HATE SPEECH DETECTION IN TURKISH NEWS USING A TRANSFORMER-BASED MODEL ENHANCED WITH LINGUISTIC FEATURES

by

Atıf Emre Yüksel

B.S., Computer Engineering, Boğaziçi University, 2019

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science

Graduate Program in Computer Engineering

Boğaziçi University

2022

ACKNOWLEDGEMENTS

I would like to thank Assoc. Prof. Arzucan Özgür for her support, kindness, patience and guidance during my Master's study and research.

I would like to express my gratitude to my jury members, Prof. Dr. Tunga Güngör and Prof. Dr. Olcay Taner Yıldız, for their valuable insights, questions and contributions. I am also grateful to Assoc. Prof. Didar Akar for evaluating the results and providing various approaches to this project in the field of linguistics.

The computer on which all experiments were performed, as well as the data set and annotations, were provided by the Hrant Dink Foundation. I would like to thank the Hrant Dink Foundation for quickly answering questions about the dataset, for working out the annotation methodology, and for their assistance with computational resources. I would also like to thank all the experts and volunteers who carefully annotated the dataset.

Finally, I thank with love to my wife Necla for her encouragement, motivation, and her endless patience. Without her support, this process would have been much more difficult.

ABSTRACT

HATE SPEECH DETECTION IN TURKISH NEWS USING A TRANSFORMER-BASED MODEL ENHANCED WITH LINGUISTIC FEATURES

Hate speech directed at ethnicities, nationalities, religious identities, and specific groups has increased not only in social media, but also in print media. This creates a need for automated hate speech detection systems that can quickly review and filter print media content before it is provided to readers if it contains hate speech. However, most of the existing automatic hate speech detection models are limited to detecting hate speech without considering the hate speech target group-specific discourse that is often used in news articles. Moreover, there are few datasets that include Turkish print media articles in the hate speech domain.

In this study, a new BERT based model enriched with a set of target-oriented linguistic features for hate speech detection is proposed. The effects of weighting different BERT hidden vectors are also investigated, instead of using only the first hidden vector of the BERT encoder, which is the classical approach. New BERT based models that integrate different attention techniques are proposed for combining hidden vectors. A new preprocessed Turkish dataset for hate speech is also published, in which the target group for all hate speech articles is annotated. Experiments on a comprehensive Turkish dataset of news articles labeled for hate speech show that competitive performance in terms of accuracy and F1-score is achieved compared to previous approaches.

ÖZET

DİLBİLİMSEL ÖZELLİKLER İLE DESTEKLENEN DÖNÜŞTÜRÜCÜ TABANLI MODEL KULLANILARAK TÜRKÇE HABERLER ÜZERİNDE NEFRET SÖYLEMİ TESPİTİ

Etnik kökenlere, milliyetlere, dini kimliklere ve belirli gruplara yönelik nefret söylemi sadece sosyal medyada değil, yazılı basında da artmaktadır. Bu artış basılı medyada okuyuculara sunulmadan önce nefret söylemi açısından hızlıca gözden geçirip filtreleyebilen otomatik nefret söylemi tespit sistemlerine ihtiyacı oluşturmaktadır. Ancak mevcut otomatik nefret söylemi tespit modellerinin çoğu haber metinlerinde sıklıkla kullanılan hedef grubuna özgü söylemi dikkate almadan nefret söylemini tespit etmekle sınırlıdır. Ayrıca nefret söylemi alanında Türkçe yazılı basın makalelerini içeren az sayıda veri seti bulunmaktadır.

Bu çalışmada nefret söylemi tespiti için birçok hedef grup odaklı dilbilimsel özelliklerle zenginleştirilmiş BERT tabanlı yeni bir model önerilmiştir. Klasik yaklaşım olan BERT kodlayıcının sadece ilk gizli vektörünü kullanmak yerine farklı BERT gizli vektörlerini ağırlıklandırmanın etkileri de araştırılmıştır. Gizli vektörleri birleştirmek için farklı dikkat katmanı tekniklerini kapsayan BERT tabanlı modeller önerilmiştir. Nefret söylemi etiketi ile birlikte nefret söylemi olarak etiketlenen haberlerin hedef kitlesinin de belirtildiği önceden işlenmiş yeni bir Türkçe nefret söylemi veri kümesi de yayınlanacaktır. Bu kapsamlı Türkçe yazılı basın veri seti üzerinde yapılan deneyler, önceki yaklaşımlara kıyasla doğruluk ve F1 puanı açısından rekabetçi bir performansın elde edildiğini göstermektedir.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS . . . iii
ABSTRACT . . . iv
ÖZET . . . v
LIST OF FIGURES . . . viii
LIST OF TABLES . . . ix
LIST OF SYMBOLS . . . xii
LIST OF ACRONYMS/ABBREVIATIONS . . . xiii
1. INTRODUCTION . . . 1
2. RELATED WORKS . . . 4
2.1. Hate Speech Detection Models . . . 5
2.2. Attention Layers . . . 7
2.3. Linguistic Features . . . 9
2.4. Hate Speech Datasets . . . 12
3. DATASET . . . 14
4. METHODOLOGY . . . 19
4.1. WeBERT . . . 20
4.2. AttWeBERT . . . 22
4.2.1. Gating Function . . . 23
4.2.2. GRU Encoder . . . 23
4.2.3. Attention Decoder . . . 23
4.2.4. Class Embedding Layer . . . 25
4.3. RuleBERT . . . 25
4.3.1. General Features . . . 27
4.3.2. Special Features . . . 29
4.3.3. BeforeTarget Features . . . 29
4.3.4. AfterTarget Features . . . 31
4.3.5. HSPointedVerb Features . . . 31
4.3.6. AntiHate Features . . . 31

4.3.7. Model Architecture 32
4.4. OnlyRule Baseline Model 33
5. EXPERIMENTS AND RESULTS 38
5.1. Experiment Details 39
5.2. Results 40
5.3. Ablation Study 42
5.4. Error Analysis 42
6. CONCLUSION AND FUTURE WORK 44
REFERENCES 47

LIST OF FIGURES

Figure 4.1. Architecture of baseline BERT model. O represents the output of classification (self-drawn). 19

Figure 4.2. Architecture of WeBERT model. O represents the output of classification (self-drawn). 21

Figure 4.3. Architecture of AttWeBERT model. O represents the output of classification (self-drawn). 24

Figure 4.4. Architecture of RuleBERT model. O represents the output of classification (self-drawn). 26

LIST OF TABLES

Table 3.1. Dataset statistics. 14

Table 3.2. Class distribution of each split in the dataset. 15

Table 3.3. Keywords covered by the dataset. 16

Table 3.4. Examples of hate speech expression from the dataset. 17

Table 3.5. The number of news which contains hate speech discourse against the specific group. 18

Table 4.1. Subtypes and feeding method of target-oriented linguistic features. The subtypes indicate the severity of the hate speech. The cumulative value is a vector with a value that accumulates the number of detected rules in a given feature type. The dimension of the one-hot vector is determined by the number of subtypes. 27

Table 4.2. Structure of the generated linguistic features. ETHN (IRK) substitutes for any ethnicity in Table 3.3. Count indicates the number of patterns used in feature generation. Window determines in which window range the pattern is searched. No window indicates that patterns must be searched by strict matching. 28

Table 4.3. Patterns used in special type of linguistic features. Higher degree points more serious hate speech discourse. 30

Table 4.4. Frequently used verbs in hate speech discourse. They are detected when they are used with one of target in given window range. Higher degree points more serious hate speech discourse. 32

Table 4.5. Misleading hate speech discourse that are found mostly non hate speech news. ETHN (IRK) substitutes for any ethnicity to generate feature. A higher rating indicates less or no hate speech. "ETHN+ist" expresses ethnicity names and the suffix "-CU" in Turkish, which is derived nationalist names from it by attached them (e.g. Türkçü, Kürtçü). 34

Table 4.6. Adjacent words that written right before keywords in news. Higher degree points more serious hate speech discourse. 35

Table 4.7. Adjacent words that written right after keywords in news. Higher degree points more serious hate speech discourse. 36

Table 4.8. General patterns that are widely used in hate speech discourse. They were extracted from large corpus by experts. ETHN (IRK) substitutes for any ethnicity to generate feature. Patterns are searched in determined window range. BEFTAR corresponds that words in Table 4.6 can be placed. AFTERTAR indicates that words in Table 4.7 can be put. 37

Table 5.1. Best combination of hyperparameters of AttWeBERT model. . . . 39

Table 5.2. Evaluation of various models over different test sets. All keywords news includes news that are written about all target keywords. Ukraine news contains only the ones published about the Ukraine war. w/ 4 hidden vectors indicates the model weighting 4 attention hidden vectors and w/o Att indicates the one without using Attention decoder. The scores are represented as percentages (%). 41

LIST OF SYMBOLS

c_i	Importance coefficient of h_i
dec_i	Output vector of GRU decoder in timestamp i
e_i	Embedding vector of h_i
exp	Exponential function
$E_{[CLS]}$	Embedding vector of the start token
h_{final}	Final hidden vector of WeBERT
h_{FCN_i}	ith hidden vector of FCN of weighting layer of WeBERT
h_i	Hidden vector of BERT encoder
h'_i	Hidden vector multiplied by importance coefficient
out_{enc_i}	Attention input vector from GRU encoder in timestamp i
out_i	Output vector of GRU encoder in timestamp i
out_{dec_i}	Output vector of Attention GRU decoder in timestamp i
$tanh$	Hyperbolic tangent function
w_i	Weighted out_{dec} in timestamp i
W_{FCN}	Weight matrix of fully-connected layer
W_{emb}	Weight matrix of embedding layer
α_i	Attention weight of out_i
$\oplus$	Dot product
σ	Sigmoid function

