may not generate revenue per se and are, therefore, not necessarily financially attractive, they can still help providers avoid getting hit with expensive fines in several regions of the world – e.g., the European Union [14]. Hence, in this work, we show that contemporary ML methods can help tackle this challenge in an accurate and cost-effective manner. Our experiments demonstrate that a universal approach combining transfer learning methods and state-of-the-art Transformer architectures can trigger the efficient development of toxic language detection models. Consequently, with this universal approach, we provide platform providers with a simplistic and functional approach capable of enabling the automated moderation of user-generated content, and as a result, hope to contribute to making the web a safer place by exposing:

1. that a simple universal approach can achieve state-of-the-art performance on a variety of toxic language identification tasks;

2. that transfer learning is not only highly efficient but that it also provides better classification performance; and

3. that state-of-the-art methods can be implemented efficiently; therefore, enabling a more accessible and broader adoption.

The paper is structured as follows: In Section 2, we present a comprehensive overview of the most prominent research efforts in the field of hate speech and offensive language detection. In Section 3, we describe our approach for the detection of toxic language before we evaluate its performance in Section 4. Finally, we give an outlook on future work and discuss the limitations of the current study (Section 5).

2. Related Work

In this section, we outline the theoretical foundations underpinning transfer learning in NLP and present a comprehensive overview of the most prominent research efforts in the field of toxic language detection.

2.1. Transfer Learning in NLP

In a traditional supervised ML setting, a statistical model is trained on previously annotated training data and then used to make predictions on future or unseen data. In this setting, we assume that both the training data and the future or unseen data are originating from the same distribution, otherwise, a new model needs to be trained for each task. However, this traditional ML approach is quite different from how humans learn. Instead of always starting from scratch, humans have the ability to reuse the knowledge and skills learned from previous tasks. This incredible ability is the fundamental motivation behind transfer learning, which can be defined as the application of knowledge learned from a previous task (source task) to a novel task (target task) (for a detailed survey on transfer learning, s. [15]).

Although many types of transfer learning exist, sequential transfer learning is the most frequently used transfer learning approach in NLP and consists of two stages: (1) a pre-training phase – i.e., training the model on a source task – and (2) an adaptation phase (see [16] for a more detailed description). In NLP, pre-training is typically performed on a large and broad annotated corpus. This pre-training phase aims to learn universal syntactic and semantic patterns about language, which one can reuse with a wide range of target tasks. In the subsequent adaptation phase, the model is trained on the actual target task. This adaptation or fine-tuning is typically performed on a smaller yet focused annotated corpus. The adaptation aims to learn patterns that are specific to the domain and the task of interest. While the pre-training phase is computationally expensive – i.e., it can take days or weeks on high-performance computing (HPC) hardware – it only needs to be performed once. In contrast, the adaptation phase, which only needs to be performed for each new task, can be very efficient.

For the pre-training task, most transfer learning approaches use some form of language modeling (LM) [17]. Broadly speaking, the goal of LM is to predict the next word(s) in a sequence of text given the preceding

[1] https://meta.tagesschau.de/hilfe/FAQ (accessed August 24, 2021)

words. As accurately predicting a word in a sequence requires knowledge of both the syntactic and semantic roles of the words in context, LM is ideally suited to learn generic reusable linguistic patterns. Besides, training a LM does not require a labeled corpus; hence, it can be done efficiently on very large corpora. In the adaptation or fine-tuning phase, one can either feed the outputs of a pre-trained LM as static features into an independent downstream model or replace the last layer(s) of a pre-trained LM with a target task-specific layer – e.g., a linear classification layer [18]. In the former case, only the downstream model is trained on the task-specific annotated corpus; in the latter case, the whole network can be fine-tuned on the task-specific labeled corpus.

The application of transfer learning has resulted in dramatic improvements in ML-based NLP, both in terms of effectiveness and efficiency. Universal Language Model Fine Tuning (ULMFiT), which is based on a Long Short-Term Memory (LSTM) recurrent neural network architecture, was one of the first transfer learning techniques proposed for NLP [19]. Experiments showed that with only 100 labeled samples, a pre-trained ULMFiT model could match the performance of other modern supervised classification models trained on a hundred times more data. Today, most of the current state-of-the-art transfer learning approaches employ a Transformer-based architecture instead of a recurrent architecture. Transformers are sequence-to-sequence models originally developed for machine translation tasks which easily be adapted for language modeling. Bidirectional Encoder Representations from Transformers (BERT), developed by Google AI, is arguably the most famous Transformer model for LM-based transfer learning [20]. BERT has been incorporated into Google Search for improving query understanding, and Google describes it as "one of the biggest leaps forward in the history of [Google] Search" [21].

