Overview of Abusive and Threatening Language Detection in Urdu at FIRE 2021

Maaz Amjad[a], Alisa Zhila[b], Grigori Sidorov[a], Andrey Labunets[c], Sabur Butt[a], Hamza Imam Amjad[d], Oxana Vitman[a] and Alexander Gelbukh[a]

[a]Instituto Politécnico Nacional (IPN), Center for Computing Research (CIC), Mexico
[b]Ronin Institute for Independent Scholarship, United States
[c]Independent Researcher, United States
[d]Moscow Institute of Physics and Technology, Russia

Abstract

With the growth of social media platform influence, the effect of their misuse becomes more and more impactful. The importance of automatic detection of threatening and abusive language can not be overestimated. However, most of the existing studies and state-of-the-art methods focus on English as the target language, with limited work on low- and medium-resource languages. In this paper, we present two shared tasks of abusive and threatening language detection for the Urdu language that has more than 170 million speakers worldwide. Both are posed as binary classification tasks where participating systems are required to classify tweets in Urdu into two classes, namely: (i) Abusive and Non-Abusive for the first task, (ii) Threatening and Non-Threatening for the second. We present two manually annotated datasets containing tweets labeled as: (i) Abusive and Non-Abusive, (ii) Threatening and Non-Threatening. The abusive dataset contains 2400 annotated tweets in the train part and 1100 annotated tweets in the test part. The threatening dataset contains 6000 annotated tweets in the train part and 3950 annotated tweets in the test part. We also provide logistic regression and BERT-based baseline classifiers for both tasks. In this shared task, 21 teams from six countries registered for participation (India, Pakistan, China, Malaysia, United Arab Emirates, Taiwan), 10 teams submitted their runs for Subtask A —Abusive Language Detection, 9 teams submitted their runs for Subtask B —Threatening Language detection, and seven teams submitted their technical reports. The best performing system achieved an F1-score value of 0.880 for Subtask A and 0.545 for Subtask B. For both subtasks, m-Bert based transformer model showed the best performance.

Keywords

Natural language processing, text classification, Twitter tweets, Urdu language, shared task, abusive language detection, threatening language detection

1. Introduction

In cyberspace, abusive and threatening content is a glaring problem that has been present since the beginning of human interaction on the internet and will continue to persist in future. Social media platforms today are the venues for free expression for all communities, and community

backlashes can result in a lot of negative externalities. Thus, with the growth of social media platforms and their audiences, regulating threatening and abusive content becomes a concern for the welfare of all stakeholders. Though leading platforms such as Twitter and Facebook have set up community standards for the prevention of cybercrimes, early detection of such content is vital for the safety of cyberspace.

Detection of abusive and threatening text is a complex problem as the platforms find it challenging to maintain a balance between limiting the abuse and giving users ample freedom to express themselves. Failing to meet the balance can result in users losing trust in the platform as well as disengagement with the content. Platforms also find it challenging to detect texts in multiple languages, especially low resourced languages and code mixed languages. Manual filtering and understanding of this content is logistically daunting and resource unfriendly. It can also result in the delay of necessary and timely action needed in case of active threats and abuse. Hence, Natural Language Processing (NLP) researchers have been working on the early detection of threats and abuse by providing various automated solutions based on machine learning and deep learning in particular.

Several studies previously have attempted to deal with the problem of abusive language [1, 2] and threat detection [3, 4]. These problems have been attempted through supervised machine learning [5, 6, 7] and deep learning [8, 9, 10] approaches breaking it down into binary, multi-label, or multi-class classification problems. However, these attempts are only limited to European languages, Arabic, and a few South Asian languages such as Hindi, Bengali, and Indonesian.

Here we present a new shared task for abusive and threatening language detection in tweets written in Urdu. The task is aimed at driving attention and effort of the research community to developing more efficient methods and approaches for this vastly spoken language and highlighting difficulties that specific to the writing system and the use of Urdu. The paper describes the abusive and threatening language tracks [1] organized by the authors within the Hate Speech and Offensive Content Identification (HASOC) evaluation tracks of the 13th meeting of Forum for Information Retrieval Evaluation (FIRE) 2021 [2] and co-hosted by Open Data Science (ODS) Summer of Code initiative 2021 [3]. The task is comprised of two sub-tasks:

1. Sub-task A: **Abusive language detection** [4]. The task offered a dataset of Twitter posts ("tweets") in Urdu language split into the training part with the annotations available to participants and the testing part provided without annotations. The dataset annotation procedure followed Twitter's definition of abusive language [5] to identify posts that are abusive towards a community, group, or an individual as the ones meant to harass, intimidate or silence someone else's voice. The tweets were annotated in a binary manner, i.e., *abusive* or *non-abusive*. The participants were asked to determine the correct labels for the testing part and submit their annotations. The solutions were evaluated using F1 and ROC-AUC metrics.

2. Sub-task B: **Threatening language detection** [6]. Similarly, the task offered a dataset

---

of tweets in Urdu annotated as *threatening* or *non-threatening* split into the training and testing parts, with the annotations of the testing part hidden from the participants. The annotation procedure for the Sub-task B dataset followed Twitter's definition of threatening tweets [7] as those that are against an individual or group meant to threaten with violent acts, to kill, inflict serious physical harm, to intimidate, or to use violent language. The task and the evaluation procedure were identical to Sub-task A

With these shared tasks our contributions are:

- spreading awareness and motivating the community to propose more efficient methods for automated detection of abusive and threatening messages in social media in Urdu as well as providing means for standardized comparison as emphasized in Section 2;
- collection and annotation of the largest so far datasets for abuse and threat detection in the Urdu language described in Section 4 , in particular, 3500 tweets annotated as abusive or not and 9,9500 tweets annotated as threatening or not;
- the train and test split that allows for a fair result comparison (see Section 5 for details and grounds) not only among the current participants but also for future research;
- provision of highly competitive baseline classifiers in Section 7;
- overview and comparison of the submitted solutions for abusive language and threat detection in Urdu in Sections 8 and 9.

2. Importance of Identifying Abuse and Threat in Urdu

Urdu is one of the largest spoken languages in South Asia. It is the national language of Pakistan and has its roots in Persian and Arabic bearing additional structural similarities with many languages from other language families, e.g., Hindi [11, 12]. Urdu is spoken by more than 170 million [8] people worldwide and the number is increasing every day. Yet it lacks solutions and resources for the most essential natural language processing problems.