LIST OF ACRONYMS/ABBREVIATIONS

BERT	Bidirectional Encoder Representations from Transformers
BiLSTM	Bidirectional Long Short-Term Memory
CLS	Class token
CNN	Convolutional Neural Network
FCN	Fully Connected Network
GPU	Graphics Processing Unit
GRU	Gated Recurrent Unit
HDV	Hrant Dink Foundation
LSTM	Long Short-Term Memory
NLP	Natural Language Processing
RNN	Recurrent Neural Network
SemEval	Semantic Evaluation Competition
URL	Uniform Resource Locator

1. INTRODUCTION

According to the Council of Europe's Council of Ministers Recommendation, hate speech can be defined as "all forms of expressions which spread, incite, promote or justify racial hatred, xenophobia, anti-Semitism or other forms of hatred based on intolerance, including: intolerance expressed through aggressive nationalism and ethnocentrism, discrimination and hostility against minorities, migrants and people of immigrant origin". However, as the United Nations noted, "Under International Human Rights Law, there is no universal definition of hate speech as the concept is still widely especially in regards to its relation to freedom of opinion and expression, non-discrimination and equality." Thus, since the boundaries of hate speech are controversial, in this study we have defined hate speech in a way that includes discriminatory speech; both because they are highly related to each other and because the boundaries of the concepts cannot be precisely determined. Essentially, hate speech is the defamation of a group based on differences such as ethnicity, nationality, religion, color, gender, sexual orientation, and other characteristics [1]. As hate speech becomes more widespread in the printed news, it becomes easier for authors to incite society against certain groups and intentionally spread war speech. Although there are certain regulations to prevent this content, authors effectively use hate speech through various techniques such as symbolization and distortion. For instance, authors prefer repetition to express hate speech against a certain ethnic group, use the technique of contrariety where one group is appreciated and the target group is criticized without using colloquial words, and deliberately point out that a person from the target group is involved in a shameful event in the news.

In addition, manual detection of hate speech is very challenging, time-consuming, and labor-intensive because hundreds of Turkish news stories are published every day, and few articles that spread hate speech contain slang that can be detected through a dictionary check. However, most articles are carefully written and express hate speech implicitly. Moreover, opinion columns are very long to quickly examine the content of

hate speech. For this reason, the systematic and rapid spread of hate speech in society highlights the need for automatic hate speech detection systems. However, the most important need of automatic hate speech detection systems is annotated data that requires manual labelling by experts.

Turkish is a low resource language for which there are fewer open-source datasets than for languages that are well researched in NLP, such as English and German. Even when the focus point becomes more specific, it is difficult to find suitable datasets, such as hate speech [2]. In addition, most open source datasets come from Internet media, and creating datasets that include written media is quite expensive. Fortunately, the HrantDink Foundation has undertaken a project to create a Turkish hate speech corpus with instances from printed media. The corpus includes articles from various newspapers and by various authors with publication dates between 2012 and 2019. In addition, HDV has established an annotation methodology for labelling hate speech that has been followed by all experts to reduce bias in the annotation of the corpus. In this study, we use a comprehensive dataset of printed media that was manually annotated and the target group provided by HDV.

In addition to the importance of the annotated data, the proposed model is another important criterion for developing a successful system. With the significant advances in pre-trained large language models and transformer architecture in natural language processing, researchers have developed various architectures based on BERT [3], which have achieved successful results in the area of hate speech detection [4–8], where they have adapted the pre-trained BERT architecture to fine-tune their own data. However, the effect of auxiliary features was not explored in these models to increase the performance of the model. Although lexical and linguistic features have been used in different model architectures [9–12], target-oriented linguistic features and their effects on the model have not been studied in detail. Also, the methodology of target group oriented linguistic features is intended to be applied in different languages in the field of hate speech.

On the other hand, BERT is capable of training different models for various downstream tasks such as sequence classification, token classification, and others. In the original work of BERT, custom sequence classification is trained on the first hidden vector of the BERT encoder. Subsequent work based on BERT has used only the hidden vector of the transformer's CLS token and has not investigated the effect of using other hidden vectors of tokens in the fine-tuning phase. In other words, the effect of combining multiple hidden vectors on performance is an important hypothesis, although the first token of BERT is identified as a special token. Moreover, multiple hidden vectors can be combined in different techniques. In this study, different attention methods are integrated into the model architecture. The main contributions of this study are summarized as follows:

- RuleBERT, a novel model that enhances BERT with hate speech target-oriented linguistic features, is proposed. The linguistic features are categorized and ranked in terms of their use and severity.
- The effects of weighting multiple hidden vectors of the BERT model are explored. Instead of using only the hidden vector of the CLS token, different attention methods are adapted to learn the soft attention weights, and their contributions are demonstrated.
- Preprocessed Turkish hate speech dataset, trained models, and the source code are made publicly available.

2. RELATED WORKS

Hate speech detection has been studied in the field of computer science since 2004 by adapting classification models. With the tremendous developments in the field of machine learning, especially deep learning, researchers have applied various techniques to this topic, and new open-source datasets have been published, increasing the number of works. Although the main focus of researchers has been on more general problems such as the development of large-scale language models, much attention has been paid to the automatic detection of hate speech due to its positive impact on society. Moreover, researchers have found that fitting state-of-the-art models alone is not sufficient to propose the best model, and that they need to be supported by additional domain-specific features. In this work, a state-of-the-art language model and different attention layers are used and domain-specific linguistic features are generated.

Section 2.1 explains the hate speech detection models chronologically, based on different architectures. Some studies trained their own domain-specific embeddings before recent advances in language modeling, and more recent work uses the embeddings of large language models such as BERT. Different deep learning architectures such as CNN [13], LSTM [14], GRU [15] and BERT have been tried.

Attention is an important concept for learning the relationship between words, sentences, and documents. There are several approaches to incorporate it into models. Although it has become popular after the great success of the transformer architecture [16], it has been used in previous studies. Also, type of attention weights is another important point to use it for a custom problem. Previous studies are explained in Section 2.2.

NLP models can be supported by well-designed auxiliary features rather than using only the representation (embedding) of words. Also, using additional features is a good option to increase success when the problem is more specific and the number

of instances in the user-defined dataset is not sufficient to train a model from scratch or refine another model. However, generating features that match the dynamics of the domain is challenging. For example, statistics-based features (e.g., the number of capital letters used) may be developed in different ways depending on the source of the data. If the data come from online social media, the number of asterisks and emojis may be significant, whereas for print media they may be insignificant. The linguistic features used in previous studies are explained in Section 2.3.

Another important point, which is as important as proposing good hate speech models, is the creation of datasets. Researchers have generally published a hate speech dataset consisting of texts from Internet media in well-studied languages such as English. On the other hand, the scope of target groups is an important criterion for hate speech datasets, as the definition of hate speech can be modified accordingly. Full details are given in Section 2.4.

2.1. Hate Speech Detection Models

With the breakthrough of word embeddings in NLP, many different deep learning architectures have been developed for automatic detection of hate speech. CNN has been mainly developed and used for computer vision and speech recognition tasks because they can extract local receptive features via filters without exceeding the computational constraints. But local features and convolutional operations can also be beneficial for NLP models. For this reason, Gamback [17] adapted convolutional neural networks to tweets by combining word embeddings and character n-grams as input. They trained their own word embeddings following Mikolov's work [18] and compared this input representation to character n-grams in CNN. Although they combined two different input representations in one model, the CNN model using only word embeddings has the best result in terms of F1 score. Moreover, Pavlopoulos [19] used a vanilla CNN model as a solid basis for comparing his own proposed model, where the CNN model architecture is based on Kim's model [20] and the input representation is word embeddings. Zhang [21] proposed an architecture that uses CNN and GRU together

to detect hate speech in tweets. They used a one-dimensional convolutional layer with 100 filters to extract features from word embeddings. These features were fed into the GRU layer after applying dropout and max-pooling layers. Park and Fung [22] developed HybridCNN model that includes both character-based and word-based CNN models. They demonstrated the effectiveness of hierarchical detection of hate speech from general (offensive) to specific category (racist).

RNN-based architectures such as LSTM and GRU were used for hate speech detection. Pavlopoulos [19] proposed an RNN-based model whose layers are GRU on user comments to detect offensive comments. They also adapted the attention mechanism to weight hidden vectors. The weighted hidden vectors were used as input to the logistic regression for classification. Pitsilis [23] preferred the embedding layer instead of using pre-trained word embeddings in RNN-based multiple LSTM models. Founta [24] proposed a combined model with two different submodels. The input of the first sub-model is only a text representation, while the other uses only metadata as input. Thus, different types of data are learned in different layers. Then, their results are combined, and joint classification layers yield the final prediction about hate speech decision. Perez and Luque [25] fed contextual embeddings obtained from a bidirectional language model [26] and bag-of-words features, and these embeddings were used in a unified model. In other words, the contextual embeddings of text input were fed into the RNN layers, while the bag-of-words features were passed to the feed-forward layer. Indurthi [27] developed a model for detecting hate speech for immigrants and women, but used sentence embeddings in decision tree models rather than a deep learning architecture.

Along with the release of large-language models on different languages, training models for user-defined problems has become much easier. BERT, the most famous example, which is able to present 11 different downstream tasks after receiving a pre-trained encoder module. Therefore, the weight of the BERT encoder is used in the hate speech detection task instead of training custom word embeddings on a limited dataset. Although the original BERT publication included embeddings for only a few

well-studied languages, several language models based on BERT have been published for resource-poor languages [28, 29]. Also BERT was trained on datasets of specific tasks, such as abusive language whose instances are banned Reddit comments, and Caselli [7] showed a comparison of BERT and the retrained version on known English abuse datasets. Mozafari [4] developed several fine-tuning approaches for the pre-trained model BERT and demonstrated their performance on tweets related to racism and sexism. In all experiments, the BERT encoder was used as the backbone module and the BiLSTM, CNN, and FCN layers were initialized for the finetuning phase. Perifanos [8] refined the model pre-trained in Greek BERT, to detect hate speech against refugees and migrants. In addition, a different implementation of the BERT architecture was used, which is faster, and the model was trained with a multimodal approach using image content from tweets. Koufakou [11] used the BERT encoder and reported results on several English datasets. They also used additional features to improve the performance of the model.

