Linguistic-based Data Augmentation Approach for Offensive Language Detection

Toygar Tanyel
Department of Computer Engineering
Yildiz Technical University
Istanbul, Türkiye
toygar.tanyel@std.yildiz.edu.tr

Besher Alkurdi
Department of Computer Engineering
Yildiz Technical University
Istanbul, Türkiye
besher.alkurdi@std.yildiz.edu.tr

Serkan Ayvaz
Department of Computer Engineering
Yildiz Technical University
Istanbul, Türkiye
sayvaz@yildiz.edu.tr

2022 7th International Conference on Computer Science and Engineering (UBMK) | 978-1-6654-7010-0/22/$31.00 ©2022 IEEE | DOI: 10.1109/UBMK55850.2022.9919562

***Abstract*—The massive amount of data generated by social media possess a great deal of toxic content that lead to serious content filtering problems including hate speech, cyberbullying and insulting. Offensive content even without profanity may result in psychological and physical harms to, particularly children and sensitive people. As of 2022, Turkey houses 7th largest Twitter community among all countries in terms of the active user size exceeding 16 million users, which represents a high diversity of people considering its population. That said, there is a growing need for a comprehensive and high-quality dataset in Turkish that can be utilized in development of NLP models for robust detection of offensive language usage in social media. Related studies in literature have mostly focused on small, synthetic and label-imbalanced datasets. Machine learning models trained on such datasets tend to favor majority class for accuracy and possess generalizability issues. However, it is challenging to create an unbiased dataset containing hate speech without offensive words, and build an accurate detection model to identify the actual hate speech Tweets. The models may lack sufficient context due to the absence of swear words. Therefore, we propose a data augmentation approach based on data mining methods utilizing the linguistic features of Turkish that can help enhance the generalizability of machine learning models without further human interaction. Furthermore, we evaluated the impact of our comprehensive dataset in detection of offensive language in social media. The NLP models training using the augmented dataset improved the macro average detection accuracy by 7.60% in comparison to the baseline approach.**

***Keywords* —Social media, offensive language, hate speech, cyberbullying, contextual models, data mining, pre-processing, linguistics.**

I. Introduction

The studies on offensive language detection have rapidly increased in recent years. However, detecting implication, irony, and other variations of hate speech still poses a challenge. Having a high-quality and comprehensive dataset is a key part of developing machine learning models that can automatically detect the use of offensive language in social media.

Nevertheless, manually creating such a labeled dataset is a challenging, error-prone and very costly task. In social media, Turkish is creatively used in both political trolling and popular culture [1]. This makes it even more difficult for models to detect real semantics of sentences without sufficient contextual information and presence of large representative dataset. Recently, [2] offered the OffensEval-2020 dataset, which is considered as carefully prepared, and commonly used for offensive language classification.

Although this dataset is labeled through manual effort, it still has a large room for improvement. It suffers from being biased on races, the lack of creative toxic phrases and hate speech with profanity commonly used in Turkish. Moreover, it can be observed that the dataset is imbalanced and models trained on it are overfitting to inoffensive context.

The aim of this study is to investigate the effects of augmenting existing labeled offensive language datasets by using an effective noisy contextual data augmentation method for offensive language identification including hate speech, cyberbullying, insulting.

The rest of this study is organized as follows: Section 2 remarks relevant studies; Section 3 explains the proposed linguistic-based data augmentation approach, and contains the dataset description and the technical details of the used learning models. Section 4 provides experimental validation with result analysis. In conclusion, section 5 contains a summary of our work and discussions.

II. Related Work

As a text classification task, offensive language detection has been investigated from various perspectives [3]. The task received a lot of attention as toxic online speech became an important problem due to growing interactions among online platform users with different cultural and educational backgrounds. In this section, we review related work with respect to the datasets and models used for offensive language detection particularly in Turkish.

