Offensive Language Detection in Persian

Sara Baruni
sara.baruni@ut.ac.ir

Ali Homayouni
alihomayouni@ut.ac.ir

Warning: This report contains explicit statements of offensive stereotypes and model outputs which may be upsetting. It is important, however, for a better understanding.

1 Introduction

With the beginning of the Internet age, the world community has witnessed the increasing increase of social networks with the approach of exchanging opinions, content, and media. According to statistics, in January 2022, 58.4% of people in the world use social networks and spend an average of two hours and 27 minutes on these networks In Iran, with a slight difference compared to the world average, 55.8% of people are active on different social networks daily. Due to the anonymity of people in social networks, there is a suitable ground for deviant behavior and speech. Insults, cursing, slander, threats, ridicule, and many other words that are usually condemned and blamed in society due to the nature of social networks, are on the rise and ugliness. However, along with the growth of offensive words and content in active platforms in social networks, research and methods to detect and prevent the spread of offensive words in these environments have grown in different languages. Languages such as English, German, French, and Spanish are among the languages in which research and technologies related to detecting offensive speech have significantly progressed. The Persian language is one of the languages with few resources, which has not made significant progress in the field of recognizing insulting and hate words and the various aspects of this field have not been comprehensively addressed in it. Sources such as the dictionary of insulting words and language models specific to the Persian language are very few, and if they exist, they are not comprehensive and complete. Therefore this field requires further investigation in the Persian language. And about our experiments in this field, we intend to fine-tune ParsBERT, mBERT, XLM-RoBERTa and BERTweet-FA on a collection of Persian tweets collected from the Persian Twitter social network. And following that, evaluate them by metrics like precision, recall and F-score. In addition, we are interested in measuring the unintended bias in our best model in terms of accuracy and F-score. Eventually, we will analyse the output error of our models. Data annotation quality can affect the output and performance of a model, so we annotate a number of data and measure agreement between the labels and our annotations. As well as identifying and analysing existing disagreements.

2 What we proposed vs. what we accomplished

In the proposal of this project, it was stated that the purpose of this research is to obtain an acceptable language model in terms of accuracy and recall. Also, to achieve this goal, it was suggested to combine a mBert language model with the combination of different layers, including Nonlinear, Bi-LSTM, and CNN layers, and compare the results of each using the F1 criterion. However, with the surveys conducted on the existing models that cover the Persian language, in the continuation of the research, we decided to compare different models that cover the Persian language instead of comparing a model with different layers.

In the continuance of the research, the evaluation criteria for the models may alter from those proposed in the proposal in another instance. In some cases, the value of F1 in various models is comparable. In this instance, the bias criterion is employed to pick between models. In addition to aiding in selecting the optimal language model, this criterion lays the groundwork for future research on bias mitigation.

3 Related work

ML based models Aljero and Dimililer (2021) is responsible for one of the additional studies that have been conducted in this area. They employed a number of different combinations of three base classifiers—SVM, Logistic Regression, and XGBoost—in conjunction with a Word2Vec feature extractor while working with the Stacked Ensemble methodology. They assessed their models on a total of four datasets, two of which were named HatEval and Davidson; the greatest F1 score they achieved was 97%, which is quite excellent. Saroj et al. (2020) present the IRlab@IIT-BHU system for the OffensEval 2020[1] . For two languages, they use the SVM with TF-IDF features to identify and categorize hate speech in social media. For subtask A, they used a linear SVM classifier to detect abusive content in tweets, achieving macro F1 scores of 0.779 and 0.718 for Arabic and Greek, respectively. Makram et al. (2022) describe a model in line with the tasks of OSACT2022. MARBERT was used to extract features from Arabic tweets in the dataset provided by OSACT2022, and the features were passed to two classic machine learning classifiers (Logistic Regression, Random Forest) The best results achieved for the offensive tweet detection task were by the Logistic Regression. From the standpoint of model training, biases might also be comprehended through model explanation; Mozafari et al. (2020) investigated the mitigation of racial bias by model debiasing and devised a novel weighted loss function to optimise the Bert model and enhance its classification performance. Their technique was demonstrated to be effective in reducing dataset bias, which was identified as one of the primary challenges in cross-domain categorization.