2.2. Attention Layers

Attention is a technique that finds relationships between instances of the layer to learn the optimal weights for each input. In the early years, attention was proposed for developing more robust alignment modules for neural machine translation tasks. In these studies, attention weights are computed between the inputs and hidden vectors of the RNN decoder. Although this attention approach achieved successful results in translation tasks, they were not suitable for all NLP tasks. For this reason, the mechanism of self-attention was developed by Vaswani [16], which is a pioneering study in NLP. Self-attention allowed large language models to be trained unsupervised, and many downstream NLP tasks could be improved thanks to this final encoder module. For example, BERT and its variants were developed using self-attention as the basis for their own architecture.

In addition, attention contains a set of importance coefficients by which each input is weighted to obtain a weighted output vector. However, the attention coefficient

can be calculated in different ways, namely hard and soft attention. In hard attention, the attention weight is used as the sampling rate for selection probability of the corresponding input in the final vector. On the other hand, soft attention generates the attention weight of each input to multiply it. In the NLP field, soft attention is a well-accepted approach on which most of the attention research is based.

Bahdanau [30] introduced an attention mechanism for neural machine translation tasks. In this work, attention was used in the RNN decoder layer when the context vector of the corresponding input word was computed. In their alignment model, they used a soft attention approach. They also used a bidirectional encoder module to encode the input sentence. This is the first study in which content-based attention scores are computed and successfully used in an alignment model. Luong [31] developed an attention mechanism similar to Bahdanau's work on using RNN encoder-decoder in neural machine translation. However, the main difference is that global and local attention values are computed and combined into an overall context vector for each input word. Local attention finds a relationship between a word and a subset of target words, while global attention establishes relationships with all target words. Luong's attention mechanism is one of the foundations of this thesis to propose an attention-based model. Yang [32] proposed a hierarchical model architecture, each using a word-level module and a sentence-level module. After the words are transformed by the word encoder, the word attention layer computes the attention weight of each word to create a sentence vector. The sentence encoder extracts the document embedding from all sentence vectors and the sentence attention layer determines the importance of each sentence to generate the final document vector.

Vaswani [16] developed the Transformer architecture. The Transformer module consists of a self-attention layer, a fully connected layer, and a dropout layer. The self-attention layer learns the relationship between each pair of input embeddings. Therefore, it generates not only a directional language model, but also a language model that captures long and short term relationships. Although self-attention is computationally intensive due to NxN matrix multiplication, new studies simplify and

speed it up. In this thesis, transformer based encoder module was used as the basis for all experiments.

2.3. Linguistic Features

Additional features are supportive indicators for deep learning models in order to increase performance. However, they should be designed carefully. For instance, the corpus should be investigated in details, experts should evaluate the design of features, and they are supported by evidence extracted from corpus. The source of the corpus, most frequent patterns are important criteria to create features. For this reason, researchers developed surface level and lexical features. However, some patterns cannot be identified easily like in surface and lexical features and they can be placed in the context. That's why, complex linguistic features should be designed for robust auxiliary features. For instance, implicit hate speech is generally detected by not detecting specific words, the message of hate speech is embedded in the context of the articles.

Surface-level features record sentences in an occurrence matrix that stores the frequency of words in related word columns. Dimensionality is very high when a large corpus is used, and there is no contextual relationship between words, although word embeddings store this type of information. N-grams are also an example of these surface features. Nobata [9] used these features for their model with many linguistic and lexical features. They reported that the character n-gram feature is most effective in the experiments. The reason why the character n-gram is so powerful is that the language of social media is different from that of printed media, and trolls and haters usually write their messages with hate speech and misspellings. So the character n-gram is able to model these types of errors in the sentence [33].

Moreover, the statistical features are another type of surface features. These statistical features have been used for hate speech detection with the n-gram features. Well-known examples of statistical features are the number of URLs, token length,

capitalization, and the number of words that are not in the dictionary. For example, Nobata [9] created statistical features in social media texts, such as the number of punctuation marks, average word length, number of URLs, number of modal words, and number of one-letter tokens, and measured their effects on the model.

Although surface features have been used for text classification models in the past, they brought with them the problem of data sparsity. The reason for this problem is that each sentence is represented with a vector containing many zeros when the number of words is higher. Even Turkish language has different suffixes to express moods and tenses, so the problem of sparseness becomes more serious. Several works have been presented to reduce feature complexity in text classification, such as word clustering. Brown [34], Warner and Hirschberg [35] created clusters for each set of words, with each word represented by its own cluster centroid. However, this approach has several key drawbacks. First, the information that allows words to be distinguished disappears, and assigning a new word to a cluster is very complicated. Therefore, the word embedding approach [18] is more suitable for expressing semantic similarity between words.

Lexical features are another type that controls whether a particular word of the lexicon is found in the instance or not. Thus, specialized lexicons are good discriminators such as ethnic slurs, irritant words for the disabled, and terminological slurs about religions. In previous work, these features were not used by themselves, but were used as auxiliary features in addition to n-grams or word embeddings. Razavi [36] used a dictionary of words and phrases in offensive language. Wiegand [10] created a dictionary of swear words collected from online resources. Koufakou [11] used lexical features to fine-tune BERT and shared a hate speech lexicon for offensive language datasets. The lexical corpus is compatible with 53 languages, which is the strength of the work. Moreover, the lexical features are represented by both one-hot encoding and embedding versions and concatenated in the output of the encoder. However, building a custom dictionary is a time-consuming manual process, and lexical resources have not been used as much in recent years after the publication of comprehensive language models.

Linguistic features have been used in hate speech detection to model various relationships not recognized by the developed model. Xu [37] used POS information on n-grams features from tweets to obtain a better model, but there is no significant difference between this approach and the baseline. Chen [38] generated typed dependency relations that have good features for non-consecutive words to model interactions between pairs of words. Since some deep learning models are not very good at storing long-term information due to the forgetting problem, the detection of this syntactic feature is not sufficient in this work because the semantic relationship between words also plays an important role in the detection of hate speech. Therefore, all semantic and syntactic features must be evaluated together.

In addition, Burnap and Williams [39, 40] used grammatical dependencies to detect othering phrases. For example, they generated typed dependencies for groups of words. To create othering phrases, they tagged the keywords "them" and "us" in the generated phrases and showed that the use of othering language is a useful feature for detecting hate speech. They also used the determiner typed dependency to express another "othering" phrase that is an example of a generalized hate speech feature: the indication of larger groups. Hüsünbeyi [12] applied the model HAN and used linguistic features extracted from othering language in Turkish print media. These features improved performance in all metrics.

Although syntactic grammatical relations are very important, hateful discourse can depend on context. For example, a sentence may be hateful to Muslims, but it may also be hateful to other religious groups. For this reason, the expertise and sociological knowledge are very important in shaping the linguistic features and lexical resources. Cultural background is also an important factor when someone evaluates an article in terms of hate speech content. In addition, linguistic features may differ in both directed and generalized hate speech. They can be generated by analyzing the aspect of hate speech, and some features can be identified for each type. For example, directed hate speech focuses on a specific target (person, location, organization), it is interested in more informal and coarse language, using these words directly, while generalized

hate speech highlights large groups, uses softer language with higher authenticity, and contains the least sexual references [41].

2.4. Hate Speech Datasets

Hate speech datasets are mostly from online sources, usually tweets on Twitter, and there are fewer datasets in low-resource languages such as Turkish. Hate speech datasets may be unbalanced because random sampling finds more tweets that do not contain hate speech. However, annotating tweets that contain hate speech is costly. For this reason, the number of balanced hate speech datasets is very small [42]. Domain is also an important criterion for defining scope. For example, some studies focus only on sexist and racist terms, while others collect data that include political terms. In addition, the target is an important annotation type for hate speech datasets because each hate speech discourse directs its hate against a specific target. For this reason, including the target label in the dataset increases its quality. Finally, some datasets are created only for detecting implicit hate speech, which is different from the majority of hate speech datasets.

Zampieri [43] published the OffensEval dataset, which contains English tweets and focuses on political phrases with annotation of the targets and the type of offensive language. The dataset contains annotations at three levels, namely whether the tweet is offensive, the category of offensive language, and the target group. Thanks to the richness of different annotation types, different models and tasks can be performed with OffensEval. Therefore, each annotation type and the corresponding tweets for different common tasks were published in SemEval 2019. In this competition, Basile [44] presented the HatEval dataset focusing on women and immigrants in English and Spanish tweets. In addition, Sap [45] created a dataset that includes annotations on target groups and implied statements collected from various online sources. This dataset is a large corpus in which 150,000 tweets are annotated as hate speech.

ElSherief [42] created an English tweet dataset that includes annotations on target groups, implicit distinction of hate speech, and implied statements. Although Sap's dataset contains implied statements, it specializes in implicit hate speech, which is different from all open source datasets in the hate speech domain. Mollas [46] collected comments on Youtube and Reddit through a crowdsourcing platform and developed an annotation protocol. In this protocol, a validation mechanism was created to reduce annotator bias in the dataset. It also includes a type annotation whose categories are directed and generalized. Mathew [47] created a dataset of HateXplain tweets with target group and multiple category annotation and evaluated the benchmarks using the bias and interpretability metrics. In addition, they evaluated offensive language and hate speech in multiple categories.

3. DATASET

The dataset is a collection of printed news published in Turkish media from various sources. It is a balanced dataset in terms of class distribution of hate speech and non-hate speech. General statistics are listed in Table 3.1 and class distribution is demonstrated in Table 3.2. In splits of the dataset, two different test set are determined. Although the test set containing only articles about the war between Ukraine and Russia is smaller than the others, it is important to test the model with the latest topics and evaluate the performance of the model with targets that are not included in the training set. Examples are listed in Table 3.4 with label of targeted group. Also, targeted group distribution of hate speech news is listed in Table 3.5.