Recently, several datasets have been published to provide foundation for studies to perform offensive language detection task such as Coltekin's OffensEval-2020 dataset [2], which is used as the baseline in our work, and a smaller dataset published by Mayda et al. [4], which focuses on hate speech. As part of the International Workshop on Semantic Evaluation (Sem-Eval2020), a group of researchers investigated various methods [5] and developed a CNN model in combination with LSTM, BiLSTM and Attention with BERTurk as an ensemble model which was evaluated on Coltekin's dataset. In the model performance evaluations, recall was not among the metrics reported by the authors of the study.

978-1-6654-7010-0/22/$31.00 ©2022 IEEE

Another study [6] investigated the performance impacts of various pre-processing methods on BERT models that were trained on Turkish corpora. The models were fine-tuned on Coltekin's OffensEval-2020 dataset. Results obtained by the study suggested that the performance was affected mainly by the BERT model and the impact of changes in pre-processing methods were minor in terms of model performance. Meanwhile, some of the studies focused on data augmentation to solve the common imbalanced label problem in offensive language datasets as in the case of our study [7], [8].

Wei et al. [9] investigated the methods that are utilized frequently for synonym word replacement based on pre-trained word embedding, random insertion, random swapping, and random deletion for English. Moreover, Wei et al. employed transfer learning to improve the results and provided the necessary metrics to analyse model performances. Other studies experimented with BERT for hate speech detection and classification on Twitter, and data augmentation using machine translation [10].

III. Methods

A. Overview of the proposed approach

Human annotation is very costly to create a sufficiently large dataset that can be used for detection of high level linguistic features of offensive language. Therefore, in order to solve the imbalanced label problem on the largest dataset [2], we attempt to combine publically available manually annotated corpora by augmenting "offensive" text from a corpus [4] and a dataset published on Kaggle*, with data mining methods to increase context variety rather than using synthetic interpolative methods such as SMOTE and ADASYN.

The main idea of our approach is to capture language features that are indicative of hate speech directed towards races, nations, religions, individuals, and other entities by using data mining techniques. Our method adds new samples to the original dataset to enhance its generalizability and robustness.

To test the effectiveness of our approach, we evaluated the performance of NLP models trained on the dataset by using both statistical and contextual embeddings (Word2Vec and BERT), a traditional machine learning method (SVM), and a deep learning pipeline (CNN-BiLSTM). Task codes, datasets and additional figures can be found at our GitHub repository†.

B. Datasets

Firstly, we utilized OffensEval 2020 Turkish Twitter dataset shared on Turkish Data Repository‡ for offensive speech identification in social media as a baseline for the task. More specifically, the Coltekin's training set has 28,000 sample with 22,596 not-offensive and 5404 offensive, development set has 3,756 sample with 3,029 not-offensive and 727 offensive, and test set has 2,812 not-offensive and 716 offensive labels.

To overcome imbalanced label problem, we extracted the offensive language samples from a Kaggle dataset* and another small dataset from [4]. The Kaggle dataset consists 5,054 total samples, where 2,073 of them are offensive. On the other hand, Mayda et al.'s dataset has 10,144 total samples and 2,502 of them are labeled as offensive.

Subsequently, 13,261 offensive tweets were extracted by our data mining method. We added 9,911 of them to the training dataset, and the rest were added to the test dataset. To get proper samples without any manual labeling, we utilized linguistic features of the relevant language as described in the following section.

Ultimately, the training corpus size before deleting duplicates was 42,486 with 22,596 not-offensive labels; 19,890 offensive labels. The distribution of class-labels in the ultimate version of the dataset was 22,589 and 19,809, respectively.

C. Pre-processing

1) Data Cleaning: During data cleaning, URLs, HTML tags, usernames, and emojis were removed from the datasets. Then, the text was transformed to lowercase in order to decrease variety and noise generated from identical words. We did not remove punctuation marks due to its effect on BERT embedding distributions. We observed that attention models can use information from punctuation marks between words, and when punctuation marks are removed from the text the performance decreases considerably. There is an existing work investigating this case [11]. We also noticed that the other pre-processing methods eliminate the contextual relationships of BERT. However, the datasets in our study do not have similar distributions. In other words, the final merged dataset was derived from extremely different distributions for the same task. We concatenated 3 different datasets and augmented further by our data mining method to decrease effects of class-distribution differences.