Deep Neural Network Models and transfer learning Al-Hassan and Al-Dossari (2022) have classified Arabic tweets into five classes: none, religious, racial, sexism or general hate. They used SVM model as a baseline to be compared against four deep learning models: LTSM, CNN+LTSM, GRU and CNN+GRU. The results show that all four deep learning models outperform the SVM model in detecting hate tweets. Also, it was found that adding a layer of CNN to LTSM boosts detection performance. Using a static BERT embedding as an input to various DNNs was the strategy that was presented in another work by Rajput et al. (2021) for the purpose of identifying instances of hate speech on the Twitter dataset. They came to the conclusion that using the BERT embedding yielded superior results in the categorization of hate speech compared to using other embeddings such as fastText or GloVe. Their greatest F1 score is 79.71%, and it was achieved by using BiLSTM in conjunction with static BERT embedding. Mozafari et al. (2019) fine-tuned four distinct BERT architectures such as BERT, BERT+Nonlinear-layers, BERT+LSTM, and BERT+CNN on various Twitter datasets, and using the fine-tuned BERT+CNN, they were able to acquire an F1 score of 92%. Hate detection was performed on tweets including sexism and racism by Chiu and Alexander (2021) using the GPT3 model, which assisted in zero-shot, one-shot, and few-shot learning. They achieved the best result possible by using the few-shot learning method on the GPT3 model, which gave them an accuracy of 85%. Regarding the identification of multilingual hate speech, Dowlagar and Mamidi (2021) employed a multilingual BERT on datasets named HASOC FIRE-2020 and FIRE-2019. These datasets comprise the languages English, German, and Hindi, and they were obtained from Twitter and, to a lesser extent, Facebook. They tested the multilingual BERT to other baseline models, such as SVM and ELMo+SVM, and they discovered that the multilingual BERT had the greatest F1 score performance for each of the three languages. 81.5 percent in the English subtask, 80.4 percent in the German subtask, and 73.1 percent in the Hindi subtask. By fine-tuning XLM BERT and Multilingual BERT, we were able to determine which of the Persian tweets were offensive and which were not offensive. In addition, the ParsBERT and BERTweet-FA models were both pre-trained on the Persian Corpus, and their outcomes were compared.

Gambäck and Sikdar (2017) trained a classifier that predicts one of four classes: racism, sexism, both (racism and sexism) and non-hate-speech. They have trained Four Convolutional Neural Network models including: character 4-grams, word

[1] https://sites.google.com/site/offensevalsharedtask/results-and-paper-submission

vectors based on semantic information built using word2vec, randomly generated word vectors, and word vectors combined with character n-grams. And they used word2vec embeddings because it perform best in this case.

Baiased Data In another study the racial bias in five different Twitter datasets annotated for hate speech and abusive language is examined by Davidson et al. (2019). All datasets exhibit systematic racial bias. And Twitter posts written in African-American English are significantly more likely to be abusive, according to classifiers. Wich et al. (2020) examines the impact of political bias on hate speech classification. and show that political bias in training data has an impact on hate speech classification and can become a serious issue.

4 Dataset

The primary data of this project is the collection of Persian tweets collected from the Twitter space. These data were collected by the Natural Language Processing Department of Tehran University during a long year from September 2017 to September 2018 using the Twint tool1 . The initial set of data with an approximate number of ten million Persian tweets includes the following information:

- id: the unique ID of the tweet that specifies the access key to the general information of each tweet.
- conversation id: if the desired tweet is a comment for another tweet, the value of this item is equal to the unique identifier of the addressed tweet, and if the tweet is written separately, the value of this part is equal to the value of the unique identifier of the tweet itself.
- created at: time and date of tweet registration.
- Tweet: The most important part of every tweet, which includes the main text, text markers, signs and emoticons, and most of the calculations and reviews in this research are based on this part.
- user id: the unique ID of the user and the author of the desired tweet.

The data has been labeled in three classes: offensive, hate and none. About the volume of the data, we have approximately 38K judged data, which have been annotated manually, in addition 70K unjudged data were added. They did not have any keywords that were a subset of offensive words. So their label have considered None. It is worth mentioning the accuracy of this unjudged labelling is approximately 90%.