Urdu is mostly written using the Nastalíq script. However, certain populations also use the Devanagari script which is normally used for writing Hindi. Hence, Urdu texts experience the phenomenon of *digraphia* that is the use of more than one writing system for the same language. Additionally, Urdu is quite complex linguistically [13] as its morphological and syntactic structure is a combination of Turkish, Arabic, Persian, Sanskrit, and English. Hence, contributing to Urdu is also fundamental for the success of other languages.

Population of the Urdu speaking countries have substantial access to social media, and millions of speakers are exposed to unregulated or poorly regulated hate, abuse, and threats. Various extremist and terrorist groups have developed communities on social media platforms that spread abuse, threats, and terror [14]. As they post in local languages, in particular, in Urdu, much of the content is left unchecked until reviewed and reported manually. Pakistan suffered decades of terrorism and had to resort to banning social media on several occasions to tackle terrorism [15]. Hence, the development of resources in Urdu for threat and abuse detection is an urgent requirement for the safety of millions.

---

[7] https://help.twitter.com/en/rules-and-policies/violent-threats-glorification
[8] https://www.ethnologue.com/language/urd

3. Literature Review

Offensive content encompasses a variety of phenomena including aggression [16, 17], sexism [18], hate speech [19, 5], threat detection [4?], toxic comment detection [20], abusive language detection [1, 2] and many others. Previous research [21] have attempted to distinguish various types of abuse such as implicit vs. explicit abuse or identity vs. person-directed abuse to identify more nuanced expressions of abuse.

Multiple annotated datasets are available for a variety of offensive content phenomena sourced from numerous social media platforms and portals. Yahoo Finance corpus [19] comprises English language texts from the Yahoo's finance portal which is annotated into two classes: clean and hate speech. Research [5] collected a dataset of Twitter posts in English and annotated them into three classes as sexism, racism, or neither. Similarly, work [22] also annotated tweets in English into three classes, yet different from work [5]: hate speech, offensive language, and neither. On the contrast, study [23] distinguished four offensive classes in their collection of Twitter posts in English: hateful, spam, abusive, and neutral.

Youtube has been another source for data collection for abusive language in English [20, 24] as well as in other languages, in particular, Arabic [2]. In particular, the study by Ashraf et al. [4] is based on the YouTube comment and replies collection introduced by Hammer et al. [24] with additional annotation of its subset as whether a threat is directed towards a group or an individual. Another study [25] collects a dataset of 2,304 YouTube comments with 6,139 replies in English and annotates it in two ways: a binary annotation for abusive language as well as a three class annotation for topics: politics, religion, and other.

Attempts have been made to create threat and abuse detection models for Bengali language [3]. Posts and comments have been collected from different pages of Facebook. Threatening and abusive language was labeled as "YES" in the dataset and the rest of the data which is not abusive was labeled as "NO". For more detailed analysis of the available datasets, we recommend these studies [26, 1, 27].

Apart from the papers proposing a single solution, a number of shared tasks have been organized to incentivize creation of multiple robust systems for offensive phenomena detection in texts. Some of the popular shared tasks are OffensEval [28, 29] with available datasets in Greek, English, Danish, Arabic, Turkish, and English; GermEval 2018 [30] for texts in German; TRAC shared task [31] for Hindi, English, and Bengali; SemEval-2019 [32] for hate speech detection in English and Spanish; HASOC 2019 and 2020 [33, 34] for German, English, Tamil, Malayalam, and Hindi.

Among the common approaches for offensive language detection, we observe feature-based approaches with traditional ML classifiers. Works [6, 7, 5, 35, 33, 34, 32, 29, 28] use various combinations of features such as N-grams, Bag-of-Words (BOW), Part-of-Speech (POS) tags, Term Frequency—Inverse Dense Frequency (TF-IDF) representation, word2vec representation, sentiments, and dependency parsing features provided as input to the traditional ML models such as Support Vector Machines (SVM), Logistic Regression (LR), Random Forest (RF), Decision Tree (DT), Naive Bayes (NB), etc.

Among the more efficient approaches for the task, we see boosting-based ensembles as well as neural networks, in particular, deep NNs such as transformers. For example, Ashraf et al. [25, 4] used n-gram and pre-trained word embeddings in combination with traditional ML

(LR, RF, SVM, NB, DT, VotingClassifier, and the boosting-based ensemble AdaBoost) as well as Neural Network based methods (MLP, 1D-CNN, LSTM, and Bi-LSTM) for the abusive language detection and for the prediction of an individual- vs. group-targeted threat correspondingly. While the BiLSTM approach achieved an F1 score of 85%, the use of conversational context along with the linguistic features achieved even higher F1 score of 91.96% using an ensemble AdaBoost classifier.

BiLSTM and Convolutional Neural Networks (CNN) were used to tackle abusive language and hate speech detection in multiple other works. Studies employing graph embeddings to learn graph representations from online texts [9], paragraph2vec [19], and Recurrent Neural Networks (RNN) with attention [10], RNN with Gated Recurrent Units (GRUs) [8] have also shown encouraging results. The pre-trained transformer methods such as RoBERTa, BERT, ALBERT and GPT-2 to detect hate speech detection can be seen achieving high accuracies [36, 37, 38]. A recent study [39] applied XLM, BERT and BETO models to achieve promising results on similar tasks for hate speech detection.

While each offensive subcategory uses different definitions for annotation, similar methods can be applied across the offensive content detection tasks. All these techniques can be used to test the best combinations for detection of abuse and threat in the Urdu language[40, 41] and our study opens vast avenues for researchers to achieve this goal.

4. Datasets Collection and Annotation

4.1. Threatening and Abusive Datasets Collection

In the beginning, we created a dictionary of most used abusive and threatening words in Urdu. We used those words as keywords on Twitter to mine tweets containing more abusive and threatening words in Urdu, which we manually added to our dictionary. The dictionary includes words that appeared even ones to threat or abuse someone. This dictionary is publicly available for research purposes[9].Thus, we collected a sufficient number of abusive and threatening seed words which were further used to crawl tweets through the Twitter Developer Application Programming Interface (API)[10] using Tweepy library. Thus, we gathered enough words and phrases that are used to threat or abuse individuals. We collected tweets containing any of these keywords from our dictionary for a 20 month period from January 1st, 2018 to August 31st, 2019. At this time the general elections being held in Pakistan in July 2018. Typically, during the election season, people tend to be more expressive when supporting as well as opposing political parties. In total we crawled 55,600 number of tweets containing the seed words.