2) Text Normalization: Zemberek, an open source framework for Turkish language processing, was used for text normalization. We used the official version§. While normalization led to improvements in models using word embeddings based on a statistical approach, it produced worse performance results with contextual word embeddings. We think that lost contextual information due to heavy pre-processing was the root cause of the reduced performance as also discussed in [12].

D. Linguistic-based Data Augmentation: Data Mining by Language Features

Linguistic-based mining approach can help save us from manually labeling random data to solve the problems faced in label-imbalanced datasets for any contextual task. The proposed method simply starts with understanding the problem and the needs like in every other task. It then proceeds to find appropriate expressions using language features that can reach the target tweets, limiting the context of the text as little as

*http://kaggle.com/datasets/kbulutozler/5k-turkish-tweets-with-incivil-content

†http://github.com/Toygarr/lingda

‡https://data.tdd.ai/#/53b6ca44-e95e-443d-bb9e-978327c59fc5

§https://github.com/ahmetaa/zemberek-nlp/

possible. In our case, we have an imbalanced dataset for a highly context dependent detection task, and we are required to fix the imbalanced structure of the dataset due to the models' tendency of overfitting to the samples labeled as not-offensive. Meanwhile, while augmenting the offensive samples, it is crucial to keep the spontaneousness of sentences. Therefore, we cannot augment the offensive data by interpolation or weakly supervised generative models. Our approach facilitates automatically retrieving natural and indigenous sentences by only utilizing a few pre-defined word combinations. Moreover, the wide variety of additional natural context obtained by word combinations helps to avoid possible overfitting on non-offensive context. We chose the word combinations (phrases) as follows: A word list of swearwords and a list of entities (people, races, adjectives, etc.) were created manually and separate from each other. Every combination that satisfy the following formula was used as a query for tweets.

$$query = swear + entity \begin{cases} \text{suffix}_{singular}(entity) \\ \text{suffix}_{plural}(entity) \end{cases}$$

We aimed to extract offensive context from tweets as shown in Figure 1. A variety of queries were generated from a combination of potential offensive words with entities and their singular / plural direct object forms.

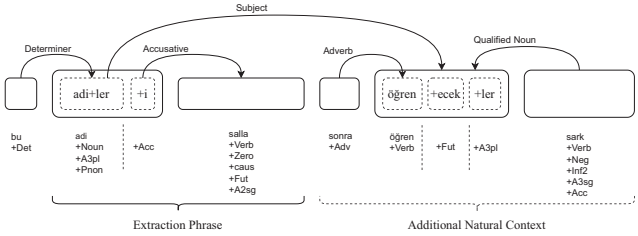

Fig. 1. Sample query result.

The tweet is then analysed morphologically that translates to "We should hang those shitboxes, so that they learn not to rackjack".

Swearwords and entities alone or even together are not sufficient to get a tweet that can be labeled as offensive automatically with high confidence, due to the use of swearwords to express strong expressions, emotions or in jokes as discussed in [13].

To illustrate this results of a query with "Japon" (Japanese) as an entity and "aq" (fu*k) as a swear word is shown in Figure 2. Note that "aq" is one of the abbreviations of the curse "f**k you" and one of the most used profanities by Twitter users in Turkey [14]. It can be observed from the sample tweets that even combining swear words with entities may yield tweets that are classified as non-offensive due to the widespread use of swearwords in the language. This happens especially on social media platforms where those words simply added to the end of the sentences for expressing a stronger emotion without their explicit meaning [15]. We used the method described above to construct a list of over 3000 queries. Those queries were used to retrieve tweets that we labelled as automatically as offensive.