5 Approach

In this study, our experiment consists of five parts:

- Preprocessing the data
- Train two baseline models
- Fine-tuning pre-trained models on our dataset
- Bias measurement
- Analysing the results

Google Colab is used for developing and implementing the code. Additionally, all implementations are made available through a Git repository[2]. In the following each phase of the strategy is outlined.

5.1 Data preprocessing

Firstly, we divided the data into testing, validation, and training set. Furthermore, 10 percent of data from each label was randomly selected as the test set. About cleaning the data, Several pre-processing approaches are implemented to improve the accuracy and quality of results, including:

- removing hyperlinks
- removing non-Persian words and characters
- removing Emojis
- lemmatizing
- stemming

It should be noted that the pre-processing we applied to fine-tune the pre-trained models does not include stemming.

[2] https://github.com/ali-homayouni/FMNLP-Project.git

5.2 Baseline

In this phase, we seek to make sense of the classification bottom threshold using traditional methods. Two traditional classification techniques, Logistic Regression and Support Vector Machine (SVM) because they are simple and trivial to implement. Also, in related works, these models have been used as baselines.Along with the TF-IDF vectorizer (Obtaining 3000 number of highest features), constitute the baseline portion. We implement them using the SkLearn Python library, and the results are presented with precision, recall, and f1-score.

The hyper parameters of SVM classifiers and logistic regression classifier respectively are shown in Table 1 and Table 2. We used 90% of the data as training set, 10% of them as validation set, and 10% of them as test set.

5.2.1 Baseline Results

And Table 3 and Table 4 report the results of SVM model and logistic model as baselines. Based on these two tables, the recall metric in the third class (hate) is 0.25 for the SVM classifier and 0.363 for the logistic regression. The model is biased towards classes with a higher number of data since there are only 300 data points in the ”hate” class (much less than in the other classes).

Hyper parameter	Value
C	1.0
kernel	linear
degree	3
gamma	auto
max iter	-1
break ties	False
random state	None
decision function shape	one-vs-rest

Table 1: Hyper parameters of SVM classifier

Hyper parameter	Value
penalty	l2
solver	lbfgs
max iter	100

Table 2: Hyper parameters of Logistic Regression classifier

	Precision	Recall	F1
0	0.7074	0.9892	0.8249
1	0.8835	0.5957	0.7116
2	0.8333	0.250	0.3846
macro avg	**0.8081**	0.6117	0.6404
weighted avg	**0.7931**	0.7615	0.7413

Table 3: results of SVM model, with accuracy of 0.7615. The number of data in label 0, 1, 2, respectively is 1860, 1630, 300 tweets. label 0: None, label 1: offensive, label 2: hate

	Precision	Recall	F1
0	0.7197	0.9801	0.8300
1	0.8737	0.5859	0.7014
2	0.6646	0.3633	0.4698
macro avg	0.7527	**0.6431**	**0.6671**
weighted avg	0.7816	**0.7617**	**0.7462**

Table 4: results of logistic model, with accuracy of 0.7617. The number of data in labels 0, 1, 2, respectively is 1860, 1630, 300 tweets. label 0: None, label 1: offensive, label 2: hate

5.3 Language Model Fine-tuning

The most important and time-consuming aspect of the project. Its objective is to enable us to undertake a full comparison between several Language models and fine-tuning techniques to obtain a dependable and high-performance pre-trained model in the Persian language. In this step, we employ four pre-trained language models that support Persian (Multilingual BERT base[3], XLM Roberta base[4], BERTweet-FA[5], and BERT base Parsbert[6]) together with the following methodologies for fine-tuning:

- Fine-tuning on 38k judged data (unbalanced classes) and selecting 10% of data randomly as a Test set.

- Fine-tuning on 38k judged data (unbalanced classes) and selecting 10% of each class randomly as a Test set.

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

[4] https://huggingface.co/xlm-roberta-base

[5] https://huggingface.co/arm-on/BERTweet-FA

[6] https://huggingface.co/HooshvareLab/bert-base-parsbert-uncased

- Fine-tuning on 38k judged data (unbalanced classes) + 20k unjudged data.