Table 3.1. Dataset statistics.

Statistics	**Detail**
The number of samples	25015
The number of sources	1242
Time period	2014 - 2019
Avg. number of sentences per news	37.85
News / Opinion column	46,21% / 53.79%

To collect hate speech news, an annotation methodology was established, 21 experts evaluated each news within its scope, and assessing authority controlled the quality of the ones labelled as hate speech. News must contain at least one target keyword to be read by the experts. These keywords are listed in Table 3.3.

The keywords are divided into 4 different groups, namely ethnicity, nationality, national & religious identity, and other groups. Ethnicity includes the majority of keywords because it is the most commonly used in hate speech articles. Nationality also includes only the keyword “Western”, which is often used to describe all ethnicity

in Europe with hate speech. Religious identity is another open wound in Turkish media, as authors use these keywords to distinguish certain religious groups from others. For example, the keyword “infidels” is often used by some media groups to describe a certain group of people as infidels, although there is no evidence to support this claim. Other groups include various communities that are given certain terms in the Turkish media for political and demographic reasons. For example, the term “refugee” has been used frequently in printed media since the beginning of the Syrian war.

Table 3.2. Class distribution of each split in the dataset.

	Hate speech	**No hate speech**
Training	10074	9944
Validation	1257	1243
Test (all keywords)	1255	1242
Test (only Ukraine)	15	15

To collect a corpus of printed media, each pebbly was collected. Some characters could not be written accurately because the titles were written on the image part of the pebbly. But the quality of the OCR tool is sufficient when experts compare the extracted text with the corresponding pebbly. Moreover, the printed media comes from PRNet, a company that provides an archive for printed media with some metadata such as pebbly, OCR tool, title, and date. Unfortunately, the corpus project was started before the hate speech detection part began. The experts annotated some articles and saved the title, date and label without keeping pebbly and link. Since these texts were missing, an article search algorithm was implemented. A time window of 3 days was set for the search. Three days after and before the article publication date were set as the search circle in the large corpus. Then, the jaccard score between the title of the article and other titles was calculated. If the jaccard score was above the threshold (80 %), the article was labelled and the text could be extracted using the OCR tool. Most articles published between 2014 and 2015 were captured by the search mechanism. In addition, some news items that were not in Turkish were removed from the corpus

using a language recognition model.

Each expert examined the news published on their assigned days, selected the news that contained hate speech, and annotated their own targeted group. On the other hand, the same number of news that did not contain hate speech were taken from their assigned days after the news that were labeled with hate speech on that day were removed from the daily set. In addition, the training, validation, and testing sets are determined after the dataset is shuffled and each class distribution is equal and balanced. To test more recent news, a second test dataset was created by selecting the news whose topic is the war between Ukraine and Russia. However, it contains fewer examples than the main set. Also, cleaned versions of news items were added to the dataset that were preprocessed to remove URLs, emails, numbers, currency symbols, punctuation marks, and non-Turkish words.

Table 3.3. Keywords covered by the dataset.

Type	**Keyword**
Ethnicity	afgan (afghan), arap (arab), ermeni (armenian), süryani (assyrian)
	bulgar (bulgarian), çeçen (chechen), çinli (chinese), ingiliz (english)
	çingene (gipsies), yunan (greek), macar (hungarian), iranlı (iranian)
	ıraklı (iraqi), italyan (italian), musevi (jew), yahudi (jewish)
	kırgız (kirghiz), kürt (kurdish), lübnanlı (lebanese), paki (pakistani)
	rus (russian), sırp (serbian), suriyeli (syrian), türkmen (turkmen)
	ezidi (yazidi), ukraynalı (ukrainian), yemenli (yemeni), yezidi (yezidis)
Nationality	batılı (western)
National & religious	alevi (alevist), ateist (atheist), katolik (catholic), hristiyan (christian)
identity	kafir (infidel), gavur (giaour), avrupalı (european), müslüman (muslim)
	ortodoks (orthodox)
Other groups	haçlı (crusader), göçmen (immigrants), mülteci (refugee)
	türbanlı (turbaned)

Table 3.4. Examples of hate speech expression from the dataset.

Example	**Target**
...Avrupalılar önce terörü üretti. Sonra güya kendileri mücadele ediyor... (...Europeans first produced terror. Then they are supposedly struggling themselves...)	Avrupalı (Europeans)
...kalıbımı basarım ki; bunlar "türk" değil, "gavur dölü" taşıyan "piç"lerdir!... (...I swear that these are not "Turks", but "bastards" carrying the "infidel seed"...)	gavur (infidel)
...yunanlılar kaşınıyor! karasularımıza girdiler... (...The Greeks are asking for trouble! They have infiltrated our territorial waters...)	Yunan (Greeks)

In methodology, experts have followed the rules below:

- Hate speech targeting groups (Table 3.3) must be annotated. Hate speech that targets specific individuals and the news containing hate speech in quotes must not be labelled.
- News including exaggeration/attribution/distortion category, which involves cases where the news text exaggerates or distorts information to create a negative perception or uses a singular event to generalize a negative trait to the entire target group, must be annotated.
- News that contains symbolization where hate speech is indexed by association of the race word/target word with negative attributes must be annotated.
- News including hostility and war discourse, where the news text uses direct insults, makes life threats, or supports oppression of a targeted group, must be annotated.

Table 3.5. The number of news which contains hate speech discourse against the specific group.

Keyword	**# of news**	**Keyword**	**# of news**
afgan (afghan)	48	sırp (serbian)	150
arap (arab)	295	suriyeli (syrian)	2333
ermeni (armenian)	2144	türkmen (turkmen)	9
süryani (assyrian)	1	ukraynalı (ukrainian)	23
bulgar (bulgarian)	45	ezidi (yazidi)	3
çeçen (chechen)	5	yemenli (yemeni)	0
çinli (chinese)	34	yezidi (yezidis)	3
ingiliz (english)	408	batılı (western)	545
çingene (gipsies)	3	alevi (alevist)	26
yunan (greek)	1111	ateist (atheist)	57
macar (hungarian)	10	katolik (catholic)	10
iranlı (iranian)	15	hristiyan (christian)	532
ıraklı (iraqi)	21	avrupalı (european)	55
italyan (italian)	30	gavur (giaour)	94
musevi (jew)	1667	kafir (infidel)	26
yahudi (jewish)	868	müslüman (muslim)	771
kırgız (kirghiz)	2	ortodoks (orthodox)	3
kürt (kurdish)	218	haçlı (crusader)	51
lübnanlı (lebanese)	2	göçmen (immigrants)	118
paki (pakistani)	19	mülteci (refugee)	18
rus (russian)	242		

4. METHODOLOGY

After proposing the transformer architecture [16], the use of the transformer-based architecture is the defacto standard in all downstream NLP tasks. This type of architecture can be quickly tuned to user-defined problems, converges successfully, and is more robust than models trained from scratch. For example, classification tasks with a small dataset for a sentiment analysis problem can be easily trained using the BERT encoder in a specific language. Multilingual BERT models have also been published that are trained on texts from multiple languages, but they are not as successful as the pre-trained models that are specified to only one language. For our study, the BERT model [29] trained on Turkish texts is preferred, and all proposed models use the encoder of this model. The fine-tuning phase of a classification task using the BERT model is shown in Figure 4.1.

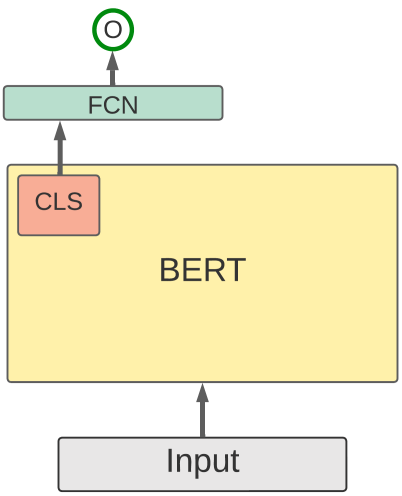

Figure 4.1. Architecture of baseline BERT model. O represents the output of classification (self-drawn).

The BERT encoder creates the hidden attention vector of the input. In the first step, the tokenizer creates an index for each token and constructs a vocabulary. In other

words, the input sentence is represented by a set of token IDs. Then, the token embedding layer generates the embedding of each token, obtaining it from the separately trained embedding model. Then, the position encodings are added with the token embeddings to create the final input representation. Thus, it contains both token and position information.

Furthermore, the aim of the encoder module is learning the relationship among embeddings. For this reason, self-attention layers is defined which learns the attention score for every pair of tokens (including itself). Hence, encoder creates hidden vector as an output that captures the relations between all input embeddings. In other words, context of the input text is represented by the hidden vectors. Thus, the hidden vectors can be utilized for all downstream NLP tasks. Also, BERT has some special tokens such as CLS, SEP (seperator token for multiple sentences), and PAD (padding token). CLS is the special classification token which is put for detecting the attention scores with all input tokens. Although the hidden vector of the CLS token is used for custom classification tasks, the hypothesis of using other hidden vectors are not investigated in previous studies.

Furthermore, hidden vectors of the encoder can be supported by auxiliary features for finetuning custom tasks. Although lexical features have been used in previous studies with the BERT architecture, the integration of target-oriented linguistic features is another important issue to obtain a more successful model.

For this reason, WeBERT and AttWeBERT are proposed to analyze the effectiveness of weighting multiple hidden vectors of the encoder. RuleBERT is developed for feeding the target-oriented linguistic features.

4.1. WeBERT

Attention can be integrated with different layers in the model. In this study, all attention layers learn the importance coefficient of the hidden vectors of the encoder,

which is different from the self-attention layer in the encoder. The self- attention layer calculates the attention for all possible pairs of vectors, while the proposed attention layers focus on weighting the hidden vectors to create a weighted hidden vector.