4.2. Threatening and Abusive Datasets Pre-processing

Since Urdu shared many common words in Persian, Turkish and Arabic, so when we crawled tweets using our initially collected words, the Twitter APA also crawled many non-Urdu tweets. Since this research was primary focused on Urdu lanaguage, we discarded all the

---

[9]https://github.com/MaazAmjad/Threatening$_D$ataset
[10]https://developer.twitter.com/en/docs/twitter-api/v1/tweets/search/api-reference/get-search-tweets

non-Urdu tweets manually. Thus, two different datasets have been created: (i) abusive dataset[11], containing 3,500 tweets, 1,750 of them are abusive and 1,750 of them are non-abusive (ii) threatening dataset[12]; containing 9,950 tweets, 1,782 threatening tweets and remaining tweets are non-threatening.

4.3. Threatening and Abusive Datasets Annotation

We defined guidelines to annotate abusive and threatening tweets.

To annotate the dataset the annotators have been recruited. All of them satisfied the following criteria: (i) country of origin - Pakistan; (ii) native speakers of Urdu; (iii) are familiar with Twitter; (iv) aged 20–35 years; (v) detached from any political party or organization; (vi) have prior experience of annotating data; (vii) educational level was a masters degree or above. We computed Inter-Annotator Agreement (IAA) using Cohen's Kappa coefficient [39] as it is a statistic measure to check the reliability between two annotators. We provided instructions with task definitions (which are reproduced below) and examples. Hierarchical annotation schema was used and the main dataset was divided into two different datasets to distinguish between whether the language is threatening ot non-threatening, abusive or non-abusive. We followed Twitter definition to describe abusive[13] and threatening[14] comments towards an individual or groups to harass, intimidate, or silence someone else's voice.

5. Training and Testing Dataset Split

Due to the requirements extended by the competition conditions and in purpose of fair evaluation of the participant's submission, a slightly larger portion of the datasets was withheld as corresponding testing parts than it would be done under 'normal' data science operations. Namely, 40% of the data was withheld for the Threatening Language task, and 32%, for the Abusive Language task. This is done, first of all, to ensure that the testing set is non-trivial and represents well the variety of possible lexical expressions for both classes. Second, during the active period of the competition, the participants could observe the scores only from the "public" part of the testing set, whereas the scores on the "private" part of the testing set were made public only after the end of the competition. The partitioning of the test set into public and private is necessary to avoid pure guessing or tampering with predictions. We ensured that each partition of the testing data was large enough to compute a score that is sufficiently reflective of the actual performance of a system. The details are presented in Table 1.

To be clear, the participants were handed out the *entire* test set without true labels. After a submission, the scores were shown only for the public partition of the test set. As it can be observed from Tables 5 and 6, there still was some amount of shake-up among the scores and corresponding ranks on the public and private partitions.

Now that both the training and the testing sets along with their true labels are available to the research community, a different approach to train/test split may be possible. However,

---

[11]https://github.com/MaazAmjad/Urdu-abusive-detection-FIRE2021
[12]https://github.com/MaazAmjad/Threatening$_D$ataset
[13]https://help.twitter.com/en/rules-and-policies/abusive-behavior
[14]https://help.twitter.com/en/rules-and-policies/glorification-of-violence

Table 1
Descriptive table of the train/test split for abusive and threatening language datasets. In parentheses the fraction of positive, *i.e.*, abusive or threatening respectively, labels is indicated.

Dataset	train	test	test$_{PUB}$	test$_{PRIV}$
Abusive	2400 *(49.50%)*	1100 *(48.81%)*	400 *(48.50%)*	700 *(47.29%)*
Threat	6000 *(17.85%)*	3950 *(18.20%)*	1000 *(15.80%)*	2950 *(19.02%)*

for a fair comparison with the competition submissions and results provided in this paper, we suggest following the original split.

6. Evaluation Metrics

The submitted systems were evaluated by comparing the labels predicted by the participants' classifiers to the hidden ground truth annotations. For quantifying the classification performance, we computed the commonly used evaluation metrics: F1 score and ROC-AUC score. F1 score serves as a better metrics for unbalanced datasets than Accuracy and, therefore, accommodates our settings. The ROC-AUC score gives an estimate of the overall quality of the model at the various level of predicted confidence thresholds and serves as a more holistic evaluator.

7. Baselines

For the competition, the organizers prepared three baseline systems: two of them reflected different aspects of traditional ML approach involving Bag-of-Words features and were meant to be lower boundary scoring baselines while the third system was based on the recent deep learning approach involving fine-tuning of the BERT model [36].

7.1. LogReg with Lexical Features

All data pre-processing steps and most of the modeling details are identical for both subtasks, abusive and threat detection, if not explicitly indicated otherwise.

First, all possible word unigrams and bigrams were extracted from the training dataset using the popular NLTK[15] [42] software package for NLP, v. 3.4.5, counting the numbers for n-gram occurrences in the dataset. Further, the occurrence threshold of 3 was applied to unigrams corresponding to the 75th-percentile of all encountered unigrams. In other words, we took the top 25% of most frequently occurring unigrams as features. Similarly, the 95th-percentile occurrence threshold of 4 was applied to bigrams. We also added 2 additional features to account for Out-Of-Vocabulary (OOV) unigrams and bigrams correspondingly. Eventually, the feature set was comprised of the top occurring unigram features, top occurring bigram features, and the two OOV features. The statistics for each feature type by the subtask dataset and the total number of features is provided in Table 2.

---

[15]https://www.nltk.org

Table 2
Feature type statistics for each subtask for the 1st baseline solution.

Subtask	unigrams	bigrams	OOV features	total features
Abusive	1775	1212	2	2989
Threat	3640	3413	2	7055

Further, each tweet instance was represented as a straightforward count of feature occurrences in the tweet, all OOV n-grams counting towards corresponding special OOV features. No normalization was done as all tweets have approximately the same length.