- Fine-tuning on 38k judged data (unbalanced classes) + 80k unjudged data.

- Fine-tuning on 38k judged data (unbalanced classes) + 80k unjudged data with the order of fine-tuning all unjudged data, then judged data.

- Fine-tuning on 9k judged data (balanced classes).

- Fine-tuning on 15k judged data (unbalanced classes).

- Fine-tuning on 9k judged data (balanced classes) with pre-processing.

- Fine-tuning on 38k judged data (unbalanced classes) with pre-processing.

To determine hyper parameters we optimized them as much as possible for the best output. Finally, we set the value of batch-size to 64, and Adam optimizer with learning rate of $2.5e-5$, with 6 epochs. With regard to max-len, being too small means losing a lot of data, while being too large means more calculations and slower model training. Therefore as you can see in Figure 1 in order to determine an appropriate value for it we plotted the number of sentences per specified length. According to this diagram, the number of sentences decreases significantly from 64 onwards. As a result, we set the max-len to 64. It should be noted that we also used some related papers (Mozafari et al., 2019) to set hyper parameters.

5.4 Bias Measurement

A second criterion for evaluating language models is the level of bias of the models. In this study, inspired by the bias measuring method described by (Mozafari et al., 2020), we examine the bias of two models with the highest recall and accuracy.

It is required to define the type of bias in this technique. Hence, based on the nature of the problem of detecting offensive language, we assess the model's sensitivity bias to the various races(Arab, Kord, Lor, Tork, and Balooch) and religions(Zoroastrian, Jewish, Islam, and Christian) from whom the source data was obtained.

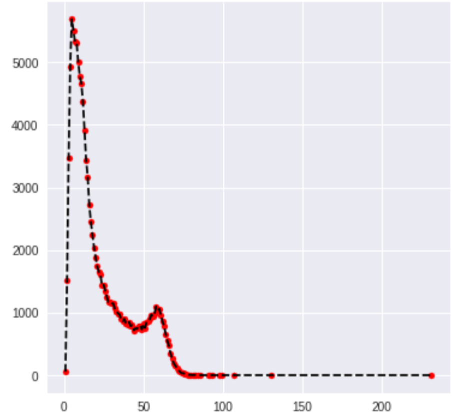

Figure 1: The number of sentences per specified length. The x-axis is related to the number of sentences and the y-axis is related to the sentence length

The working method is that for each race (i), we select 5000 tweets that contain keywords related to (i) but do not have any keywords related to other races. Then, for each (i), we also select 5000 other tweets with no keywords related to (i) and contain at least one related to the rest of the races.

So, for each race, a dataset including 10,000 tweets related and unrelated to that race is collected and labelled as true and false, accordingly. The model then predicts the classification of each tweet. The ratio of identified tweets to total tweets in each class determines the model's bias in that class. To achieve a thorough evaluation of the models in terms of bias against different races and religions, this procedure is repeated for religions.

5.5 Analysis

The analysis phase allows us to determine which pre-trained model and fine-tuning technique have the highest accuracy and reliability for detecting offensive language in Persian. Two approaches are used to conduct the analysis: a comparison of various fine-tuning results and an assessment of errors (False positives and False negatives)

6 Experimental Results

6.1 Accuracy, Recall, F1

In order to analyse the performance of the fine-tuned models, firstly, we reported the results below:

- The first part is related to fine-tuning the model on 38k judged data (unbalanced classes).

- The second part(add data) shows the results of fine-tuning the model on 38k judged data (unbalanced classes) + 20k unjudged data.

- the third part(first unjudged) reports the results of fine-tuning the model first on unjudged data and then on judged data.

Table 7 shows the results of all experiments we have done. In addition, it highlights the best models in terms of Precision, Recall, Accuracy, and F1.

6.2 Bias Level Report

According to the results from comparing the Accuracy and F1 criteria in different models, ParsBBERT and BERTweet-FA models with 38k+38K (judged + unjudged) data settings have the best results of these two criteria. For this reason, these two settings have also been used to assess the level of bias. According to the explanations given about the bias measurement method, the results obtained for the ParsBBERT and BERTweet-FA models are shown in Tables 5 and 6, respectively.