2.2. Toxic Language Detection

In this section, we present a comprehensive overview of the most prominent research efforts in the field of toxic language detection. As addressed in the introduction, toxic language can take many forms – e.g., abuse, racism, or sexism. Consequently, even though the papers exhibited in Table 1 all focus on the same broad topic, the exact purpose of every research varies slightly while the various methods, architectures, datasets, and supported languages differ extensively. A comparison between the individual papers is thus made more difficult. However, one problem has already become clear: researchers often start from scratch, being either unable or unwilling to leverage existing implementations or models.

Back in 2009, one of the first research papers published on the subject proposed a supervised learning approach aimed at detecting harassment in user-generated content [22]. In this early attempt, the authors focused on a binary classification task by experimenting with three distinct datasets – i.e., Kongregate, Slashdot, and MySpace – and various linear Support Vector Machine (SVM) models. Alternative SVM approaches for the binary classification of toxic language are proposed by [23], [24] and [31]. The work by [31] should also be highlighted, as it is dedicated to a low-resource language, namely Arabic, and provides both a method and a dataset. Recently, various works focusing on various European languages, such as German [34], Spanish [35], and Italian [33], have been published.

Stepping away from more traditional methods, [25] experimented with character and word n-grams by training logistic regression models on various combinations of character-level features and unusual features, such as the gender of the author and the originating location of the Tweet. According to the authors, the morphological features provided by character-level approaches outperform traditional lexical features when it comes to classifying user-generated content [1]. Following [24], they argued that due to the ever-changing standards and guidelines on social media, languages could evolve in a conscious or in an unconscious manner. Hence, by adjusting their writing styles, users can circumvent the lexical-based filtering systems used by social networking platforms [1]. A character-level approach would, most likely, ignore these modifications.

In one of the most extensive studies, [26] focused on more contemporary techniques by experimenting with deep learning approaches. By making use of modern neural network architectures, they were the first to truly experiment with such methods in the field of hate speech detection and demonstrated that these approaches can outperform approaches based on char and n-gram representations [26]. Following this work, [27] experimented with four different deep learning architectures to classify hateful comments on Twitter. In this research, their goal was to prove that Convolutional Neural Networks can provide state-of-the-art performance for hate speech categorization.

However, the lack of adaptability and reusability of the proposed approaches and datasets remain. The annotation of domain-specific datasets is very

Table 1. Related work on toxic language identification

	Year	Task	Classifier	Language	Dataset	F1-score
Yin et al. [22]	2009	Binary	SVM	English	Own	0.481
Warner & Hirschberg [23]	2012	Binary	SVM	English	Own	0.630
Nobata et al. [24]	2016	Binary	SVM	English	Djuric et al.	0.774
Mehdad & Tetreault [1]	2016	Binary	NBSVM	English	Djuric et al.	0.770
Waseem & Hovy [25]	2016	Multi.	Log. Reg.	English	Own	0.739
Badjatiya et al. [26]	2017	Multi.	GBDT	English	Waseem	0.930
Gambäck & Sikdar [27]	2017	Multi.	CNN	English	Waseem	0.782
Davidson et al. [28]	2017	Multi.	Log. Reg.	English	Own	0.900
Risch et al. [29]	2019	Binary	BERT (base-cased)	German	GermEval	0.764
		Multi.				0.510
Mozafari et al. [30]	2019	Binary	BERT (base) + CNN	English	Waseem	0.880
		Binary			Davidson et al.	0.920
Mulki et al. [31]	2019	Binary	NB	Arabic	Own	0.896
		Binary	SVM			0.820
Paraschiv & Cercel [32]	2019	Binary	BERT (base-cased)	German	GermEval	0.769
		Multi.	BERT (base-cased)			0.535
Corazza et al. [33]	2020	Binary	LSTM	English	Waseem	0.823
		Binary	LSTM	Italian	Sanguinetti et al.	0.805
		Binary	GRU	German	GermEval	0.758

labor-intensive and thus prevents the previous methods from being used by others [28]. In order to address the lack of annotated, unbiased data (especially for languages other than English) and in order to increase the reusability of models in other domains and languages, more recent work [29, 30] focus on pre-trained language models such as BERT [20]. A significant advantage of Transformer-based approaches is that they can satisfactorily apply knowledge, once learned, to new tasks.