Query	Tweet
japon (japanese)	Japon deyişi (The whole world would move aside for a human who knows what he want. A Japanese saying)
aq (fu*k)	Biraz sessiz kalayım belki beni özler dedim çocuk ortada yok aq (I said to myself, let's be quiet for a while, maybe he will miss me, but wtf, the boy is missing)
aq japon (fu*k japanese)	He knk baya halkın içinden :dd japon iş adamı sakura bile bu kadar sırıtmazdı aq (Yep bro really one of the people :dd fu*k, even sakura the japanese businessman wouldn't act that weirdly)
aq japon**u** (fu*k **the** japanese)	Sus aq Japon**u**. Sen nereden bileceksin (Shutup **you** fu**ing Japanese. How would you know)

Fig. 2. Examples of the results obtained, as seen in the last sentence, show why we should use linguistic features to obtain the determined target context, with the result of "aq japon**u**". The translations were written to reflect the meaning of the sentences in English after they were understood in Turkish by the authors. We tried not to deviate from the style of the tweets while adding context into account.

We also observed that our method is effective when combining homophones that can be used in a non-offensive context with entities. Using those words alone as queries result in tweets that cannot be labelled as offensive with confidence as in the example in Figure 3.

Query	Tweet
vururum (I hit)	vururum her gece kiyina (I still lap against your shores every night)
adamı vururum (I shoot **the** man)	Bir daha bu adamı kaale alıp da dinleyen Trabzonspor'luyu çeker vururum bak 0 şaka (I will shoot any Trabzonspor fan who listens to **this** man and takes him seriously. No kidding)

Fig. 3. An example of homophones and corresponding tweets

It worthwhile to note that swearwords or potential offensive words are not always directed at the entity in our query. Tweets containing entities with directed objects combined with potential offensive words tend to be offensive in almost all cases in our experiments. To assess our observation, we sampled a hundred tweets queried from Twitter using our method described previously, and asked two annotators to label the tweets as offensive or not-offensive. Only 2% of the tweets were labeled by both annotators as not-offensive, suggesting that our method is scalable and effective.

E. Models

1) Word2Vec: Word2Vec was trained using Gensim library v4.1.2 on our training dataset with a window size of 7 and a vector size of 300 for 16 epochs. Another version ($\text{word2vec}_{\text{large}}$) was trained using the same parameters on tweets collected by querying Twitter using a Turkish word list[¶] collected from VikiSözlük[‖], a total of 115,798 tweets (42,486 from the train dataset and 73,312 queried using the word list aforenamed) were used to train $\text{word2vec}_{\text{large}}$.

[¶]https://github.com/mertemin/turkish-word-list
[‖]https://tr.wiktionary.org

TABLE I

THE TABLE OF PERFORMANCE RESULTS FOR DIFFERENT MODELS. TEST A REPRESENTS COLTEKIN'S TEST SET. TEST B REPRESENTS OUR NEWLY CREATED TEST SET WHICH INCLUDES TEST A, AND ADDITIONAL RANDOMLY CHOSEN SAMPLES FROM THE NEW DATASET. OUR DATASET COLUMNS REPRESENT THE NEW DATASET THAT IS PROPOSED BY US TO TRAIN MODELS. RECALL METRIC REPRESENTS ONLY THE CORRECTLY LABELED "OFFENSIVE" SAMPLES. $\text{RECALL}_{\text{avg}}$ REPRESENTS THE MACRO AVERAGE RECALL SCORE OF BOTH "NOT-OFFENSIVE" AND "OFFENSIVE" LABELS. F1_{avg} REPRESENTS MACRO AVERAGE OF F1 SCORES OF THE RESULTS. WHETHER TEXT NORMALIZATION IS APPLIED OR NOT IS SHOWN BY COLUMNS TO INVESTIGATE EFFECTS OF TEXT NORMALIZATION.