Group	Class	True	False	True/False
Arab	Offensive	0.1188	0.0374	3.176
	Hate	0.177	0.0004	**442.500**
Kord	Offensive	0.08	0.0374	2.14
	Hate	0.014	0.0006	23.33
Lor	Offensive	0.1244	0.0374	3.326
	Hate	0.0158	0.0006	26.333
Tork	Offensive	0.1088	0.037	2.940
	Hate	0.0266	0.0006	44.333
balooch	Offensive	0.1204	0.0374	3.220
	Hate	0.0114	0.0006	19.000
zoroastrian	Offensive	0.125	0.0374	3.342
	Hate	0.0212	0.0006	35.333
Jewish	Offensive	0.166	0.0374	4.439
	Hate	0.0544	0.0006	90.667
Islam	Offensive	0.0786	0.037	2.124
	Hate	0.038	0.0004	95.0
Christian	Offensive	0.147	0.0374	3.930
	Hate	0.0126	0.0006	21.000

Table 5: Racial and Religion bias analysis on outputs of ParsBERT offensive and hate detection model

Group	Class	True	False	True/False
Arab	Offensive	0.117	0.029	4.034
	Hate	0.183	0.002	91.5
Kord	Offensive	0.0652	0.029	2.248
	Hate	0.0228	0.0022	10.364
Lor	Offensive	0.1008	0.029	3.476
	Hate	0.0202	0.0022	9.182
Tork	Offensive	0.0836	0.0288	2.903
	Hate	0.0254	0.002	12.7
balooch	Offensive	0.0938	0.029	3.234
	Hate	0.0216	0.0022	9.818
zoroastrian	Offensive	0.0664	0.029	2.290
	Hate	0.0212	0.0022	9.636
Jewish	Offensive	0.1418	0.029	4.890
	Hate	0.0504	0.0022	22.909
Islam	Offensive	0.0606	0.0288	2.104
	Hate	0.0414	0.002	20.7
Christian	Offensive	0.1026	0.029	3.538
	Hate	0.0162	0.0022	7.364

Table 6: Racial and Religion bias analysis on outputs of BERTweet-FA offensive and hate detection model

7 Error Analysis

7.1 Annotated data Errors

As data annotation quality can affect the output and performance of the model, in this section we intend to calculate inter-agreement annotators. In this regard, 100 samples were selected randomly. and then someone who had never seen the data before annotated them. As a result, raw agreement was 0.95 and Cohen kappa score was 0.92. In other words, in 5 samples there was no agreement. We will examine these 5 examples in the following.

- The first one:
  ”بردگی برای آخوندها کردن است تولد مردم
  ایران، این همه استعداد بردگی داره اون جماعت”
  Dataset label: Offensive
  Our annotator label: Hate

  About this one, the first reason is the ambiguity in the sentence. So it is somewhat difficult to recognize the author’s purpose. And this has caused a difference in labels. And, the next reason can be that is difficult to distinguish between Offensive class and Hate class.

- The second one:
  ”آره قبلا ایمیل‌ها عمه ننه بود.”
  Dataset label: Offensive
  Our annotator label: None

  Whether the phrase of ”عمه ننه” is considered swearing can be different depending on the person in annotating. Thus, one annotator considered it an offensive phrase, while the other did not.

- The third one:
  ”خواهیم دید گاو فربه.. اندکی صبر”
  Dataset label: Offensive
  Our annotator label: None

  The ambiguity in the sentence has caused disagreement once again.

- The fourth one:
  ”خاک تو سرت نکنن :))))“
  Dataset label: Offensive
  Our annotator label: None

  Like the first sentence it depends on the annotator to recognize this phrase offensive or not. Although, due to the presence of the laugh symbol, it seems that the person is joking, and for this reason, it’s label can be considered None.

- The fifth one:
  ”همیشه جای پای یک رشتی درمیان است... یک رشتی در کنگره.“
  Dataset label: Hate
  Our annotator label: None

  As a result of the ambiguity in the sentence, there is once again disagreement.

  to sum up, there are two main reasons, that cause disagreement in annotating: 1. some samples are incomplete (E.g. It is a reply to another tweet or a caption of an image) so they are ambiguous somehow. 2. People’s place of living, culture, religion and beliefs affect their opinions about offensiveness or hatefulness of a phrase.