3. Universal Approach

In this section, we present the four fundamental building blocks of the proposed universal approach (see Figure 1). This sequential pipeline, which leverages the benefits of transfer learning and contemporary Transformer architectures, allows the efficient and straightforward development of classification models capable of achieving state-of-the-art performance on various toxic language identification tasks.

3.1. Dataset Selection or Generation

First and foremost, selecting or generating a representative dataset is crucial to any (supervised) machine learning task. Rationally, the quality of the data fed into a machine learning algorithm directly correlates with the performance of the resulting model. Besides some arguably subjective factors, such as the pertinence of the selected sources and the quality of the annotation, the size of the training sample is also known to have an impact on the overall quality of statistical models [36, 37, 38, 39]. This notion becomes even more significant when one is dealing with textual data. In fact, because of the sheer complexity of human languages as well as

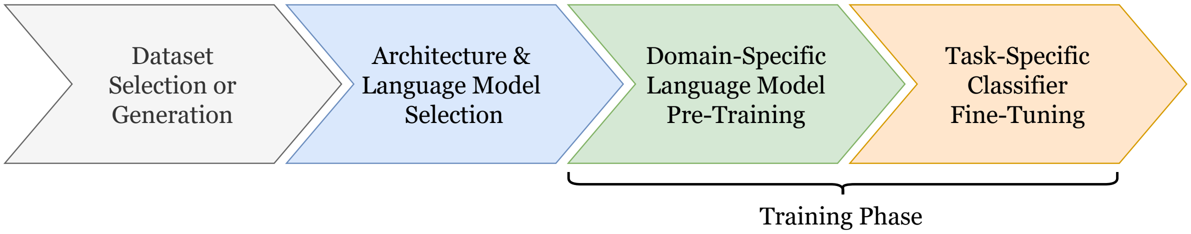
Dataset Selection or Generation
Architecture & Language Model Selection
Domain-Specific Language Model Pre-Training
Task-Specific Classifier Fine-Tuning
Training Phase

Figure 1. Transformer-based universal approach

the high-dimensional nature of the problems at hand, NLP tasks, such as text classification, most commonly benefit from large amounts of data [38, 40]. This being said, generating a large dataset is a painstaking, costly endeavor that is not always feasible or even affordable.

Consequently, by relying on recent advancements in transfer learning and language modeling techniques, Transformer-based models can achieve state-of-the-art results with just a fraction of the (annotated) data previously required. Hence, leveraging a meticulously selected corpus makes it possible to deploy cutting-edge text classifiers while avoiding the usual overhead associated with the data collection and annotation process. Therefore, the idea is to acquire qualitative, representative, and diversified data from various platforms and channels.

3.2. Architecture & Model Selection

Second, selecting the right (pre-trained) language model is, in combination with a representative dataset, of paramount importance in the above approach. In truth, we believe that these two primary steps are critical to the successful and efficient development of the intended classifier. With the field of language modeling growing at an incredible pace, it may seem hard to keep up with all the latest breakthroughs. Yet, as previously exposed, Transformer architectures have become, over the last couple years, the go-to models for many NLP applications, such as neural translation, question answering, and text classification.

Designing with efficiency in mind, the idea is to avoid starting from scratch by leveraging the knowledge embedded into pre-trained Transformer models. These models, which are readily available in a multitude of languages, act as a foundation for the steps that follow. Thereby, the success of this approach partially relies upon this selection.

3.3. Domain-Specific Model Pre-Training

As can be expected, most readily available pre-trained Transformer models are trained on gigabytes of generic data – e.g., Wikipedia or Common Crawl. However, the type of language represented in these corpora may not accurately portray the domain-specific language that one is trying to model. As one can imagine, toxic language is usually conveyed through comments, tweets, or even blogs. This online jargon is most likely to be different from the more formal language found on websites such as Wikipedia. Slang words, abbreviations, as well as grammatical and typographical errors are all characteristics related to the type of language found on social networking platforms. Furthermore, some harmless words and expressions, such as the word "tool" in the demeaning expression "You are a tool!", may even be used in a harmful manner. As a result, with the help of domain-specific unlabelled data, one first wants to tweak, in an unsupervised manner, the already pre-trained language model so that it matches our target language; therefore, increasing the likelihood of success.

3.4. Task-Specific Classifier Fine-Tuning

The second and last stage of the training phase consists of building a classifier on top of the selected Transformer architecture. Keep in mind that the idea behind language modeling is not to predict a target variable, but rather to "assign probabilities to sequences of words" [41]. This means that a vanilla Transformer model is not capable of performing any given classification task. However, by replacing a model's last layer with a discriminative classifier head – e.g., logistic regression – one can leverage the previously acquired language knowledge while giving the model the capability to assign an arbitrary number of classes to given samples. With the help of annotated data, the model, including its classifier layer, can be fine-tuned to suit the task at hand.