Logistic regression was selected as the classifier algorithm for our traditional ML baseline solutions. In the 1st system, we used the implementation from scɪkɪt-learn[16] [43] v. 0.22.1, which is a popular software package that includes a number of ML algorithms. The `max_iter` parameter was set to 1000 to make sure the training converges.

For the Threat Subtask dataset, where the positive and negative classes are imbalanced, we also set the `class_weight` parameter to *balanced* which ensured automatic instance reweighing.

The code is available at the organizers's GitHub repository[17].

The balanced baseline secured the 8th place on the Threat Subtask private leaderboard with F1-score equal to 0.49186, ROC-AUC, to 0.76991. The unbalanced version applied in the Abusive Subtask came 12th on the private leaderboard scoring 0.78684 F1-score, 0.88295 ROC-AUC.

7.1.1. A version of LogReg with lexical features and TF-IDF count

We also submitted a variation of the Log-Reg based classifier with a few technical as well as conceptual modifications. Instead of a simple *n*-gram occurrence count, the TF-IDF vectorization approach was used for text representation. For this, the `TfidfVectorizer` function from the scɪkɪt-learn package was used. It is to note that the types of features were unigrams only. The number of features was set as in the previous approach.

Another purely technical difference was that the LogReg classifier was implemented as a "single node neural network" which is algorithmically and equationally equivalent to logistic regression.

The implementation was done using the PyTorch framework[18] [44]. This training set-up converged much sooner, with mere 30 epochs, or in terminology of traditional ML, iterations, for both datasets. The optimal number of epochs was determined using a validation dataset which was 10% of the corresponding training data.

For the threatening language detection dataset, similarly to the previous approach, the dataset balancing was performed by applying `torch.nn.BCEWithLogitsLoss` function.

These differences in approaches were reflected in the final score difference. Interestingly, for the abusive language detection task, while this variant showed slightly higher scoring (0.77008 F1-score for this version vs 0.72928 F1 for the above version, and 0.86674 vs 0.85286 ROC-AUC)

---

[16]https://scikit-learn.org
[17]https://github.com/UrduFake/urdutask2021/
[18]https://pytorch.org

and, hence, the rank (11th *vs.* 13th) on the public leaderboard, it actually showed same scores on the private leaderboard, 0.78684 F1 and 0.88295 ROC-AUC, to the extent of decimal precision displayed, sharing the 12th and 13th ranks.

More notably, in the threatening language detection task, the results and scores returned by the two versions, not only varied largely, but the score difference of the systems actually flipped significantly between the private and public leaderboards. On the public leaderboard, the scɪkɪt-learn version gained higher scores: 0.46471 F1 *vs.* 0.45161 F1 for the PyTorch version, and 0.79502 ROC-AUC *vs.* 0.78899 ROC-AUC for the PyTorch version. Yet on the private leaderboard the scɪkɪt-learn version gained less: 0.49186 F1 *vs.* 0.51404 F1 for the PyTorch version, and 0.76991 ROC-AUC *vs.* 0.78212 ROC-AUC, respectively.

This brings us to a likely conclusion that, no matter the ML package, for the abusive task, the LogReg classifier along with the lexical bag-of-word features is a sufficiently powerful tool that can properly converge on the provided dataset learning a coherent pattern.

However, the threat detection task is a more complex task not only due to the label imbalance but also due to the intrinsic semantic complexity of the phrases, the latter having a much larger effect. Therefore, simple classifiers and purely lexical features are too weak to capture higher levels of semantic complexities and should not be relied on for this subtask.

7.2. BERT-based baseline

The dataset sizes of 2400 and 6000 items along with training example length below 200 characters made the tasks approachable with transfer learning-based methods using foundational deep BERT-like models.

The proposed deep learning-based solutions for both subtasks, Abusive and Threat detection, used pretraɪned multɪlɪngual uncased BERT[19] [36] from huggingface transformers library [45] as a base model.

Huggingface "built-in" `BertForSequenceClassification`[20] class with 2 output units was selected as a classification head, where pooled output from [CLS] token is passed through a dropout layer, followed by a linear layer with output units leading into cross-entropy loss function.

For the Abusive Subtask, we split the provided training dataset into TRAIN/DEV sets via a standard 80:20 ratio. Using the TRAIN set, the model is further fine-tuned for the target classification task for 3 epochs with minibatch size of 32 and 60 minibatches per epoch. The total number of minibatches, and correspondingly optimization steps, was 180. The fine-tuning process used the DEV set to evaluate the model performance every 4 minibatches in order to load a model with the best F1 score from checkpoints at the end of the fine-tuning.

For the Threat Subtask, we split the provided training dataset into TRAIN/DEV sets via a 85:15 ratio. We deviated from the standard 80:20 split to let the model train with slightly more data and more negative examples as a result at the cost of less accurate F1 score. Using the TRAIN set, the model was later fine-tuned for the target task for 5 epochs with minibatch size of 32 and 160 minibatches per epoch (total number of minibatches / optimization steps was

---

[19]https://huggingface.co/bert-base-multilingual-uncased
[20]https://github.com/huggingface/transformers/blob/27d4639779d2d316a7c5f18d22f22d2565b84e5e/src/transformers/models/bert/modeling_bert.py#L1486

800). In our set-up, the model for the Threat Subtask converged slower than the one for the Abusive subtask, therefore we trained the network for 5 epochs instead of 3. The cross-entropy loss function additionally used inverse class sizes as weights to account for imbalance. The fine-tuning used the DEV set to evaluate the model every 8 minibatches (not 4 due to longer training) in order to load a model with the best F1 score from checkpoints at the end of the fine-tuning.

The first baseline model for the Abusive Subtask came 3rd on the private leaderboard with F1-score equal to 0.86221, ROC-AUC to 0.92194. The second baseline model for the Threatening Subtask came 9th on the private leaderboard scoring 0.48567 F1-score, 0.70047 ROC-AUC. Considering the original BERT's [36] scores at GLUE and other benchmarks, as well as further progress in language model pretraining [46], the first model's relatively high F1 score was expected. The Abusive Subtask was a sentence classification task with little specific constraints (such as overly large sequence length or similar obstacles), where deep bidirectional architecture-based and other large pretrained language models generally outperform traditional machine learning approaches in a number of domains. At the same time, better handling of class imbalance in the Threat Subtask could help the second baseline model achieve better convergence and a higher F1 score. We speculate, that domain-specific improvements at preprocessing, additional intermediate-task training, and complementary handcrafted features used along with the sentence embeddings can further boost the score for both models. In other words, subject matter knowledge of language and relevant threat landscape is indispensable for real-world threat and abuse detection in Urdu language. Finally, we see incorporating continued training and more domain-specific research in adversarial training, out-of-distribution detection, and outlier detection as viable directions to make a model robust to adversarial examples and distribution shifts when it is deployed.