Test A	*Without Text Normalization*						*With Text Normalization*					
	Coltekin's Dataset			*Our Dataset*			*Coltekin's Dataset*			*Our Dataset*		
Model	Recall	$\text{Recall}_{\text{avg}}$	F1_{avg}	Recall	$\text{Recall}_{\text{avg}}$	F1_{avg}	Recall	$\text{Recall}_{\text{avg}}$	F_{avg}	Recall	$\text{Recall}_{\text{avg}}$	F1_{avg}
Word2Vec-SVM	20.39	59.80	61.72	33.94	64.59	67.14	29.05	63.85	67.14	40.08	67.62	70.42
$\text{Word2Vec}_{\text{large}}$-SVM	23.04	60.88	63.21	38.97	66.71	69.22	27.65	63.06	66.09	40.92	67.67	70.21
BERT-SVM	44.83	70.99	74.74	57.96	75.37	76.71	43.58	70.17	73.71	56.28	74.80	76.48
BERT-CNN-BiLSTM	68.85	81.05	**81.59**	**86.31**	**83.80**	77.22	72.07	80.45	**81.59**	72.91	82.03	81.13
Coltekin Sub-Task A	-	76.20	77.30	-	-	-	-	-	-	-	-	-

Test B	*Without Text Normalization*						*With Text Normalization*					
	Coltekin's Dataset			*Our Dataset*			*Coltekin's Dataset*			*Our Dataset*		
Model	Recall	$\text{Recall}_{\text{avg}}$	F1_{avg}	Recall	$\text{Recall}_{\text{avg}}$	F1_{avg}	Recall	$\text{Recall}_{\text{avg}}$	F1_{avg}	Recall	$\text{Recall}_{\text{avg}}$	F1_{avg}
Word2Vec-SVM	69.81	84.60	84.28	81.86	88.66	88.62	76.81	87.86	87.73	84.55	89.99	89.97
$\text{Word2Vec}_{\text{large}}$-SVM	75.20	87.11	86.95	84.21	89.54	89.52	76.40	87.60	87.47	84.78	89.85	89.84
BERT-SVM	77.40	87.46	87.35	87.61	90.25	90.25	76.87	87.07	86.96	87.13	90.29	90.28
BERT-CNN-BiLSTM	91.12	92.22	92.23	**96.47**	88.88	88.80	92.91	**92.48**	**92.48**	93.18	92.35	92.35

2) BERT: We utilized BERTurk [16], a Turkish BERT model with 128k uncased vocabulary.

3) Training Models: We utilized the default version of the SVM Classifier. 1D CNN layers are utilized due to its capability to extract as many features as possible from the text. BiLSTM is utilized because it is effectively increase the amount of information available to the network, improving the context available to the algorithm.

IV. RESULTS

A. Evaluation Metrics

Recall and F1 were used for evaluating the performance of models. The Coltekin's dataset is imbalanced, thus using Macro F1 is a default choice as a metric. However, our main focus was increasing the recall score given the task objective. Recall[**] is the ability of a model to find all the relevant data within a dataset. In offensive language detection, finding non-offensive comments is not as important as identifying the offensive texts since failing to detect hate speech is much more harmful. We utilized the confusion matrix as shown seen in Figure 4 that presents the actual performance of finding offensive context. In Table I, we employed macro recall of offensive labels, $\text{recall}_{\text{avg}}$ and F1 scores to evaluate performance. We observed that using $\text{recall}_{\text{avg}}$ and F1 score can give misleading results as the Coltekin's dataset has imbalanced labels. F1 is calculated for N classes as follows:

$$F1_{avg} = \frac{1}{N}\sum_{i=1}^{\text{N}} \frac{2 \times precision_i \times recall_i}{precision_i + recall_i}$$