4. Experiments

As argued in the introduction, leveraging the benefits of transfer learning and contemporary Transformer architectures can yield state-of-the-art results on the task of toxic language detection and, ultimately, allow the efficient and straightforward development of moderation tools. Hence, in this section, we train and evaluate various classification models capable of detecting toxic language following the universal approach exposed in the previous section. To make the task a bit more challenging and, therefore, to prove the predictive capabilities of this simplistic pipeline, we decided to focus on a language that does not benefit from as many resources as English, namely German. As a result, by focusing on a language such as German, we believe that our experiments is representative of the target group that could benefit from employing the disclosed methodology. Lastly, to prove the generalization capabilities of this universal approach, we evaluate its performance and portability on another dataset provided by a third party – i.e., unseen data gathered and annotated by an independent organization.

4.1. Experimental Setup

For the sake of consistency and reproducibility, the following experiments were all carried out on a single workstation[2] using the same PyTorch[3] development environment – i.e., an open-source machine learning library. As to the transfer learning part of the implementation, we made use of the readily available pre-trained models available in the Transformers[4] library provided by Hugging Face[5]. In addition, to demonstrate that one can tackle this challenging problem by leveraging the efficiency of transfer learning methods alone, we deliberately did not take advantage of powerful virtual machines hosted on cloud platforms or even TPU-accelerated hardware.

4.2. Resources

As briefly discussed, the quality of the data involved in the training phase plays a crucial role in a machine learning model's performance. However, for many languages, the resources are pretty scarce when it comes to toxic language identification. Hence, for most low-resource languages, the first step of any approach will, as expose earlier, consist of manually extracting and annotating data.

Regarding the German language, the lack of appropriate resources is indisputable, and that, despite the effort by a handful of research groups. Nonetheless, in an attempt to promote research in this highly relevant field, the *Shared Task on the Identification of Offensive Language* was put in place by GermEval [42, 43]. This workshop required participants to train machine learning models capable of identifying toxic language with the help of the provided annotated datasets [42, 43]. Even though these publicly available corpora show rather exciting characteristics, such as a multi-class approach and highly comprehensive definitions, they were sourced solely from Twitter and are relatively small in size. Studies have repeatedly shown that different demographics use different social networks [44]. As a result, the linguistic features, the expressions, and the discussed topics vary widely from platform to platform. To overcome this drawback and generate a representative corpus, we built our dataset using data originating from three different social networks, namely Instagram, Twitter, and YouTube, while utilizing the multi-class approach and definitions provided by [45]. Furthermore, we also made sure to collect data from diverse categories – e.g., lifestyle, sports, and politics.

In preparation for the annotation process, we acquired over 200,000 comments from the platforms mentioned above. Following the annotation guidelines proposed by [45], we annotated, for the sake of our experiments, 50,000 randomly selected comments. Logically, we discarded all comments containing only emojis or those written in a foreign language. Table 2 reveals the details about our final dataset containing a total of 20,000 manually annotated samples. The remaining 150,000 comments were set aside for the pre-training phase of the exposed approach.

Table 2. Dataset overview

Label	Instagram	Twitter	YouTube	Total
Other	5,042	5,734	4,224	**15,000**
Offense	283	347	687	**1,317**
Insult	747	634	1,284	**2,665**
Abuse	225	228	565	**1,018**
Total	*6,297*	*6,943*	*6,760*	***20,000***

Finally, in order to assess the quality of our annotations and, therefore, the value and consistency of our dataset, we proceeded with an inter-annotator reliability evaluation. The resulting Krippendorff's values of $\alpha = 0.91$ for the binary setting and $\alpha = 0.79$ for the multi-class setting do not only speak for the quality of the annotation, but also the quality of the definitions and guidelines provided by [45]. In order to promote further research, the above dataset can be made available by the authors upon request.