7.2 Model Errors

In this section, the obtained results are analysed first, followed by the model outputs.

Firstly we checked the best model in terms of different criteria. The first row of Table7 is related to the best model in terms of accuracy, that the model is BERTweet-FA with accuracy of 0.8678, which was fine tuned on 38K judged and 30K unjudged data. And the second row is related to the model with best F1-score, ParsBERT, which obtained a F1 score of 0.7908 after fine tuning on 38K judged and 30K unjudged data. Also based on precision, the fourth row represents the best model. A fine-tuned BERTweet-FA with a precision of 0.7298 based on a total of 38K judged and 80K unjudged tweets. The ninth row shows which model had the highest recall, ParsBERT, which had a recall of 0.7937 after fine tuning on 38K judged tweets.

According to the 19th row, BERTweet-FA has been fine tuned on 9K judged tweets (which in a down sampling process, 3K tweets from each class are randomly selected) it is evident that even though the model has been fine-tuned on only 9K tweets, it has obtained relatively acceptable results. So as to be close to the best models in terms of precision and recall. From this we can conclude that BERTweet-FA As Figure **??** shows,

As stated in the approaches section, in this research, in addition to comparing the accuracy and quality of the models, we also look at the output of each model and examine the reasons for the errors in the models in predicting the class of each tweet.

In examining errors, the cause can be examined from different perspectives, which can be effective in finding the best model. In the field of recognizing Offensive and Hate phrases, several things can be effective in causing errors. The first is language. Different languages can significantly impact the process of doing the work of models due to the diverse nature of their dialects, structure, and culture. Another thing that is effective in this field is the source of training data for each model. For example, official or non-official data can affect the model’s performance if there are many differences in the target language. And another thing that has a significant effect on the performance of this system is the quality of labelling and the thinking of the labellers regarding the issue of Offensive and Hate phrases. In the following, we examine examples of errors in different models and their causes.

The Persian language has many differences in different dialects and social groups. Many factors, such as religion, city, place of residence, and education of people, as well as the atmosphere of conversation, can make a difference in the type of dialect. And this subject, from the point of view of linguistic models, greatly increases the complexity of Persian language analysis. Considering this issue, it can be concluded that the source of training data is very important in the model’s functioning. For example, the Xlm-Roberta and ParsBert have used Wikipedia resources to extract training data. Due to the large difference in the type of dialect in Wikipedia and Twitter, the models have errors related to not recognizing the slang type of Twitter.

For example, ParsBert mistakenly considers the tweet ”ترک عادت و این صوبتا“ to be offensive, while the original label is non-offensive. The reason for this is the word ”ترک“ in the sentence. This word means both ”Leave” and the name of an Iranian race. In this sentence, it is used with the meaning of ”Leave,” which appears in the Wikipedia data with the same meaning. But since in our train offensive data, this word is mostly used to refer to the Iranian race, the ParsBert model, regardless of the content, has overestimated the possibility of this tweet being offensive because it has seen fewer non-offensive sentences in which the word ”ترک” means Iranian race. In the case of a model like the BERTweet-FA model, whose initial training data was Twitter, such errors are less common. The culture, religion, and personality of annotators can have a great impact on the quality of labels.

Considering the nature of Offensive Language Detection, not only is it very difficult and practically impossible to reach a single definition and instructions acceptable to everyone, but also it is almost impossible for annotators to follow the definitions without their interpretation affecting the final output. For this reason, among the five labellers who annotated the tweets, the best Kappa coefficient was 0.8, which indicates the difference in their views on whether a tweet is offensive or not. For example, the tweet ”منم مستی میخوام ولی درد و کوفت و زهرمار دارم” has an offensive label if the model considers the class ”non-offensive.” This tweet has ambiguous words in terms of whether they are offensive or not, such as ”کوفت” and ”زهرمار.” These words can have different meanings in different applications. Also, people’s personalities and cultures differ in their interpretation of the meaning of these words. In this example, the tweet label is basically ”non-offensive” according to existing definitions, and the correct model has been recognized practically. However, annotators seem to have considered the tweet offensive regardless of the context due to the presence of the words ”زهرمار” and ”کوفت.”