B. Training and Evaluation of The BERTurk Model Pipeline

The dataset was separated into training, validation and test sets. We only utilized the validation set to explore the epoch where our model starts learning, and its saturation point. We validated our model with 42,398 training, 1,756 validation, and 8,851 test samples. For BERT embeddings *"dbmdz/bert-base-turkish-128k-uncased"* was used. "max_length" parameter which determines max number of features is chosen as 200 dimensions. Moreover, BERTurk model adds extra 768 dims features which makes the output dims (, 200, 768). For the CNN module 32 dimensions were used with a filter size of 3. BiLSTM has 100 layers with 0.2 dropout rate on both ways. For the overall architecture, learning rate was 1e-5, Adam was used as an optimizer, and experiments were conducted with a batch size of 128. We trained the model for 3 epochs on the baseline dataset, and 18 epochs on our dataset until the model stopped learning. The difference on the number of epochs was unusual since in most of NLP tasks 3 to 6 epochs were the default choice and enough for BERT to learn a task. However, intuitively we assume that increasing the extreme contextuality requires more epochs to figure out the task for Turkish. All experiments were performed on Colab Notebook GPU and TPU offerings with Python 3.7.13 and Tensorflow 2.8.0. In BERT-CNN-BiLSTM evaluation, we noticed an interesting empirical issue with increasing batch sizes (ex. 64, 256) and different learning rates (ex. 1e-4, 1e-6) in both datasets where models may directly overfitted or simply diverged and learned nothing from data.

C. Model Performance Analysis

To assess the impact of variations in model contexts, we performed evaluations with and without text normalization on statistical and attention-based approaches using a range of models, including Word2Vec-SVM, $\text{Word2Vec}_{\text{large}}$-SVM, BERT-SVM, and BERT-CNN-BiLSTM. The comparison of model performance results is presented in Table I. Word2Vec and $\text{Word2Vec}_{\text{large}}$ demonstrated inconsistent results. In case

[**]http://scikit-learn.org/stable/modules/generated/sklearn.metrics.recall_score

of training without text normalization, $\text{Word2Vec}_{\text{large}}$ performed much better than the smaller Word2Vec. Yet it could not come close to the context-aware attention models. SVM can be considered as one of the fast alternatives to neural networks. Thus, we analyzed it with two word embeddings and a language model. Word2Vec does not provide enough word connections to be used by SVM to solve the task, while BERT offers more contextual information to distinguish the sentences. This explains the difference between Word2Vec-SVM and BERT-SVM on Test A. Meanwhile, BERT-CNN-BiLSTM pipeline dominated the results. BERT provides enormous amount of feature to extract, CNN can discover the important features to feed the BiLSTM, and BiLSTM uses extracted features to understand the connections between words. Furthermore, the effect of normalization is favorable in the case of statistical models such as Word2Vec that which require careful pre-processing due to their word root dependency. However, BERT is negatively affected by text normalization. BERT is a language model that can handle unknown and misspelled words. It is highly dependent on the sentence itself as a whole to determine the meaning of a word in a sentence. Overall, BERT-CNN-BiLSTM outperformed the other models in comparison in both test cases, demonstrating the advantage of providing more context with the dataset.

D. Error Analysis

In Figure 4 we present the confusion matrices of the BERT-CNN-BiLSTM pipeline that gives best results for the datasets without text normalization. In the following matrices "0" represents the "not-offensive" samples and "1" represents the "offensive" samples. Confusion matrices clearly showed the precision-recall trade-off. With the state-of-the-art model pipelines trained on Coltekin's dataset peaked at 0.69 true positives whereas after training on our new dataset it reached 0.86. Models trained on our dataset outperformed the models trained on the Coltekin's dataset in Coltekin's test set.

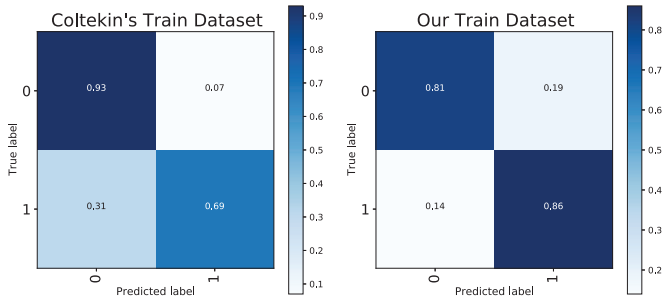

Fig. 4. Confusion matrices of the model results on Coltekin's test dataset.