[2] CPU: Intel i7-9800X; RAM: 8x16GB (128 GB); GPU: 1x NVIDIA Titan RTX (24 GB)

[3] https://pytorch.org

[4] https://github.com/huggingface/transformers

[5] https://huggingface.co

4.3. Text Preprocessing

When it comes to preprocessing text documents, approaches, methods, and views vary widely. As a matter of fact, some of the studies presented in the related work section of this paper employed heavy preprocessing – e.g., [22] – while others did not even bother to address the subject – e.g., [27]. Hence, since our approach revolves around efficiency, we decided that it would be best to follow a more minimalist approach. Thus, our preprocessing consisted of converting the documents to lowercase before removing all unwanted characters, symbols, emojis, hashtags, and mentions. Even though it can be rightfully argued that various entities such as emojis or hashtags contain valuable information beneficial to the overall performance of a classification model, the goal of our experiments is to demonstrate the straightforward predictive capabilities of the universal approach by solely focusing on language.

4.4. Language Modeling and Classifier Fine-Tuning

As exposed, the field of language modeling is currently growing at an incredible pace, and Transformer models are at the center of this evolution. For this reason, we decided to experiment with the most common Transformer-based architecture, namely, BERT [20]. This architecture is considered by many to be one of the most exciting developments in NLP in recent years and acts as the groundwork for many contemporary works.

With the success of the approach relying on the selection of a suitable pre-trained model, we opted for the uncased version of the so-called German BERT model[6]. This model, which is based on the work by [20] comprises 110 million parameters and was trained on some 16 GB of data originating from Wikipedia, the EU Bookshop corpus, Open Subtitles, CommonCrawl, ParaCrawl, and News Crawl. Hence, with this model, we can jumpstart our implementation by leveraging the knowledge gathered from more than 2.3 billion tokens and a vocabulary containing over 30,000 tokens, which was generated using the language-independent subword tokenizer known as SentencePiece[7] [46]. This subword approach, in combination with the self-attention mechanism characteristical of the Transformer architecture [47], allows the model to deal with linguistic ambiguity, linguistic subtleties, word variations, and even misspellings.

Even though German BERT provides a solid foundation for our implementation, we can further pre-trained the model's weights using domain-specific data. This first unsupervised step of the training phase may help our model, at a later stage, to better recognize words, patterns, and even expressions distinct to our target language. Hence, before moving on to the classifier and, therefore, the actual task, we further pre-trained the German BERT model in its entirety – i.e., without freezing any layers – with the 150,000 unlabeled comments mentioned earlier.

At last, we can now build a task-specific classifier on top of our architecture. This second and final stage of the training phase consists, as explained earlier, of replacing the last layer of our pre-trained language model with a classifier head before fitting the model's parameters, or weights, to our downstream task. To proceed, we trained two versions of the German BERT model for both our binary and multi-class experiments – i.e., one without pre-training and one with domain-specific pre-training – using 5,000, 10,000, and 15,000 annotated training samples. Experimenting with various training sets enabled us to assess the effects of the sample size on the model's overall performance. Please note that we did not freeze any layers for our experiments – i.e., we fine-tuned both the weights of the pre-trained language model and those of the classifier.

4.5. Evaluation

Table 3 and Table 4 expose the classification results obtained by our binary and multi-class models when tested on various sample sizes. In order to put these results into perspective, we also trained and evaluated several baseline models that were optimized using a Bayesian hyperparameter tuning approach and the same training and test data. The results show that the models trained using the Transformer-based universal approach outperform all other models in every single category. Even though marginal, the models pre-trained on domain-specific data also manage to beat their fine-tuned German BERT counterparts. Furthermore, the results reveal that the models based on the universal approach do not benefit from larger training sets. This suggests that one can achieve state-of-the-art results with a minimal amount of training data and, consequently, minimize the effects of the cold-start problem.

4.6. Benchmark Evaluation

In order to be able to judge the performance of our proposed approach, especially with regards to generalization, we assess how our models stack

[6] https://huggingface.co/dbmdz/bert-base-german-uncased

[7] https://github.com/google/sentencepiece

Table 3. Binary evaluation for various training sample sizes

	Model	Accuracy	Precision	Recall	F_1-score
5,000 samples	TF-IDF + Logistic Regression	0.851	0.893	0.710	0.749
	TF-IDF + XGBoost	0.878	0.899	0.771	0.810
	German BERT (fine-tuned)	0.949	0.932	0.933	0.932
	Universal Approach	**0.955**	**0.940**	**0.939**	**0.939**
10,000 samples	TF-IDF + Logistic Regression	0.877	0.905	0.764	0.804
	TF-IDF + XGBoost	0.892	0.911	0.798	0.835
	German BERT (fine-tuned)	0.959	0.948	0.942	0.945
	Universal Approach	**0.962**	**0.949**	**0.951**	**0.950**
15,000 samples	TF-IDF + Logistic Regression	0.891	0.913	0.794	0.833
	TF-IDF + XGBoost	0.904	0.921	0.821	0.857
	German BERT (fine-tuned)	0.962	0.951	0.949	0.950
	Universal Approach	**0.964**	**0.952**	**0.952**	**0.952**