WeBERT uses a soft attention layer that computes the weights of each hidden vector. The model architecture is shown in Figure 4.2. The fully-connected layer, the ReLU layer, the dropout layer, and the softmax layer are used in the weighting layer, respectively. The fully-connected layer processes multiple hidden vectors and constructs a latent hidden vector. The ReLU activation function and the dropout layer are each applied to the latent hidden vector to reduce the risk of overfitting.

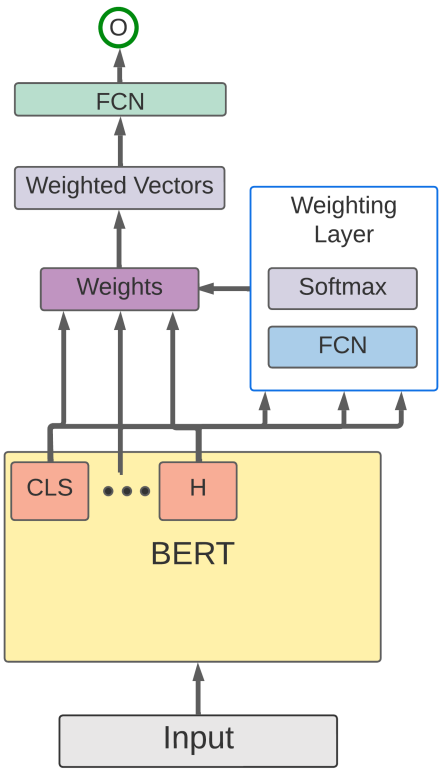

Figure 4.2. Architecture of WeBERT model. O represents the output of classification (self-drawn).

The final weights are obtained from the softmax layer. The softmax layer generates an attention probability for each hidden vector. The hidden vectors are multiplied by the softmax probabilities and summed to obtain the final hidden vector. The final fully-connected layer has the same parameters as the baseline model. The formula for finding the final hidden vector is as

$$\alpha_i = \frac{exp(h_{FCN_i})}{\sum_{j=1}^{n} exp(h_{FCN_j})} \tag{4.1}$$

$$h_{final} = \sum_{i=1}^{n} \alpha_i.h_i, \tag{4.2}$$

where h_{FCN_i} corresponds to ith hidden vector of the weighting layer, h_i corresponds to ith output of the BERT encoder, and h_{final} is final hidden vector after attention scores are applied.

4.2. AttWeBERT

AttWeBERT uses different layers to apply attention to hidden vectors. It utilizes encoder-decoder based architecture for attention weighting layer. The gating function learns the importance of each hidden vector. The GRU encoder [15] encodes the hidden vectors, multiplied by the importance value, and the output of each time step is fed into the Attention decoder layer [31], where their attention weights are learned. The context vector containing the attention values is multiplied by all the outputs to obtain a weighted vector. Moreover, the class embedding layer [48] finds the association between hidden vectors and classes.

Also, the first N hidden vector of the BERT encoder is selected. The intuition behind this logic is that using the vector containing the correlation between different tokens, except the CLS token, can capture more information for better classification. Also, N multiple hidden vectors can be randomly selected from the set, but this has the disadvantage of increasing the stochasticity in analysing the behaviour of the model. Moreover, the main objective of this research is to test the effect of using multiple vectors instead of using only the hidden vector of the CLS token.

The details of each module used in attention weighting layer and class embedding layer are explained in following sections. The model architecture is shown in Figure 4.3.

4.2.1. Gating Function

The gating function consists of a fully-connected layer (FCN) using the time-distributed approach and a sigmoid function. The sigmoid function determines the importance of each hidden vector after it is fed into the FCN. It is formulated as

$$c_i = \sigma(W_{FCN}.h_i) \tag{4.3}$$

$$h'_i = c_i.h_i \tag{4.4}$$

where h_i denotes ith hidden vector and c_i is the corresponding attention value.

4.2.2. GRU Encoder

The GRU encoder applies one directional GRU layer that has one layer and no dropout. It produces an output whose size is 128 for each time step. The layer is formulated as

$$out_i = GRU(h_i, out_{i-1}). \tag{4.5}$$

4.2.3. Attention Decoder

The Attention decoder follows an attention method developed by [31]. It takes the start token, which is CLS, the last hidden state, and the outputs of the GRU encoder as inputs. The GRU layer, which is initialized by the last hidden state and has one layer and 0.1 dropout, is applied to the embedding vector of the start token and obtains the outputs of each time step in order to use in the calculation of attention. It is formulated as

$$dec_i = GRU(E_{[CLS]}, out_n) \tag{4.6}$$

where $E_{[CLS]}$ denotes the embedding of the start token and out_n is the last hidden state of the GRU encoder.

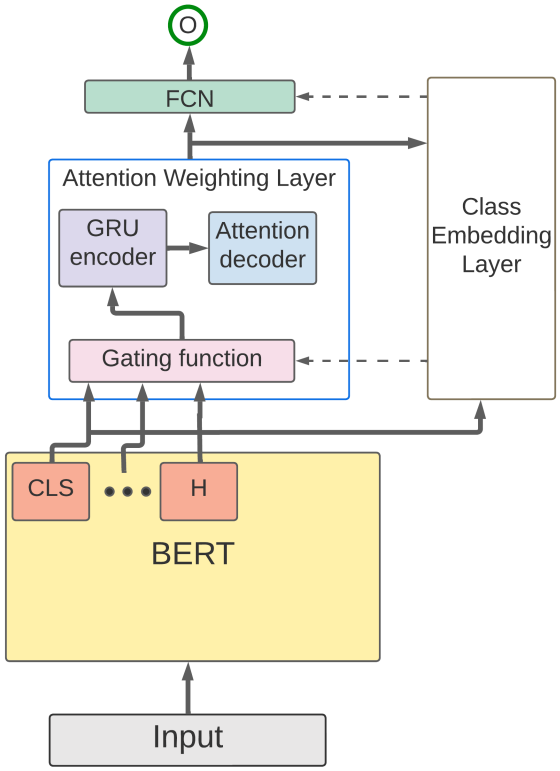

Figure 4.3. Architecture of AttWeBERT model. O represents the output of classification (self-drawn).

The context vector, including attention weights, is computed by matrix multiplication between the outputs of the encoder and those of the decoder. The softmax layer is applied to this vector. In other words, this layer detects the association of which timestamp produces the most useful outputs in both the GRU encoder and the GRU decoder. The outputs of the encoder are multiplied by the timestamp weighting probability to create a context vector. The formula is as

$$\alpha_i = \frac{exp(out_{dec_i}.out_{enc_i})}{\sum_{j=1}^{n} exp(out_{dec_j}.out_{enc_j})} \tag{4.7}$$

$$w_i = \alpha_i.out_{enc_i} \tag{4.8}$$

where out_{enc_i} corresponds to ith output of the GRU encoder, out_{dec_i} corresponds to ith output of the GRU decoder, and context vector consists of w_i for all outputs of the GRU encoder.

In a final step, the weighted outputs of the encoder and outputs of the decoder are concatenated and fed into the FCN layer and the tanh activation function, respectively, resulting in the final weighted vector of the GRU decoder. The final vector is obtained from the following formula as

$$v_{weighted} = tanh(FCN(out_{dec} \oplus w_{context})). \tag{4.9}$$

4.2.4. Class Embedding Layer

This layer finds the relationship between the class vector of each hidden vector and the classes. It contains an embedding matrix that produces a 100-dimensional embedding of class vectors. The class vector is generated by feeding the hidden vector into the FCN or ReLU activation layer. It is formulated as

$$e_i = W_{emb}.FCN(h_i) \tag{4.10}$$

where e_i denotes class embedding vector of h_i.

4.3. RuleBERT

RuleBERT is designed as a BERT based model supported by target-oriented linguistic features that serve as additional indicators for hate speech detection. The architecture is demonstrated in Figure 4.4. Although there are no custom layers in this model as in AttWeBERT, it is the best model in all experiments.

Linguistic features are defined as target-oriented because the order of use of target group and hateful word, rules that consider only ethnicity, and special patterns that capture hate speech for a particular target group are important reasons for identifying

target-oriented features. Thus, they are formulated by analyzing a corpus that does not contain data used in the experiments. In other words, there are no leaks in the formulation of the linguistic features.

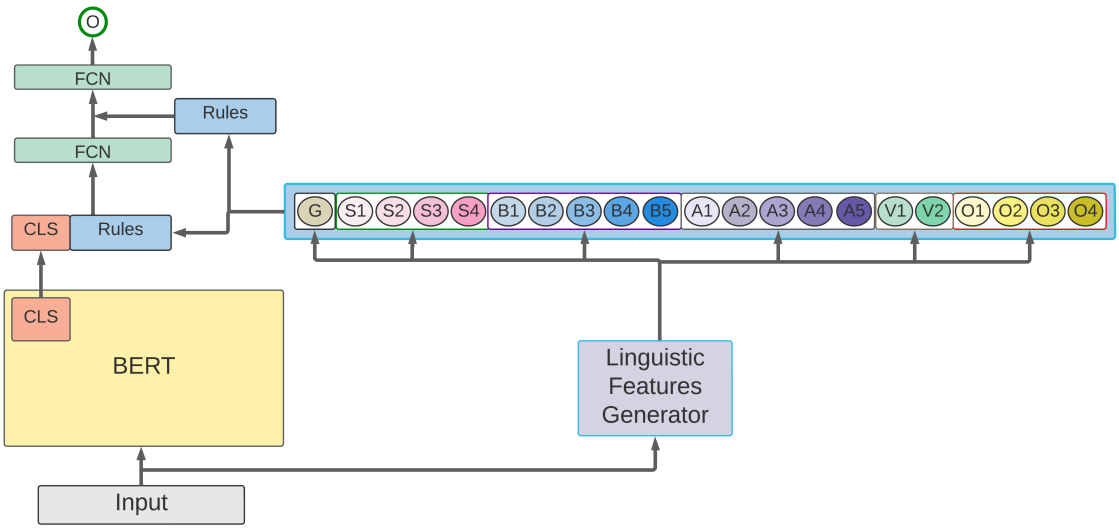

Figure 4.4. Architecture of RuleBERT model. O represents the output of classification (self-drawn).