The code for this baseline is available on organizers' GitHub repository[21].

8. Overview of Submitted Solutions

This section gives a brief overview of the systems submitted to this competition. 21 teams registered for participation, 10 teams submitted their runs for Subtask A —Abusive Language Detection, 9 teams submitted their runs for Subtask B —Threatening Language detection. Registered participants were from different countries: India, Pakistan, China, Malaysia, United Arab Emirates, Taiwan. This wide range of the regions where the interested participants were located confirms the importance of the task. The team members came from various types of organizations: universities, research centers, and industry.

8.1. Approaches to Text Representation

Participants used a variety of text representation techniques for tweet representation. Team SAKSHI SAKSHI represented tweets using contextual embedding representations that were obtained from training on an Urdu news corpus. Individual participant Muhammad Hamayoun used traditional bag-of-words representation for Subtask A and word2vec for word n-grams,

---

Table 3

Approaches used by the participating systems for Subtask A: Abusive language detection

System/Team Name	Feature Type	Feature Weighting Scheme	Classifying algorithm	NN-based
hate-alert	laser embedding	*N/A*	multi-lingual BERT	Yes
SAKSHI SAKSHI	contextual embedding	*N/A*	Urduhack, BERT, and XLM-RoBERTa	Yes
Muhammad Hamayoun	word 1, 2-grams	frequency	SVM (sigmoid kernel)	No
Alt-Ed	BoW	TF-IDF	LogReg	No
Abhinav Kumar	char 1, 6-grams	TF-IDF	ensemble SVM+LogReg+RF	No
SSNCSE_NLP (*late subm.*)	char 10-gram	TF-IDF	MLP	Yes

$n = 1, 2$, for Subtask B. The hate-alert team used pre-trained Urdu laser embeddings and multi-lingual BERT[22] pre-trained embeddings generated from an Arabic dataset. Team Alt-Ed used TF-IDF text representation. Participant Abhinav Kumar used $1, 6$-gram character level TF-IDF features for tweet representation. A summary of approaches is presented in Tables 3, 4.

Table 4

Approaches used by the participating systems for Subtask B: Threatening language detection

System/Team Name	Feature Type	Feature Weighting Scheme	Classifying algorithm	NN-based
hate-alert	laser embedding	*N/A*	multi-lingual BERT	Yes
SAKSHI SAKSHI	contextual embedding	*N/A*	RoBERTa	Yes
Muhammad Hamayoun	word2vec embedding	*N/A*	SVM (poly d=3)	No
Abhinav Kumar	char 1, 6-grams	TF-IDF	ensemble SVM+LogReg+RF	No
SSNCSE_NLP*	char 10-gram	TF-IDF	MLP	Yes

8.2. Classification Methods

To implement their classifiers, some participating teams used the traditional, i.e., non-neural network based machine learning algorithms, while other teams' submissions were based on various neural network architectures.

For Subtask B, team SAKSHI SAKSHI fine-tuned a pre-trained RoBERTa model from the popular HuggingFace library[23] on the Urdu news corpus in an unsupervised manner. The same team used three transformer-based techniques for Subtask A: (i) Urduhack, (ii) BERT, and (iii) XLM-Roberta. Team hate-alert used Hate-speech-CNERG/dehatebert-mono-arabi[24] model which is preliminary fine-tuned on an Arabic hate speech dataset. Another participant, Muhammad Humayoun, used SVM with sigmoid kernel for Subtask A and SVM with polynomial kernel of degree 3 for Subtask B. Participant Abhinav Kumar used an ensemble of ML models SVM + LogReg + RF for both subtasks. Similarly to one of the organizers' baseline systems, team Alt-Ed used Logistic Regression for Subtask A, which turned out to be team's best classifier for the task.

A summary of approaches is presented in Tables 3, 4.

---

[22]https://huggingface.co/bert-base-multilingual-cased
[23]https://huggingface.co/transformers/model_doc/roberta.html
[24]https://huggingface.co/Hate-speech-CNERG/dehatebert-mono-arabic

Table 5
Final results and ranking for Subtask A: Abusive language detection

Team Name	Rank	Private Leaderboard		Public Leaderboard	
		F1	ROC AUC	F1	ROC AUC
hate-alert	1	0.880	0.924	0.853	0.920
SAKSHI SAKSHI	2	0.868	0.935	0.839	0.934
Org's BERT-based sol., out-of-comp.	*N/A*	0.862	0.921	0.846	0.923
SATLab	3	0.856	0.917	0.827	0.910
Jie Yi Xiang	4	0.853	0.914	0.842	0.915
Vlad Balanda	5	0.849	0.945	0.824	0.901
Muhammad Humayoun	6	0.825	0.824	0.805	0.9820
Igor Shatalin	7	0.821	0.207	0.797	0.184
Alt-Ed	8	0.820	0.890	0.770	0.874
Abhinav Kumar	9	0.808	0.907	0.808	0.907
ai	10	0.791	0.908	0.743	0.875
Org's LogReg v1 sol., out-of-comp.	*N/A*	0.786	0.882	0.730	0.853
Org's LogReg v2 sol., out-of-comp.	*N/A*	0.786	0.882	0.770	0.866
SSNCSE_NLP, *late subm.*	*N/A*	0.771	0.757	0.689	0.699

Table 6
Final results and ranking for Subtask B: Threatening language detection

Team Name	Private Leaderboard			Public Leaderboard	
	Rank	F1	ROC AUC	F1	ROC AUC
hate-alert	1	0.545	0.810	0.489	0.798
SATLab	2	0.545	0.815	0.548	0.814
Somnath Banerjee	3	0.518	0.791	0.475	0.776
Org's LogReg v2 sol., out-of-comp.	*N/A*	0.514	0.782	0.451	0.788
Muhammad Humayoun	4	0.501	0.704	0.461	0.698
Jie Yi Xiang	5	0.494	0.770	0.463	0.771
SAKSHI SAKSHI	6	0.492	0.781	0.534	0.819
Org's LogReg v1 sol., out-of-comp.	*N/A*	0.491	0.769	0.464	0.795
Org's BERT-based sol., out-of-comp.	*N/A*	0.485	0.700	0.519	0.765
Vlad Balanda	7	0.478	0.799	0.444	0.786
Abhinav Kumar	8	0.349	0.735	0.425	0.745
Owais Raza	9	0.551	0.757	0.129	0.740
SSNCSE_NLP, *late subm.*	*N/A*	0.805	0.657	0.825	0.661

9. Results and Discussion

Table 5 presents results and ranking for Abusive Language detection subask. Table 6 presents results and ranking for Threatening Language detection subtask. The systems are ranked by their F1 score on the private leaderboard.