Table 4. Multi-class evaluation for various training sample sizes

	Model	Accuracy	Precision	Recall	F_1-score
5,000 samples	TF-IDF + Logistic Regression	0.828	0.838	0.468	0.537
	TF-IDF + XGBoost	0.864	0.803	0.618	0.676
	German BERT (fine-tuned)	0.923	0.836	0.839	0.836
	Universal Approach	**0.933**	**0.862**	**0.845**	**0.851**
10,000 samples	TF-IDF + Logistic Regression	0.848	0.825	0.540	0.614
	TF-IDF + XGBoost	0.872	0.828	0.638	0.696
	German BERT (fine-tuned)	0.935	0.851	0.860	0.855
	Universal Approach	**0.942**	**0.869**	**0.866**	**0.867**
15,000 samples	TF-IDF + Logistic Regression	0.864	0.841	0.597	0.672
	TF-IDF + XGBoost	0.878	0.833	0.662	0.717
	German BERT (fine-tuned)	0.945	0.877	0.885	0.880
	Universal Approach	**0.947**	**0.887**	**0.886**	**0.886**

Table 5. Benchmark evaluation (GermEval 2019 [43])

	Model	Accuracy	Precision	Recall	F_1-score
Binary	Paraschiv et al. [32]	0.794	0.764	0.776	0.770
	Universal Approach	**0.819**	**0.794**	**0.784**	**0.788**
Multi-class	Paraschiv et al. [32]	0.736	0.585	0.494	0.536
	Universal Approach	**0.740**	**0.575**	**0.544**	**0.558**

up against the best-performing models submitted to GermEval 2019 [43]. As can be seen in Table 5, both our binary as well as our multi-class models outperform the best-performing model submitted by [32]. Even though the models proposed by [32] were also based on a Transformer architecture, we believe that our efficient pre-training provided our implementations with a serious performance advantage.

5. Conclusion

In the last few years, the need for moderation to maintain an orderly interaction on social media and sharing platforms has grown massively. Even though technical solutions are continually evolving [4], the persistent increase in harmful language on the web indicates that the situation is only becoming worse [7, 8, 9, 10]. Consequently, this paper contributes to making the implementation of automated solutions more accessible by exposing how a universal approach can enable the efficient and cost-effective development of accurate machine learning models capable of detecting toxic language. As our empirical experiments showed, it is possible to build state-of-the-art moderation solutions with minimal resources, thanks to modern techniques such as transfer learning and Transformer models.

As with most studies, however, these results must be interpreted in light of some limitations. First, our experiments focused solely on the detection of toxic language and, therefore, did not reflect every type of threat encountered on the web. In fact, threatful actions such as trolling – i.e., the act of instigating conflict by intentionally posting provocative or offensive messages – or doxing – i.e., the act of publicly revealing someone else's private information – are all detrimental behaviors that would need to be addressed in future work. Second, because of a language's complex and diverse nature, ill-intentioned individuals may be able to circumvent automatic moderation systems by adapting their writing style – e.g., by voluntarily introducing typos or non-alphabetical characters. Even though the subword approach, in combination with the self-attention mechanism of the Transformer architecture, makes our models robust to such linguistic alterations, future work should focus on testing the reliability of various architectures by putting those under the stress of adversarial attacks. Lastly, it would be interesting to examine the adaptability and transferability of the presented models to different domains, such as gaming. Even though these models were trained on diverse datasets, the language found on social media platforms may differ from the jargon encountered on gaming platforms.

References

[1] Y. Mehdad and J. Tetreault, "Do Characters Abuse More Than Words?," in *Proceedings of the 17th Annual Meeting of the SIGDD*, pp. 299–303, 2016.

[2] R. Kowalski, G. W Giumetti, A. Schroeder, and M. R Lattanner, "Bullying in the Digital Age: A Critical Review and Meta-Analysis of Cyberbullying Research Among Youth," *Psychological Bulletin*, vol. 140, no. 4, pp. 1073–1137, 2014.

[3] N. Djuric, J. Zhou, R. Morris, M. Grbovic, V. Radosavljevic, and N. Bhamidipati, "Hate Speech Detection with Comment Embeddings," in *Proceedings of the 24th International Conference on World Wide Web*, pp. 29–30, 2015.