They are classified into 6 different types. They are explained in detail in the following sections. These types are determined on the basis of some criteria. Each type has common characteristics in terms of feature formulation, hate speech discourse search methodology, and semantic expression. Moreover, except for the general type, they are divided into several subtypes. The number of subtypes and their scope are determined by linguistics experts based on the severity of hate speech. Degree 1 represents the least severe, while Degree 5 represents the most severe hate speech. The details of the subtypes are listed in Table 4.1.

In addition, the feature generation method is listed in Table 4.1. The methods used are one-hot encoding and cumulative values, where each subtype is represented as a separate dimension, generated, and fed to the RuleBERT model. The details of

the feature generation method are explained in Section 4.3.7.

Table 4.1. Subtypes and feeding method of target-oriented linguistic features. The subtypes indicate the severity of the hate speech. The cumulative value is a vector with a value that accumulates the number of detected rules in a given feature type. The dimension of the one-hot vector is determined by the number of subtypes.

Type.	**Num Subtypes**	**Feature Method**
General	No	Cumulative value
Special	4	One-hot vector
BeforeTarget	5	One-hot vector
AfterTarget	5	One-hot vector
HSPointedVerb	2	One-hot vector
AntiHate	4	One-hot vector

In addition, the number of features for each type is different. For example, the After-Target type has the most hate speech patterns. Each type also has a window parameter where the feature is searched in the article. The window parameter is important because some feature patterns may contain more than one word and may not necessarily be used respectively. As shown in the examples in Table 4.2, the pattern "ETHN who are puppets of OTHER ETHN " is searched for each 15-word window of the article, and all keywords in the ethnicity group are searched for. Some properties and examples are shown in Table 4.2.

4.3.1. General Features

General features aim to identify commonly used patterns for all ethnicity and nationalities in news articles. These 14 rules are used to formulate hate speech patterns that are preferred by authors for almost all target groups. They are searched in a

window of 15 words using different patterns, adjectives and nouns that are common in hate speech. Each rule contains the keyword ETHN, which replaces any keyword in the ethnicity group. For example, if an article contains the words “look at that Greek” in a given window, it is recognised using the rule ID-2. If the article contains the words “Jewish largest and destructive invasion”, it is detected by the rule ID-5. The group of words “largest and destructive” is not an obstacle to the detection of the rule due to the 15-window feature. Formulation of all general rules are written in Table 4.8.

Table 4.2. Structure of the generated linguistic features. ETHN (IRK) substitutes for any ethnicity in Table 3.3. Count indicates the number of patterns used in feature generation. Window determines in which window range the pattern is searched. No window indicates that patterns must be searched by strict matching.

Type	**Count**	**Window**	**Example**
General	14	15	IRK+ın kuklası olan IRK (ETHN who are puppets of OTHER ETHN)
Special	19	No	kripto ermeni (crypto armenian)
BeforeTarget	38	No	istilacı IRK (invading ETHN)
AfterTarget	60	No	IRK soykırımı (ETHN massacres)
HSPointedVerb	8	15	IRK ... çizmeyi aştı (NATIONALITY ... overstepped the mark)
AntiHate	30	15	IRK fanatiği (ETHN fanatic)

Rules have also been created to detect hate speech about two different ethnicities. Therefore, articles dealing with historical events of two different ethnicities can be detected by appropriate rules. For example, “ETHN backed OTHER ETHN attacks” is a common pattern in Turkish printed media because some media groups try to make society see a certain target group as the reason for the disaster. “Greek backed Armenian attacks” is one example pattern used in the news.

General linguistic features are not represented by a one-hot vector to feed the model because they have no severity degree (subtype). For this reason, it is represented by a cumulative value, where the number of detected rules represents the value of the general linguistic feature. It is normalized between 0 and 1 to avoid the problem of overfitting in neural networks. For example, if 7 rules are observed in a news article, the value of the general feature is 0.5. In addition, the patterns in the BeforeTarget and AfterTarget feature types are typically used ID-14 to capture more general uses of hate speech in the articles.

4.3.2. Special Features

Special features are created to detect target group specific patterns generally associated with a particular group in news. There is no window property for searching the patterns. Thus, each pattern should be strictly matched in given order of words. All special patterns are listed in Table 4.3.

In addition, there are subtypes for particular characteristics that correspond to the severity of the hate speech. For example, the pattern "western ignorance" is not as harmful as "killer rum" because "killer" defines the target group more hatefully, while "western ignorance" can be used for historical realities based on some previous research.

Almost all patterns include a specific target group. For example, "crusader mentality" is a pattern deliberately defined by the authors but using the word "mentality" does not become a characteristic hate speech pattern for other target groups due not to find in large corpus.

4.3.3. BeforeTarget Features

BeforeTarget features are organized for adjacent words used right before the target keywords to describe the groups by carefully selected hate words. All adjacent

features are based on direct pattern matching (no window parameter) and are grouped according to their own severity. For example, the pattern “thief” should be found directly before the target group in the article. Although some patterns can be considered to use right after the target group, this issue exists due to the fact that they are written in English. In Turkish, the “hırsız” pattern is usually used before the target group, e.g. “hırsız yunan”, “hırsız mülteci”. All patterns are listed in Table 4.6.

Table 4.3. Patterns used in special type of linguistic features. Higher degree points more serious hate speech discourse.

Degree 1	**Degree 2**	**Degree 3**	**Degree 4**
vahşi toplumlar	batıl batı	yahudi ajanı	yahudi çakallığı
(wild societies)	(superstitious west)	(jewish agent)	(jewish cowardice)
batı cehaleti		yahudi uşağı	katil rum
(western ignorance)		(jewish servant)	(killer rum)
haçlı zihniyeti		kripto ermeni	haydut rumlar
(crusader mentality)		(crypto armenian)	(rogue Greeks)
kriptolar		suriyeli işgali	rum zorbalığı
(cryptos)		(syrian invasion)	(Greek bullying)
		afgan işgali	kafir alevi
		(afghan invasion)	(infidel alevist)
		mülteci işgali	ateist alevi
		(refugee occupation)	(atheist alevist)
		pakistanlı işgali	
		(pakistani invasion)	
		arapların işgali	
		(invasion of the arabs)	

4.3.4. AfterTarget Features

AfterTarget features consist of adjacent words written immediately after the target keywords. They indicate the specific hate speech discourse that the authors are trying to popularize in the news. For instance, the pattern "games" is often used directly after the target group in articles where "Greek games" and "Jewish games" are obvious examples. Listing the patterns in English, which presents the same problem as the BeforeTarget features, does not affect the results because all of these patterns are recognized by their Turkish form in the articles.

The AfterTarget group also has subtypes which indicate the severity of hate speech. They are categorized by linguistic experts. All patterns are listed in Table 4.7.

4.3.5. HSPointedVerb Features

HSPointedVerb features are commonly used verbs in hate speech discourse. They must be found in a 15-word window after the target headwords. In the large corpus, it is clear that the majority of these verbs are used in the news article title because they can attract readers' attention and easily spread the hate speech. For example, when the word "turn somewhere into bloodbath" is used in connection with a specific target group, the target group is associated with some destructive events. In Turkish print media, news articles with titles similar to "Yunan Ege'de ortalığı kana buladı" are the most common. All frequently used verbs are listed in Table 4.4.

4.3.6. AntiHate Features

Anti-Hate features are misleading patterns used in the news about target groups, but usually do not express hate speech. They are created by detecting specific patterns of a word and the words used after the target keyword or before the target keyword. All anti-hate speech patterns are listed in Table 4.5.

Table 4.4. Frequently used verbs in hate speech discourse. They are detected when they are used with one of target in given window range. Higher degree points more serious hate speech discourse.

Degree 1	**Degree 2**
kaşınıyor (let oneself in for something)	kana buladı (ensanguine)
haddini aştı (go beyond the limit)	kan gölüne çevirdi (turn somewhere into bloodbath)
çizmeyi aştı (overstep the mark)	
şımardı (be spoilt)	
kudurdu (see red)	

In addition, some news articles reflect some historical events that are accepted by all people. However, they may be classified as hate speech by the model if the model is not able to evaluate them in the correct time period. Therefore, "today in the calendar" and "today in history" are important AntiHate patterns to distinguish whether they are current events or not.

Also, articles can quote some hateful speeches of politicians. According to the methodology of this study, this article should not be evaluated as hate speech because the content of hate speech is expressed by another source. For this reason, the patterns "expressed as" and "explained as" are included in the AntiHate patterns.

4.3.7. Model Architecture

In the RuleBERT architecture, all target-oriented linguistic features are fed in after they are represented by a vector. All vectors of the different types are normalized between 0 and 1. Some feature types have no subtypes such as the general type, the others have several ones. For this reason, they should be defined in different ways to generate inputs to neural networks in correct form.

For the feature type that has no subtypes, the cumulative value is used. The number of detected patterns is counted and normalized to the total number of patterns of that type. Therefore, a vector that has a value is placed as part of the vector of all rules. An example is shown in Figure 4.4, where a circle with the character "G" indicates the cumulative value of the general features (Section 4.3.1).

The one-hot vector is a vector representation commonly used in machine learning when the feature values are the discrete elements of a category. In this study, all subtypes indicate a category, so feature types with subtypes are represented by a one-hot vector. Each dimension of the one-hot vector corresponds to whether or not at least one pattern is used in that subtype. If one or more than one pattern is used, the value of this dimension is assigned as 1.