We observe that except for one participant system, all the other participating teams' systems outperformed the proposed LogReg baselines in terms of F1 score for Subtask A. However, only two systems, hate-alert's and SHAKSHI SHAKSHI's, outperformed the proposed BERT-based baseline. For Subtask B, on the contrast, quite a few systems scored below the described LogReg baseline solutions. Interestingly, even the organizers' BERT-based solution did not achieve higher scores than the LogReg baselines despite that the size of the training dataset for Subtask B was larger than the one for Subtask A. Eventually, only the top 3 systems, hate-alert, SATLab, and participant Somnath Banerjee, achieved higher F1 scores than the organizers' Keras-based implementation of Logistic Regression described in Section 7.1.1. Interestingly, although the two LogReg-based baselines score closely on Subtask A, their scores differ substantially for Subtask B. It might be due to the different values of the number of iterations parameter that permitted the LogReg-v2 system to converge on the larger training set in Subtask B, while in Subtask A LogReg convergence arrives sooner, partly due to a smaller training set size.

Among all the submitted runs for both sub-tasks, the hate-alert team's solution achieved the best F1 score and ranked highest. Their solutions are based on mBERT dehatebert-mono-arabıc[25] model that is trained on an Arabic news corpus. It is plausible that the combination of a powerful deep learning model and fine-tuning on a relevant, although somewhat unexpectedly, dataset was key for the high performance. These results may open a way to further research about the effect of direct knowledge transfer among languages that use the same script, in particular, Nastalíq.

Table 7

Aggregated performance statistics for Subtask A: Abusive language detection. *Numbers after a slash include organizers' solutions.*

Agg. Stats. and Percentiles	*Private Leaderboard*		*Public Leaderboard*	
	F1 Score	**ROC AUC**	**F1 Score**	**ROC AUC**
mean	0.831/*0.827*	0.830/*0.844*	0.800/*0.796*	0.827/*0.839*
std	0.033/*0.035*	0.214/*0.190*	0.049/*0.049*	0.225/*0.199*
min	0.771/*0.771*	0.207/*0.207*	0.689/*0.689*	0.184/*0.184*
10%	0.791/*0.786*	0.757/*0.777*	0.743/*0.734*	0.699/*0.745*
25%	0.814/*0.795*	0.857/*0.882*	0.784/*0.770*	0.875/*0.868*
50%	0.825/*0.823*	0.908/*0.907*	0.808/*0.806*	0.907/*0.904*
75%	0.855/*0.855*	0.921/*0.920*	0.833/*0.836*	0.917/*0.919*
90%	0.868/*0.866*	0.935/*0.932*	0.842/*0.845*	0.934/*0.931*
max	0.880/*0.880*	0.945/*0.945*	0.853/*0.853*	0.982/*0.982*

Overall for Subtask A, 75% of the participating systems obtained F1 score higher than 0.814 as it can be observed from the 25th percentile in Table 7. This is a good indicator that the task of abuse detection for tweets in Urdu can be achieved by automated means. In Table 5 we also observe that most of the top performing systems achieve both better F1 score and better ROC AUC for Subtask A.

In contrast, the task of threat detection for tweets in Urdu turned out to be extremely challenging as more than 90% of the systems could not pass the 0.8 F1 score bar as may be

---

[25]https://huggingface.co/Hate-speech-CNERG/dehatebert-mono-arabic

Table 8
Aggregated performance statistics for Subtask B: Threatening language detection. *Numbers after a slash include organizers' solutions.*

Agg. Stats. and Percentiles	Private Leaderboard		Public Leaderboard	
	F1 Score	**ROC AUC**	**F1 Score**	**ROC AUC**
mean	0.528/*0.521*	0.762/*0.759*	0.479/*0.479*	0.761/*0.766*
std	0.113/*0.099*	0.050/*0.048*	0.168/*0.146*	0.051/*0.045*
min	0.349/*0.349*	0.657/*0.657*	0.129/*0.129*	0.661/*0.661*
10%	0.465/*0.479*	0.699/*0.701*	0.395/*0.429*	0.694/*0.706*
25%	0.492/*0.491*	0.741/*0.735*	0.448/*0.451*	0.741/*0.745*
50%	0.510/*0.501*	0.776/*0.770*	0.469/*0.464*	0.774/*0.776*
75%	0.545/*0.545*	0.797/*0.791*	0.523/*0.519*	0.795/*0.795*
90%	0.576/*0.550*	0.810/*0.808*	0.576/*0.545*	0.815/*0.811*
max	0.805/*0.805*	0.815/*0.815*	0.825/*0.825*	0.819/*0.819*

observed in Table 8. Nevertheless, the top performing system SSNCSE_NLP achieving F1 score of 0.805 (Table 6) provides a promising perspective that this task is also solvable with the current methods and means of NLP available for the Urdu language.

However, at this moment it is still too soon to judge whether any of these approaches are ready to be applied "in the wild". While the results of over 0.88 F1 score shown by the winning hate-alert system on Subtask A are impressively high, the modest size of the provided training and testing datasets cannot guarantee the same performance on an arbitrary text input. To ensure the robustness of the presented approaches, more multifaceted research at a larger scale is needed. We see that one of the paths is a community-driven effort towards the increase of available resources and datasets in the Urdu language.