In table II, we randomly chose samples from Coltekin's test set that is misclassified by the BERT-CNN-BiLSTM pipeline trained on our dataset. The table contains two positive and three negative labeled sentences. The first two samples (1-2) were marked as not-offensive, and the next three samples (3-5) marked as offensive context. We observed that there exists a notable number of mislabeled samples in the dataset. This situation limits the performances of all models. Samples 1, 4 and 5 were clearly mislabeled in the dataset where samples 2 and 3 were predicted wrong indeed. Sample 1 included clear hateful context in politics and the word "idam" had the same meaning as "execution". Sample 2 is a sentence that includes emotions, not hate. It resembles a piece of poem. There is no hateful context. The word abstractions may have led the model to be confused. Sample 3 ridicules about the current situation of the economy, however, it may include hate speech in hidden meanings. It is quite deep to be captured by our current model. Sample 4 is clearly mislabeled, it is a casual and a weird tweet but not offensive. In sample 5 it is hard to say that the person who says get rid of the preconception is hateful. The sentence has no offensive context at all.

TABLE II
MISCLASSIFIED EXAMPLES. A "OFF" REPRESENTS A OFFENSIVE, AND A "NOT" REPRESENTS A NOT OFFENSIVE SAMPLE.

ID	Text	Label	Predicted
1	mustafa kemal olsaydı olsaydı ama sadece chp'lileri idam ederdi şapka takmıyorlar diye. *(if mustafa kemal was here he would only have executed CHP supporters because they don't wear hats)*	NOT	OFF
2	bazen tek isteğim her şeyden kaçıp sana sığınmak. kendimi ve hayatımı bir anlığına unutup senin boynuna sokulmak öylece. *(Sometimes all I want is to run away from everything and take shelter in you. To forget myself and my life for a moment and just hug you by the neck.)*	NOT	OFF
3	ekonomi ile top yekün mücadele için bol bol sigara için, her paket de extra 2 tl katkınız olsun... *(for a total economical struggle have a lot of cigarettes, give an extra 2 tl contribution for each pack...)*	OFF	NOT
4	oha yuh cus kac gundur sizi bensiz biraktigim icin sorryy *(whoa oh ahhh sorry for leaving you without me for many days)*	OFF	NOT
5	önyargı şeysini bir kafanızdan atın artık *(get rid of the prejudice thing already)*	OFF	NOT

V. DISCUSSION AND CONCLUSION

We observed that Coltekin's dataset has significant problems and that the dataset is clearly imbalanced, includes many mislabeled samples and is racially biased. In this work, we focused on creating a solution to solve the imbalanced label problem that we observed in the existing offensive language identification datasets.

For this purpose, we employed a linguistic approach to augment data, BERT to extract high level language features, and a CNN-BiLSTM model pipeline to evaluate both the Coltekin's dataset and our new dataset. We investigated the effects of text normalization, the choice between word embeddings and

a language model, and the differences between a traditional machine learning method (SVM) and deep learning methods (CNN and BiLSTM). We shifted the metric focus for the task from macro average F1 to macro recall of offensive label given the task objective for real-life usage.

Our method can be used to increase data size where data quality is important and the labeling is very costly. With more contextual information, more generalized predictions tend to occur for extreme cases. We have shown that utilization of simple language features can be used to retrieve related contextual text, and solve the imbalanced label problem for contextual tasks such as offensive language detection. Moreover, we also presented the problems with the choice of a scoring metric since the existing dataset is imbalanced. We increased the proposed baseline macro average recall by 7.60%. We also dramatically improved the offensive recall score by 17.46% from 68.85 to 86.31 on the baseline test set.

Having said that, the proposed dataset still has room to improve. Precision-recall trade-off is obvious in our dataset, as the augmentation focus was limited to offensive texts. To decrease the gap between our recall and precision in a positive way, the available data size can be increased until satisfactory results are obtained. Furthermore, our dataset also endures the race bias retrieved from the Coltekin's dataset. This is clearly because we only retrieve offensive context about the races. In future work, our approach can be utilized to tackle the problem with racial bias by augmenting the data with positive context for a list of entities. By doing this, we think that models trained on the data can learn from positive and negative examples that contain references to the entity so as not to classify sentences based on mere mention of the entities in question.

Although Turkish was used to investigate the linguistic-based augmentation method, it can be employed for any language that uses the structure and features of the relevant language.