[4] A. R. McGillicuddy, J. Bernard, and J. A. Cranefield, "Controlling Bad Behavior in Online Communities: An Examination of Moderation Work," in *Proceedings of the 37th International Conference on Information Systems*, 2016.

[5] U. Bretschneider, T. Wöhner, and R. Peters, "Detecting Online Harassment in Social Networks," in *Proceedings of the 35th International Conference on Information Systems*, 2014.

[6] F. E. Ayo, O. Folorunso, F. T. Ibharalu, and I. A. Osinuga, "Machine Learning Techniques for Hate Speech Classification of Twitter Data: State-of-the-Art, Future Challenges and Research Directions," *Computer Science Review*, vol. 38, p. 100311, 2020.

[7] M. Wolfe-Robinson, "Still a Safe Space for Racists: New Report Criticizes Social Media Giants for allowing Hate Speech," Aug. 2021.

[8] C. Bernatzky, M. Costello, and J. Hawdon, "Who Produces Online Hate?: An Examination of the Effects of Self-Control, Social Structure, & Social Learning," *American Journal of Criminal Justice*, pp. 1–20, 2021.

[9] C. Arcila-Calderón, D. Blanco-Herrero, M. Frías-Vázquez, and F. Seoane, "Refugees Welcome? Online Hate Speech and Sentiments in Twitter in Spain during the Reception of the Boat Aquarius," *Sustainability*, vol. 13, no. 5, p. 2728, 2021.

[10] H. Mansourifar, D. Alsagheer, R. Fathi, W. Shi, L. Ni, and Y. Huang, "Hate Speech Detection in Clubhouse," *arXiv preprint arXiv:2106.13238*, 2021.

[11] J. Orlando, "Young People are Exposed to more Hate Online during COVID. And it Risks their Health."

[12] T. Hyman, "The Harms of Racist Online Hate Speech in the Post-COVID Working World: Expanding Employee Protections," *Fordham L. Rev.*, vol. 89, p. 1553, 2020.

[13] J. Uyheng and K. M. Carley, "Characterizing Network Dynamics of Online Hate Communities around the COVID-19 Pandemic," *Applied Network Science*, vol. 6, no. 1, pp. 1–21, 2021.

[14] F. Jordans, "Germany passes law against online hate speech," Jun 2017.

[15] S. J. Pan and Q. Yang, "A Survey on Transfer Learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, 2009.

[16] S. Ruder, *Neural Transfer Learning for Natural Language Processing*. PhD thesis, NUI Galway, 2019.

[17] X. Qiu, T. Sun, Y. Xu, Y. Shao, N. Dai, and X. Huang, "Pre-Trained Models for Natural Language Processing: A Survey," *arXiv preprint arXiv:2003.08271*, 2020.

[18] M. E. Peters, S. Ruder, and N. A. Smith, "To Tune or Not to Tune? Adapting Pretrained Representations to Diverse Tasks," in *Proceedings of the 4th Workshop on Representation Learning for NLP*, pp. 7–14, 2019.

[19] J. Howard and S. Ruder, "Universal Language Model Fine-tuning for Text Classification," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, pp. 328–339, 2018.

[20] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," in *Proceedings of the 17th Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 4171–4186, 2019.

[21] P. Nayak, "Understanding Searches Better than ever Before," *Google Blog, October*, vol. 25, 2019.

[22] D. Yin, Z. Xue, L. Hong, B. D. Davison, A. Kontostathis, and L. Edwards, "Detection of Harassment on Web 2.0," in *Proceedings of the Content Analysis in the WEB 2.0*, vol. 2, pp. 1–7, 2009.

[23] W. Warner and J. Hirschberg, "Detecting Hate Speech on the World Wide Web," in *Proceedings of the Second Workshop on Language in Social Media*, pp. 19–26, 2012.

[24] C. Nobata, J. Tetreault, A. Thomas, Y. Mehdad, and Y. Chang, "Abusive Language Detection in Online User Content," in *Proceedings of the 25th International Conference on World Wide Web*, pp. 145–153, 2016.

[25] Z. Waseem and D. Hovy, "Hateful Symbols or Hateful People? Predictive Features for Hate Speech Detection on Twitter," in *Proceedings of the NAACL Student Research Workshop*, pp. 88–93, 2016.

[26] P. Badjatiya, S. Gupta, M. Gupta, and V. Varma, "Deep Learning for Hate Speech Detection in Tweets," in *Proceedings of the 26th International Conference on World Wide Web*, pp. 759–760, 2017.