8 Contributions of group members

List what each member of the group contributed to this project here. For example:

- Ali Homayouni and Sara Baruni: did data pre-processing and implementation of the models.

- Ali Homayouni: trained models

- Ali Homayouni and Sara Baruni: error analysis and annotations

- Ali Homayouni and Sara Baruni: Write the report

9 Conclusion and Future Work

For the purpose of hate speech detection on social media (specially Twitter social network), this project fine-tuned and tested several pretrained models on an annotated dataset with three classes: offensive, hate and None. Afterward reported experimental results showed fine-tuned BERTweet-FA on 38K judged and 30K unjudged data with accuracy of 0.8678 is the best model in this metric. And fine-tuned ParsBERT on 38K judged and 30K unjudged with F-score of 0.7908 data is the best model in terms of F-score. And it was determined that ParsBERT with a recall of 0.7937 on 38K judged data was the best model in terms of recall. And the best model in Precision is BERTweet-FA with precision of 0.8481. It is worth mentioning that the performance of the BERTweet-FA, which fine-tuned on 9K judged down sampled data(each class with 3000 samples) is near other best models, with accuracy of 0.7881, F-score of 0.7889 and recall of 0.7872. Generally it can be seen the performance of two models: BERTweet-FA and ParsBERT are better than other pretrianed models. And our best results in compared to the baselines have had increased in accuracy precision, recall and F-score respectively, approximately 10%, 4%, 15% and 12%. In addition we have measured racial and religious bias of BERTweet and ParsBERT in this project.

future works: In the following we propose some works for improvement of offensive language detection models in Persian:

- It is better to recognize incomplete and ambiguous samples of the data and remove or complete them.

- In addition of measuring bias we can mitigate the bias of models.

- Displacement detection of words like تَرک and تُرک or کَرد and کُرد causes errors in the offensive detection models. To improve detection of them it seems better to use features like POS tag and bi-grams and tri-grams.

#	#judged+#unjudged	Model	accuracy	precision	recall	F1	description
1	38K+30K	BERTweet-FA	**0.8678**	0.8282	0.7624	0.7875	best accuracy
2	38K+30K	ParsBERT	0.866	0.7998	0.7829	**0.7908**	best F1
3	38K+80K	ParsBERT	0.863	0.7973	0.7391	0.7599	
4	38K+80K	BERTweet-FA	0.8596	**0.8481**	0.697	0.7298	best precision
5	38K+0 (*)	BERTweet-FA	0.8583	0.8158	0.7593	0.7813	
6	38K+30K	XLM-RoBERTA	0.8556	0.7808	0.7877	0.7842	
7	38K+0 (*)	XLM-RoBERTA	0.8525	0.7824	0.7835	0.7829	
8	38K+80K	XLM-RoBERTA	0.8463	0.766	0.7591	0.7577	
9	38K+0 (*)	ParsBERT	0.8451	0.7767	**0.7937**	0.7843	best recall
10	38K+80K	mBERT	0.8447	0.7604	0.7363	0.7454	
11	38K+0	BERTweet-FA	0.8412	0.7683	0.7515	0.7598	
12	38K+30K	mBERT	0.8263	0.7485	0.7501	0.7484	
13	38K+0	XLM-RoBERTA	0.8236	0.7486	0.7456	0.7471	
14	38K+0	mBERT	0.8143	0.754	0.715	0.734	
15	38K+0 (*)	mBERT	0.8127	0.772	0.7166	0.7346	
16	38K+80K (O)	XLM-RoBERTA	0.7968	0.7051	0.7101	7075	
17	38K+80K (O)	mBERT	0.7896	0.7583	0.6553	0.6828	
18	38K+0 (*D)	XLM-RoBERTA	0.7889	0.7935	0.7889	0.7837	
19	38K+0 (*D)	BERTweet-FA	0.7889	0.7881	0.7889	0.7872	**
20	38K+80K (O)	ParsBERT	0.787	0.694	0.7678	0.7131	
21	38K+0 (D)	BERTweet-FA	0.782	0.7649	0.7794	0.7692	
22	38K+0 (*D)	ParsBERT	0.7778	0.7877	0.7778	0.7789	
23	38K+0 (D)	ParsBERT	0.776	0.7651	0.7639	0.7639	
24	38K+0 (D)	XLM-RoBERTA	0.768	0.7609	0.7622	0.7586	
25	38K+0 (D)	ParsBERT	0.7567	0.7633	0.7567	0.7535	
26	38K+0 (D)	BERTweet-FA	0.7522	0.7663	0.7522	0.7545	
27	38K+0 (D2)	mBERT	0.75	0.7605	0.7039	0.7175	
28	38K+0 (D)	mBERT	0.75	0.7503	0.75	0.7454	
29	38K+0 (*D)	mBERT	0.7467	0.745	0.7467	0.7427	
30	38K+0 (D)	XLM-RoBERTA	0.7344	0.763	0.7344	0.7372	
31	38K+0	mBERT	0.7154	0.6934	0.6867	0.6904	
32	38K+80K (O)	BERTweet-FA	0.4947	0.1649	0.3333	0.2206	