4.4. OnlyRule Baseline Model

All target-oriented linguistic features are fed into the RuleBERT architecture as auxiliary features. However, they are very strong when it comes to expressing the hate speech in the articles. For this reason, the OnlyRule model is proposed, which is a second baseline that uses target-oriented linguistic features. OnlyRule is a simple neural network that includes 2 fully connected layers, batch normalization and dropout layers. It also uses the nonlinear activation function ReLU.

Moreover, in this model, the same linguistic feature vector is used only for input, and texts and embeddings of tokens are not fed into the OnlyRule model. For this reason, the model tries to learn whether using only linguistic features can detect the hate speech or not.

Table 4.5. Misleading hate speech discourse that are found mostly non hate speech news. ETHN (IRK) substitutes for any ethnicity to generate feature. A higher rating indicates less or no hate speech. "ETHN+ist" expresses ethnicity names and the suffix "-CU" in Turkish, which is derived nationalist names from it by attached them (e.g. Türkçü, Kürtçü).

Degree 1	**Degree 2**	**Degree 3**	**Degree 4**
IRK çeteleri	haçlı seferi	diye belirtti	futbol
(ETHN gangs)	(crusade)	(s/he stated)	(football)
IRK fanatiği	STK	dedi	spor
(ETHN fanatics)	(non-governmental organizations)	(said)	(sport)
pontus rum	tarihte bugün	şeklinde açıkladı	maç
(pontus empire)	(today in history)	(expressed as)	(match)
IRK polisi	takvimde bugün	şeklinde ifade etti	antik yunan
(ETHN police)	(today in the calendar)	(explained as)	(ancient greek)
IRK yaygaracılığı			yunan düşünür
(ETHN fuss)			(greek thinker)
IRK askerleri			yunan filozof
(ETHN soldiers)			(greek philosopher)
IRK milisleri			
(ETHN militia)			
IRK militanları			
(ETHN militants)			
IRK yerleşimciler			
(ETHN settlers)			
IRK milliyetçiler			
(ETHN nationalists)			
IRK güçleri			
(ETHN forces)			
IRK isyanı			
(ETHN revolt)			
radikal IRK			
(radical ETHN)			
ırkçı IRK			
(racist ETHN)			
siyonistler			
(zionist jew)			
IRK+cı terör örgütü			
(ETHN+ist terrorist organization)			

Table 4.6. Adjacent words that written right before keywords in news. Higher degree points more serious hate speech discourse.

Degree 1	**Degree 2**	**Degree 3**	**Degree 4**	**Degree 5**
sapıtan	katleden	kripto	işgalci	hain
(amok)	(murderer)	(crypto)	(invader)	(traitor)
çakma	korkak	sinsi	gaspçı	katleden
(fake)	(coward)	(sly)	(grabber)	(murderer)
facir	yamyam	açgözlü	lanetlenmiş	gavur
(sinner)	(cannibal)	(greedy)	(damned)	(infidel)
gizli	başbelası	dönek	Allah'ın lanetlediği	kalleş
(hidden)	(the very devil)	(renegade)	(cursed by god)	(treacherous)
kışkırmış		iki yüzlü	zalim	kan gölüne çeviren
(spoiled)		(two-faced)	(cruel)	(vicious killer)
hırsız		azgın	şerefsiz	eli kanlı
(thief)		(ferocious)	(dishonourable)	(bloody hand)
		edepsiz	gaddar	bebek katili
		(shameless)	(grim)	(baby murderer)
		yağmacı	gasıp	cani
		(predatory)	(usurper)	(villain)
		çapulcu	canavarlaşmış	vahşi
		(marauder)	(monstrous)	(wild)
				insanlık suçu işleyen
				(perpetrator of crimes
				against humanity)

Table 4.7. Adjacent words that written right after keywords in news. Higher degree points more serious hate speech discourse.

Degree 1	**Degree 2**	**Degree 3**	**Degree 4**	**Degree 5**
işbirlikçisi	yalanları	baskısı	terörü	gaddarlığı
(collaborator)	(its lies)	(pressure)	(province of terror)	(atrocity)
inadı	iftiraları	bozma	terör üssü	imha
(stubbornness)	(its slanders)	(violation)	(terror base)	(destruction)
doyumsuzluğu	tehdidi	yalakalığı	saldırıları	zulmü
(dissatisfaction)	(threat)	(fawning)	(attacks)	(cruelty)
karısı	oyunu	entrikaları	terörizmi	kırımı
(his/her wife)	(games)	(intrigues)	(terrorism)	(politicide)
dolandırıcı	yağmacılar	fesatları	sapkınlığı	vahşeti
(swindler)	(looters)	(mischief)	(heresy)	(brutality)
parmağı	çapulcusu	sürüleri	köpekler	zalimi
(his/her finger)	(marauder)	(herd)	(dogs)	(ferocity)
provakasyonu	haydutlar	kötülükleri	terör örgütü	hain
(provocation)	(bandits)	(their evil)	(terrorist organization)	(traitor)
artığı		dönekliği	sırtlanlar	kalleşliği
(reversion)		(apostasy)	(hyenas)	(treachery)
uşaklığı		açgözlülüğü	soysuzlar	canilikleri
(servitude)		(greed)	(retrograde)	(murderousness)
aşığı		sinsiliği	çakallar	kıyımları
(lover)		(snakiness)	(coyotes)	(massacres)
gaspçılar		yüzsüzlüğü	yamyamlar	piçleri
(usurpers)		(sassiness)	(cannibals)	(bastards)
kuklası		iki yüzlülüğü	vandallar	
(puppets)		(hypocrisy)	(vandals)	
piyonları		baskını		
(pawns)		(raid)		
teröristi		tohumu		
(terrorists)		(seed)		
virüsü		dölleri		
(virus)		(sperm)		

Table 4.8. General patterns that are widely used in hate speech discourse. They were extracted from large corpus by experts. ETHN (IRK) substitutes for any ethnicity to generate feature. Patterns are searched in determined window range. BEFTAR corresponds that words in Table 4.6 can be placed. AFTERTAR indicates that words in Table 4.7 can be put.

Id	**Pattern**
1	IRK+lık yapmak (act like ETHN)
2	IRK'a bak sen (look at that ETHN)
3	IRK IRK+lığını yap(mak) (s/he does her/his ETHN)
4	IRK kurşunlarıyla/bombalarıyla/parasıyla (with the bullets/bombs/money of ETHN)
5	IRK paryası/skandalı/işgali/baskını (ETHN pariah/scandal/occupation/invasion)
6	IRK+ın gerçekleştirdiği/yaptığı katliam/zulüm/soykırım (massacre/persecution/cruelty/oppressure of ETHN)
7	IRK+ın uşağı/işbirlikçisi/piyonu/kuklası (olan) IRK (ETHN servant/pawn/collaborator/puppets of OTHER ETHN)
8	IRK destekli IRK darbesi/saldırıları/katliamı/soykırımı (ETHN backed OTHER ETHN coup/genocide/massacre/attacks)
9	IRK tarafından saldırıya/katliama/soykırıma maruz kalmak/uğramak (being attacked/subjected to genocide/massacred by the ETHN)
10	IRK tarafından gerçekleştirilen/yapılan katliam/zulüm/soykırım (massacre/persecution/cruelty/oppressure done/carried out by ETHN)
11	IRK ... öldürdü/katletti/etnik temizlik yaptı/kirletti/bastı/şehit etti (ETHN ... killed/massacred/did ethnic cleansing/disgloried/martyrized)
12	IRK tarafından ... öldürüldü/katledildi/etnik temizlik yapıldı/basıldı/şehit edildi (killed/massacred/did ethnic cleansing/disgloried/martyrized ... by ETHN)
13	IRK tarafından IRK+a yönelik saldırılar/katliam/zulüm/soykırım (genocide/massacre/persecution/cruelty/oppressure/attack of ETHN by OTHER ETHN)
14	IRK+ın hain(ce)/vahşi(ce)/insanlık dışı/hunharca/kan donduran/şeytani/sinsi/BEFTAR teşebbüsleri/planları/oluşumları/AFTERTAR ETHN's sneaky/traitorous/wild/subhuman/bloodthirstily/terrifical/satanic/BEFTAR attempts/plans/organizations/AFTERTAR

5. EXPERIMENTS AND RESULTS

In this study, hate speech detection is a kind of binary classification task. In classification tasks, accuracy, precision, recall, and F1-score are defacto metrics for evaluating models. Although F1-score is more important than the others in classification tasks trained on an imbalanced dataset, the dataset in this study is balanced, so all metrics are important and accurately express the performance of the model.

In addition, specifying model hyperparameters and the details of experiments are as important as training the best model in research, different combinations of parameters in training strategies are the key point of experiments in machine learning. Also, setting a random seed is very important to get robust evaluation results. For this reason, the random seed is set to the same value for all experiments. Details of the experiments are given in Section 5.1.

For a fair evaluation of the models, the design of the test scenarios is as important as the definition of the evaluation metrics. For this reason, the details of the test sets and model performances are presented in Section 5.2. The question of which modules provide the best result is also an important criterion for evaluating the implementation of new layers in the study. For example, developing a new encoder-decoder attention module may improve the results, but the contribution of the decoder attention layer cannot be determined without performing an ablation study for that layer. Therefore, different scenarios are planned for the ablation study, which are explained in Section 5.3. Finally, RuleBERT errors, possible causes, and test examples that were misclassified by the baseline model but corrected by RuleBERT are examined in the Error Analysis section. They are explained in Section 5.4.

5.1. Experiment Details

For all experiments, Turkish BERT model [29] is used as an encoder. Target-oriented linguistic features and attention layers are integrated in this encoder module for models. All hyperparameters are selected based on performance in the validation set. The learning rate, the number of GRU layers, the option for bidirectional recurrent layers, the number of units in each GRU layer, the number of hidden units in fully-connected layers, and the optimizer are important parameters for model training. Selected hyperparameters are listed in Table 5.1.

Table 5.1. Best combination of hyperparameters of AttWeBERT model.