[27] B. Gambäck and U. K. Sikdar, "Using Convolutional Neural Networks to Classify Hate-Speech," in *Proceedings of the 1st Workshop on Abusive Language Online*, pp. 85–90, 2017.

[28] T. Davidson, D. Warmsley, M. Macy, and I. Weber, "Automated Hate Speech Detection and the Problem of Offensive Language," in *Proceedings of the International AAAI Conference on Web and Social Media*, vol. 11, 2017.

[29] J. Risch, A. Stoll, M. Ziegele, and R. Krestel, "hpiDEDIS at GermEval 2019: Offensive Language Identification using a German BERT model," in *Proceedings of the 15th Conference on NLP*, pp. 405–410, 2019.

[30] M. Mozafari, R. Farahbakhsh, and N. Crespi, "A BERT-based Transfer Learning Approach for Hate Speech Detection in Online Social Media," in *Proceedings of the International Conference on Complex Networks and Their Applications*, pp. 928–940, 2019.

[31] H. Mulki, H. Haddad, C. Bechikh Ali, and H. Alshabani, "L-HSAB: A Levantine Twitter Dataset for Hate Speech and Abusive Language," in *Proceedings of the 3rd Workshop on Abusive Language Online*, pp. 111–118, 2019.

[32] A. Paraschiv and D.-C. Cercel, "UPB at GermEval-2019 Task 2: BERT-Based Offensive Language Classification of German Tweets," in *Proceedings of the 15th Conference on NLP*, pp. 398–404, 2019.

[33] M. Corazza, S. Menini, E. Cabrio, S. Tonelli, and S. Villata, "A Multilingual Evaluation for Online Hate Speech Detection," *ACM Transactions on Internet Technology*, vol. 20, no. 2, 2020.

[34] B. Ross, M. Rist, G. Carbonell, B. Cabrera, N. Kurowsky, and M. Wojatzki, "Measuring the Reliability of Hate Speech Annotations: The Case of the European Refugee Crisis," *arXiv preprint arXiv:1701.08118*, 2017.

[35] J. C. Pereira-Kohatsu, L. Q. Sánchez, F. Liberatore, and M. Camacho-Collados, "Detecting and Monitoring Hate Speech in Twitter," *Sensors*, vol. 19, no. 21, 2019.

[36] J. Cho, K. Lee, E. Shin, G. Choy, and S. Do, "How Much Data is needed to Train a Medical Image Deep Learning System to Achieve Necessary High Accuracy?," *arXiv preprint arXiv:1511.06348*, 2015.

[37] S. J. Raudys and A. K. Jain, "Small Sample Size Effects in Statistical Pattern Recognition: Recommendations for Practitioners," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 3, pp. 252–264, 1991.

[38] M. Sordo and Q. Zeng-Treitler, "On Sample Size and Classification Accuracy: A Performance Comparison," in *Biological and Medical Data Analysis*, pp. 193–201, 2005.

[39] H. He and E. A. Garcia, "Learning from Imbalanced Data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 21, no. 9, pp. 1263–1284, 2009.

[40] J. Hestness, S. Narang, N. Ardalani, G. Diamos, H. Jun, H. Kianinejad, M. Patwary, M. Ali, Y. Yang, and Y. Zhou, "Deep Learning Scaling is Predictable, Empirically," *arXiv preprint arXiv:1712.00409*, 2017.

[41] D. Jurafsky and J. H. Martin, *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition*. Unpublished, 3 ed., 2020.

[42] M. Wiegand, M. Siegel, and J. Ruppenhofer, "Overview of the GermEval 2018 Shared Task on the Identification of Offensive Language," in *Proceedings of the 14th Conference on NLP*, 2018.

[43] J. M. Struß, M. Siegel, J. Ruppenhofer, M. Wiegand, M. Klenner, *et al.*, "Overview of GermEval Task 2 Shared Task on the Identification of Offensive Language," in *Proceedings of the 15th Conference on NLP*, 2019.

[44] M. Duggan and J. Brenner, *The Demographics of Social Media Users, 2012*, vol. 14. Pew Research Center's Internet & American Life Project, 2013.

[45] J. Ruppenhofer, M. Siegel, and M. Wiegand, "Guidelines for IGGSA Shared Task on the Identification of Offensive Language," 2018.

[46] T. Kudo and J. Richardson, "SentencePiece: A Simple and Language Independent Subword Tokenizer and Detokenizer for Neural Text Processing," *arXiv preprint arXiv:1808.06226*, 2018.

[47] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is All You Need," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, p. 6000–6010, 2017.