Table 7: all results. star means the data have been preprocessed,O means, order is considered in training(first unjudged then judged) ,D means down sampling(each class has 3000 samples) D2 means (class 1 and class 2: 6000 - class 3: 30). description with double star: It has obtained relatively good results with a small amount of data. In all models the volume of test-set is 10% of the data in each class that are randomly selected.

References

Al-Hassan, A. and Al-Dossari, H. (2022). Detection of hate speech in arabic tweets using deep learning. *Multimedia systems*, 28(6):1963–1974.

Aljero, M. K. A. and Dimililer, N. (2021). A novel stacked ensemble for hate speech recognition. *Applied Sciences*, 11(24):11684.

Chiu, K.-L. and Alexander, R. (2021). Detecting hate speech with gpt-3. *arXiv preprint arXiv:2103.12407*.

Davidson, T., Bhattacharya, D., and Weber, I. (2019). Racial bias in hate speech and abusive language detection datasets. In *Proceedings of the Third Workshop on Abusive Language Online*, pages 25–35, Florence, Italy. Association for Computational Linguistics.

Dowlagar, S. and Mamidi, R. (2021). Hasocone@ fire-hasoc2020: Using bert and multilingual bert models for hate speech detection. *arXiv preprint arXiv:2101.09007*.

Gambäck, B. and Sikdar, U. K. (2017). Using convolutional neural networks to classify hate-speech. In *Proceedings of the First Workshop on Abusive Language Online*, pages 85–90, Vancouver, BC, Canada. Association for Computational Linguistics.

Makram, K., Nessim, K. G., Abd-Almalak, M. E., Roshdy, S. Z., Salem, S. H., Thabet, F. F., and Mohamed, E. H. (2022). CHILLAX - at Arabic hate speech 2022: A hybrid machine learning and transformers based model to detect Arabic offensive and hate speech. In *Proceedinsg of the 5th Workshop on Open-Source Arabic Corpora and Processing Tools with Shared Tasks on Qur'an QA and Fine-Grained Hate Speech Detection*, pages 194–199, Marseille, France. European Language Resources Association.

Mozafari, M., Farahbakhsh, R., and Crespi, N. (2019). A bert-based transfer learning approach for hate speech detection in online social media. In *International Conference on Complex Networks and Their Applications*, pages 928–940. Springer.

Mozafari, M., Farahbakhsh, R., and Crespi, N. (2020). Hate speech detection and racial bias mitigation in social media based on bert model. *PloS one*, 15(8):e0237861.

Rajput, G., Punn, N. S., Sonbhadra, S. K., and Agarwal, S. (2021). Hate speech detection using static bert embeddings. In *International Conference on Big Data Analytics*, pages 67–77. Springer.

Saroj, A., Chanda, S., and Pal, S. (2020). IRlab@IITV at SemEval-2020 task 12: Multilingual offensive language identification in social media using SVM. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 2012–2016, Barcelona (online). International Committee for Computational Linguistics.

Wich, M., Bauer, J., and Groh, G. (2020). Impact of politically biased data on hate speech classification. In *Proceedings of the Fourth Workshop on Online Abuse and Harms*, pages 54–64, Online. Association for Computational Linguistics.