10. Conclusion

This paper presents a shared task in identifying threatening and abusive language in Urdu, namely, the CICLing 2021 track @ FIRE 2021 co-hosted with ODS SoC 2021. For this track, the organizers collected two original datasets of text tweets in Urdu, one annotated for abuse (Subtask A) and the other, for threatening content (Subtask B). We also provided a training and testing split for both datasets, with the ground truth labels hidden from the participants for the testing parts of the datsets. The solutions were submitted in the form of proposed annotations for the testing sets along with the confidence score provided by the participants' systems. The submitted annotations were compared with the ground truth label to compute the F1 score, while the submitted confidence scores served for ROC AUC metric computation. The solutions were ranked by the achieved F1 scores.

In this shared task, twenty one team from six different countries registered for the competition, and seven teams submitted their solutions. Participants used different techniques ranging from the traditional feature-crafting and application of traditional ML algorithms to word representation through pre-trained embeddings to contextual representation and end-to-end transformer based methods. An uncommon solution included an ensemble of traditional ML

classifiers, SVM+LogReg+RF, whereas the particularly successful solutions used specialized BERT-based systems such as multi-lingual BERT and XLM-RoBERTa.

In the abuse detection subtask, team hate-alert outperformed all other systems with m-BERT transformer model achieving F1 score of 0.880. This and the rest of the top 3 results in Subtask A indicate that the specialized transformer based models tend to perform better compared to the feature-based traditional ML models.

In the threat detection subtask, the hate-alert team was also a leader during the official part of the competition with the 0.545 F1 score achieved by the same m-BERT system. However, the results submitted by team SSNCSE_NLP after the official part of the competition was closed showed a much higher F1 score of 0.805. We advert that after the end of the official part of the competition, the ground truth annotations for the testing sets were revealed to public, by this potentially putting the late submitting teams into a more advantageous position compared to the official track participants. Therefore, late submissions were not assigned a rank. Additionally, the technical details of SSNCSE_NLP's solution should be enquired from the corresponding team.

This shared task aims to attract and encourage researchers working in different NLP domains to address the threatening and abusive language detection problem and help to mitigate the proliferation of offensive content on the web. Moreover, this track offers a unique opportunity to fully explore the sufficiency of textual content modality and effectiveness of fusion methods. And last but not least, the annotated datasets in Urdu are provided to the public to encourage further research and improvement of automatic detection of threatening and abusive texts in Urdu.

Acknowledgments

References

[1] U. Naseem, S. K. Khan, M. Farasat, F. Ali, Abusive language detection: a comprehensive review, Indian Journal of Science and Technology 12 (2019) 1–13.

[2] H. Mubarak, K. Darwish, W. Magdy, Abusive language detection on arabic social media, in: Proceedings of the first workshop on abusive language online, 2017, pp. 52–56.

[3] P. Chakraborty, M. H. Seddiqui, Threat and Abusive Language Detection on Social Media in Bengali Language, in: 2019 1st International Conference on Advances in Science, Engineering and Robotics Technology (ICASERT), 2019, pp. 1–6. doi:10.1109/ICASERT.2019.8934609.

[4] N. Ashraf, R. Mustafa, G. Sidorov, A. Gelbukh, Individual vs. group violent threats classification in online discussions, in: Companion Proceedings of the Web Conference 2020, 2020, pp. 629–633.

[5] Z. Waseem, D. Hovy, Hateful symbols or hateful people? predictive features for hate speech detection on twitter, in: Proceedings of the NAACL student research workshop, 2016, pp. 88–93.

[6] Y. Chen, Y. Zhou, S. Zhu, H. Xu, Detecting offensive language in social media to protect adolescent online safety, in: 2012 International Conference on Privacy, Security, Risk and Trust and 2012 International Confernece on Social Computing, IEEE, 2012, pp. 71–80.

[7] C. Van Hee, E. Lefever, B. Verhoeven, J. Mennes, B. Desmet, G. De Pauw, W. Daelemans, V. Hoste, Detection and fine-grained classification of cyberbullying events, in: International conference recent advances in natural language processing (RANLP), 2015, pp. 672–680.

[8] J. Pavlopoulos, P. Malakasiotis, I. Androutsopoulos, Deep learning for user comment moderation, in: Proceedings of the First Workshop on Abusive Language Online, Association for Computational Linguistics, 2017, pp. 25–35.

[9] N. Cecillon, V. Labatut, R. Dufour, G. Linares, Graph embeddings for abusive language detection, SN Computer Science 2 (2021) 1–15.

[10] E. Wulczyn, N. Thain, L. Dixon, Ex machina: Personal attacks seen at scale, in: Proceedings of the 26th international conference on world wide web, 2017, pp. 1391–1399.

[11] T. Ahmed, A. Hautli, Developing a basic lexical resource for Urdu using Hindi WordNet, Proceedings of CLT10, Islamabad, Pakistan (2010).

[12] K. Visweswariah, V. Chenthamarakshan, N. Kambhatla, Urdu and Hindi: Translation and sharing of linguistic resources, in: Coling 2010: Posters, Beijing, China, 2010, pp. 1283–1291.

[13] F. Adeeba, S. Hussain, Experiences in building Urdu WordNet, in: Proceedings of the 9th workshop on Asian language resources, 2011, pp. 31–35.

[14] L. Bertram, Terrorism, the Internet and the Social Media Advantage: Exploring how terrorist organizations exploit aspects of the internet, social media and how these same platforms could be used to counter-violent extremism., Journal for deradicalization (2016) 225–252.

[15] K. Hassan, Social media, media freedom and Pakistan's war on terror, The Round Table 107 (2018) 189–202.

[16] S. T. Aroyehun, A. Gelbukh, Aggression detection in social media: Using deep neural networks, data augmentation, and pseudo labeling, in: Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018), 2018, pp. 90–97.

[17] A. Y. A. R. B. Farhan, A. Noman, R. U. Mustafa, Human aggressiveness and reactions towards uncertain decisions, International Journal of Advanced and Applied Sciences 6 (2019) 112–116.

[18] S. Butt, N. Ashraf, G. Sidorov, A. Gelbukh, Sexism identification using BERT and Data Augmentation–EXIST2021, in: International Conference of the Spanish Society for Natural Language Processing SEPLN, 2021.