REFERENCES

[1] E. Bulut and E. Yörük, "Mediatized populisms— digital populism: Trolls and political polarization of twitter in turkey," *International Journal of Communication*, vol. 11, no. 0, 2017. [Online]. Available: https://ijoc.org/index.php/ijoc/article/view/6702

[2] Ç. Çöltekin, "A corpus of Turkish offensive language on social media," in *Proceedings of the 12th Language Resources and Evaluation Conference*. Marseille, France: European Language Resources Association, May 2020, pp. 6174–6184. [Online]. Available: https://aclanthology.org/2020.lrec-1.758

[3] J. Risch, R. Ruff, and R. Krestel, "Offensive language detection explained," in *Proceedings of the Second Workshop on Trolling, Aggression and Cyberbullying*. Marseille, France: European Language Resources Association (ELRA), May 2020, pp. 137–143. [Online]. Available: https://aclanthology.org/2020.trac-1.22

[4] I. Mayda, Y. E. Demir, T. Dalyan, and B. Diri, "Hate speech dataset from turkish tweets," in *2021 Innovations in Intelligent Systems and Applications Conference (ASYU)*, 2021, pp. 1–6.

[5] A. Ozdemir and R. Yeniterzi, "Su-nlp at semeval-2020 task 12: Offensive language identification in turkish tweets," in *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, 2020, pp. 2171–2176.

[6] A. Özberk and İ. Çiçekli, "Offensive language detection in turkish tweets with bert models," in *2021 6th International Conference on Computer Science and Engineering (UBMK)*. IEEE, 2021, pp. 517–521.

[7] V. Rupapara, F. Rustam, H. F. Shahzad, A. Mehmood, I. Ashraf, and G. S. Choi, "Impact of smote on imbalanced text features for toxic comments classification using rvvc model," *IEEE Access*, vol. 9, pp. 78 621–78 634, 2021.

[8] E. Ekinci, "Classification of imbalanced offensive dataset–sentence generation for minority class with lstm," *Sakarya University Journal of Computer and Information Sciences*, vol. 5, no. 1, pp. 121–133, 2022.

[9] B. Wei, J. Li, A. Gupta, H. Umair, A. Vovor, and N. Durzynski, "Offensive language and hate speech detection with deep learning and transfer learning," *CoRR*, vol. abs/2108.03305, 2021. [Online]. Available: https://arxiv.org/abs/2108.03305

[10] J. Liu, Y. Yang, X. Fan, G. Ren, L. Yang, and Q. Ning, "Offensive-language detection on multi-semantic fusion based on data augmentation," *Applied System Innovation*, vol. 5, no. 1, p. 9, 2022.

[11] A. Ek, J.-P. Bernardy, and S. Chatzikyriakidis, "How does punctuation affect neural models in natural language inference," in *Proceedings of the Probability and Meaning Conference (PaM 2020)*. Gothenburg: Association for Computational Linguistics, Jun. 2020, pp. 109–116. [Online]. Available: https://aclanthology.org/2020.pam-1.15

[12] M. Pota, M. Ventura, H. Fujita, and M. Esposito, "Multilingual evaluation of pre-processing for bert-based sentiment analysis of tweets," *Expert Systems with Applications*, vol. 181, p. 115119, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417421005601

[13] R. Jonsson, "Swedes can't swear: Making fun at a multiethnic secondary school," *Journal of Language, Identity & Education*, vol. 17, no. 5, pp. 320–335, 2018.

[14] M. Ciot, M. Sonderegger, and D. Ruths, "Gender inference of twitter users in non-english contexts," in *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, 2013, pp. 1136–1145.

[15] W. Wang, L. Chen, K. Thirunarayan, and A. P. Sheth, "Cursing in english on twitter," in *Proceedings of the 17th ACM conference on Computer supported cooperative work & social computing*, 2014, pp. 415–425.

[16] S. Schweter, "Berturk - bert models for turkish," apr 2020. [Online]. Available: https://doi.org/10.5281/zenodo.3770924