[19] N. Djuric, J. Zhou, R. Morris, M. Grbovic, V. Radosavljevic, N. Bhamidipati, Hate speech detection with comment embeddings, in: Proceedings of the 24th international conference on world wide web, 2015, pp. 29–30.

[20] A. Obadimu, E. Mead, M. N. Hussain, N. Agarwal, Identifying Toxicity within YouTube video comment text data, in: International Conference on Social Computing, Behavioral-Cultural Modeling and Prediction and Behavior Representation in Modeling and Simulation,

Springer, 2019, pp. 214–223.

[21] Z. Waseem, T. Davidson, D. Warmsley, I. Weber, Understanding Abuse: A Typology of Abusive Language Detection Subtasks, in: Proceedings of the First Workshop on Abusive Language Online, 2017, pp. 78–84.

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

[23] A. M. Founta, C. Djouvas, D. Chatzakou, I. Leontiadis, J. Blackburn, G. Stringhini, A. Vakali, M. Sirivianos, N. Kourtellis, Large scale crowdsourcing and characterization of Twitter abusive behavior, in: Twelfth International AAAI Conference on Web and Social Media, 2018.

[24] H. L. Hammer, M. A. Riegler, L. Øvrelid, E. Velldal, Threat: A large annotated corpus for detection of violent threats, in: 2019 International Conference on Content-Based Multimedia Indexing (CBMI), 2019, pp. 1–5. doi:10.1109/CBMI.2019.8877435.

[25] N. Ashraf, A. Zubiaga, A. Gelbukh, Abusive Language Detection in YouTube Comments Leveraging Replies as Conversational Context, PeerJ Computer Science (2021).

[26] B. Vidgen, L. Derczynski, Directions in abusive language training data, a systematic review: Garbage in, garbage out, PloS one 15 (2020) e0243300.

[27] P. Nakov, V. Nayak, K. Dent, A. Bhatawdekar, S. M. Sarwar, M. Hardalov, Y. Dinkov, D. Zlatkova, G. Bouchard, I. Augenstein, Detecting abusive language on online platforms: A critical analysis, arXiv preprint arXiv:2103.00153 (2021).

[28] M. Zampieri, P. Nakov, S. Rosenthal, P. Atanasova, G. Karadzhov, H. Mubarak, L. Derczynski, Z. Pitenis, Ç. Çöltekin, SemEval-2020 Task 12: Multilingual Offensive Language Identification in Social Media, (OffensEval), International Committee for Computational Linguistics (2020) 1425–1447.

[29] M. Zampieri, S. Malmasi, P. Nakov, S. Rosenthal, N. Farra, R. Kumar, SemEval-2019 Task 6: Identifying and Categorizing Offensive Language in Social Media (OffensEval), Association for Computational Linguistics (2019) 75–86.

[30] M. Wiegand, M. Siegel, J. Ruppenhofer, Overview of the GermEval 2018 Shared Task on the Identification of Offensive Language (2018).

[31] P. Fortuna, J. Ferreira, L. Pires, G. Routar, S. Nunes, Merging datasets for aggressive text identification, in: Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018), 2018, pp. 128–139.

[32] V. Basile, C. Bosco, E. Fersini, N. Debora, V. Patti, F. M. R. Pardo, P. Rosso, M. Sanguinetti, et al., Multilingual detection of hate speech against immigrants and women in Twitter at SemEval-2019 task 5: Frequency analysis interpolation for hate in speech detection, in: 13th International Workshop on Semantic Evaluation, Association for Computational Linguistics, 2019, pp. 54–63.

[33] T. Mandl, S. Modha, P. Majumder, D. Patel, M. Dave, C. Mandlia, A. Patel, Overview of the HASOC track at FIRE 2019: Hate Speech and Offensive Content Identification in Indo-European Languages, in: Proceedings of the 11th forum for information retrieval evaluation, 2019, pp. 14–17.

[34] T. Mandl, S. Modha, A. Kumar M, B. R. Chakravarthi, Overview of the HASOC Track at FIRE 2020: Hate Speech and Offensive Language Identification in Tamil, Malayalam,

Hindi, English and German, in: Forum for Information Retrieval Evaluation, Association for Computing Machinery, 2020, pp. 29–32.

[35] D. Yin, Z. Xue, L. Hong, B. D. Davison, A. Kontostathis, L. Edwards, Detection of harassment on Web 2.0, in: Proceedings of the Content Analysis in the WEB, volume 2, 2009, pp. 1–7.

[36] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, BERT: Pre-training of deep bidirectional transformers for language understanding (2019) 4171–4186.

[37] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, et al., Language models are unsupervised multitask learners, OpenAI blog 1 (2019) 9.

[38] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, R. Soricut, ALBERT: A Lite BERT for Self-supervised Learning of Language Representations (2020).

[39] N. Vashistha, A. Zubiaga, Online multilingual hate speech detection: experimenting with Hindi and English social media, Information 12 (2021) 5.

[40] M. Amjad, N. Ashraf, A. Zhila, G. Sidorov, A. Zubiaga, A. Gelbukh, Threatening language detection and target identification in Urdu tweets, IEEE Access 9 (2021) 128302–128313.

[41] M. Amjad, N. Ashraf, A. Zhila, G. Sidorov, L. Chanona-Hernandez, A. Gelbukh, Automatic abusive language detection in urdu tweets, Acta Polytechnica Hungarica (2021) 1785–8860.

[42] S. Bird, E. Loper, NLTK: The natural language toolkit, in: Proceedings of the ACL Interactive Poster and Demonstration Sessions, Association for Computational Linguistics, Barcelona, Spain, 2004, pp. 214–217. URL: https://aclanthology.org/P04-3031.

[43] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, Scikit-learn: Machine learning in Python, Journal of Machine Learning Research 12 (2011) 2825–2830.

[44] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, S. Chintala, Pytorch: An imperative style, high-performance deep learning library, in: H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, R. Garnett (Eds.), Advances in Neural Information Processing Systems 32, Curran Associates, Inc., 2019, pp. 8024–8035.

[45] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. L. Scao, S. Gugger, M. Drame, Q. Lhoest, A. M. Rush, Transformers: State-of-the-art natural language processing, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, Association for Computational Linguistics, Online, 2020, pp. 38–45. URL: https://www.aclweb.org/anthology/2020.emnlp-demos.6.

[46] 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). URL: http://arxiv.org/abs/1907.11692.