Parameter	**Value**
Learning rate	1e-5
Weight decay	1e-2
Optimizer	AdamW [49]
# of layers in GRU encoder	1
# of hidden units in GRU encoder	100
# of layers in GRU decoder	1
# of hidden units in GRU decoder	100
Prop. of dropout in GRU decoder	0.1
Non linear activation function	ReLU

The F1 score is determined to evaluate the success of the models in the validation set. The validation performance is evaluated for every 500 steps. BERT (Figure 4.1), WeBERT (Figure 4.2) and RuleBERT (Figure 4.4) are trained with 2 epochs, AttWeBERT (Figure 4.3) is trained with 4 epochs. AttWeBERT needs more epochs since it has more layers to converge. For all experiments, the AdamW optimizer [49] is used, whose learning rate is 1e-5 and weight decay is 1e-2.

The scheduler for the learning rate is integrated, where the patience is 2 evaluation steps and the reduction factor is 0.5. All fully-connected blocks in all proposed models consist of a dropout layer [50] whose probability is 0.5.

The models are implemented using Pytorch 1.11.0, Huggingface Transformers 4.18.0, and clean-text 0.6.0. The experiments are performed on a computer with an NVIDIA GeForce RTX 3080.

5.2. Results

Table 5.2 shows the performance of all models in terms of accuracy, precision, recall, and F1 score. The results are given for two different test sets. The test set for all keywords contains the news published about all target groups included in the annotation methodology of HDF [51]. The test set for Ukraine news contains only the news written about the war between Ukraine and Russia. RuleBERT outperforms the other models on all metrics on the test set for all keywords, despite having fewer parameters than AttWeBERT and WeBERT. Moreover, together with AttWeBERT, it achieves the best results for the Ukraine news test set. The reason of obtaining closer scores in Ukraine news among the models is that Ukraine news contains fewer articles than the all keywords news.

AttWeBERT and WeBERT outperform the baseline BERT model and baseline OnlyRule model in terms of accuracy and F1-score on both test sets. Although their approach for integrating attention into the architecture is different, they achieve a very similar performance on the test set for all keywords. However, AttWeBERT outperforms WeBERT on the Ukraine news.

All results show that the use of target-oriented linguistic features improves the performance of the model and reduces the number of false negative examples compared to the baseline model BERT. Thus, the number of unrecognised articles containing hate speech can be reduced thanks to RuleBERT when using hate speech detection systems.

The results in Ukraine show that target-oriented features are not only beneficial for the keywords used in the training data, but also increase the performance of the model in the test set with different target keywords. For this reason, target-oriented linguistic features can be generated in both the training and validation and testing phases without anchoring the list of targets.

Table 5.2. Evaluation of various models over different test sets. All keywords news includes news that are written about all target keywords. Ukraine news contains only the ones published about the Ukraine war. w/ 4 hidden vectors indicates the model weighting 4 attention hidden vectors and w/o Att indicates the one without using Attention decoder. The scores are represented as percentages (%).

Model	All keywords news				Ukraine news			
	Accuracy	Precision	Recall	F1-score	Accuracy	Precision	Recall	F1-score
OnlyRule	56.08	55.71	65.66	60.28	53.34	52.00	86.67	65.00
BERT	90.23	90.60	89.88	90.24	90.00	98.20	80.21	88.92
AttWeBERT	90.43	89.93	91.16	90.54	**96.66**	**98.42**	**93.33**	**96.62**
w/ 4 hidden vectors	89.79	88.70	91.34	89.99	86.67	98.16	73.34	84.62
w/ 10 hidden vectors	89.68	91.30	87.81	89.52	83.33	98.31	66.67	80.02
w/o cls emb	89.79	89.00	90.92	89.95	96.62	98.40	93.31	96.59
w/o Att	90.11	88.59	92.19	90.36	93.33	98.24	86.67	92.86
WeBERT	90.51	90.59	90.52	90.55	93.33	98.24	86.67	92.86
RuleBERT	**91.39**	**90.69**	**92.35**	**91.51**	**96.66**	**98.42**	**93.33**	**96.62**

The statistical test is performed to check the predictions of the model for disagreement. In other words, the goal is to test the hypothesis that BERT and the RuleBERT model disagree in the same way, which is the null hypothesis. For this reason, McNemar's statistical test is chosen, which is paired and non-parametric (distribution-free). The statistical value and p-value of the test are calculated as 6.82 and 0.009, respectively. Consequently, the null hypothesis is rejected and the result of the test confirms that there is a statistically significant difference in the disagreements between the predictions of BERT and the predictions of RuleBERT.

5.3. Ablation Study

Ablation experiments are performed for AttWeBERT to investigate the contribution of the different modules to the weighting of several hidden vectors and performance metrics of the different ablation models are listed in Table 5.2. First, the impact of the number of hidden vectors used in the attention module is investigated by setting it as 4 and 10, while AttWeBERT has 20 hidden vectors. The results show that using 20 hidden vectors performs better than using a smaller number of hidden vectors in terms of accuracy and F1 score. However, AttWeBERT with 4 hidden vectors achieves better recall, accuracy, and F1 score than the model with 10 hidden vectors in terms of accuracy and F1 score.

Moreover, the class embedding layer is removed. The results show that AttWeBERT with class embedding layer increases the performance on all metrics. Finally, AttWeBERT without the attention decoder module was tested to determine whether hidden vector weighting increases performance. AttWeBERT with attention decoder performs better in all metrics on both test sets.

5.4. Error Analysis

A randomly chosen subset of the results were analyzed qualitatively to compare BERT and RuleBERT. We observed that RuleBERT had improvements in capturing patterns such as “Ermeni uşağı” (servant of Armenians) in ID-7 general rule of Table 4.8. Additionally, many false positive examples were prevented, presumably via the addition of limiting phrases such as “terrorist organization” to the AntiHate set in Table 4.5. Hence, utilizing target-oriented linguistic features both improves the metrics by decreasing both the number of false positive and false negative predictions.

Further analysis of RuleBERT’s errors revealed that the absence of punctuation marks and the lack of importance given to article titles and subtitles lead to an increase in false negatives. Unfortunately, the training datasets of the pre-trained model BERT

are based on texts without punctuation marks, so the model cannot use them directly. On the other hand, these marks can be defined as a new token in fine-tuning, but the dataset must be large enough to learn the relationship between them and the other tokens. For this reason, this experiment is not performed because the size of our dataset is limited.

In addition, the model has a bias for hate speech in articles that cite historical events and contain words that are commonly used in hate speech. Moreover, figurative expressions that are not used in the training set but occur in the test set confuse RuleBERT.

6. CONCLUSION AND FUTURE WORK

In this study, the effects of utilizing target-oriented linguistic features and weighting multiple hidden vectors of BERT for hate speech detection are investigated. The RuleBERT model is proposed to demonstrate the contribution of linguistic features, and the WeBERT and AttWeBERT models are proposed to analyze the importance of using attention modules to weight hidden vectors. Experiments on a comprehensive dataset of Turkish print media show that all proposed models lead to improved performance, with RuleBERT achieving the best results, suggesting that incorporating target-oriented linguistic knowledge into a transformer model is a promising approach for hate speech detection.

A pre-processed Turkish hate speech dataset is also published, containing printed news collected from various sources and annotated by experts. It also contains the identification of the target group against which the authors use hate speech (only for articles labelled with hate speech). For each article, information about the author, source, publication date and title is extracted and the sentences are split and preprocessed in addition to the raw text.

Furthermore, a methodology for target-oriented linguistic features is proposed. In this methodology, linguistic features are classified into different types and their generation method is explained in detail in 4.3. Sub-types of each type are determined, corresponding to the severity of the hate speech. For each type, a window parameter is selected to search hate speech in the token set. The integration of targeted features into the BERT based architecture is demonstrated, where features are represented by either one-hot encoding or an cumulative value that stores the number of detected rules.

Moreover, various attention modules are proposed to intelligently weight multiple hidden vectors. Multiple hidden vectors are selected from the output of the BERT

encoder and weighted by the learned coefficients. Although only a first hidden vector has been used in classification tasks, this study investigates the effect of using multiple hidden vectors, and the results show an increase in overall performance.

On the other hand, this study focuses on printed news, which is published in newspapers rather than on the Internet. Therefore, the language is more formal than social media where more swearing and direct hate speech are used. For this reason, the methods used to generate targeted linguistic set are based on printed news. In addition, this work aims to detect hate speech against ethnicity, national and religious entities, and immigrants, as mentioned in the dataset methodology. For this reason, news articles that contain hate speech but not the target keyword specified in Table 3.3 are not included in the scope of this study. Also, all models were trained with the Turkish news data set, and some of the patterns (words) used in the targeted linguistic features may be specific to Turkish discourse. As future work, this study can be extended to a larger number of keywords and a general hate speech detection model can be proposed that detects hate speech both in printed media and on the Internet.

Given the limitations of the study, it can be enriched and improved by adding various capabilities. For example, a dependency parser module can be beneficial for the target-oriented linguistic features side, since the hate speech patterns are used against a specific group in the sentence and the dependency parser allows the model to find the related subject for which the pattern is used. Moreover, lexical resources can be integrated into the set of linguistic feature patterns to enrich them for better generalization. A sentiment lexicon can also be created for this problem. On the other hand, these lexical resources should be compatible with the definition of the problem. For example, they should not contain only hateful words that are widely used in the spoken language. In addition, morphological analysis can be useful to identify the root word of both target group keywords and other patterns, and to capture all versions of patterns. Although almost all patterns are captured by the complex regex queries, they can also be detected after applying morphological analysis. However, since Turkish is an agglutinative language, another type of error may occur, namely confusion between

state names and target group names. For example, if the root word of Yunanistan (Greece) is found to be Yunan (Greeks), this is contrary to the goal of this study, which is to identify hate speech against specific target groups rather than states. As a final future work, the use of Wikipedia resources to introduce the historical events and their own specific terms as a lexicon. Creating a dictionary that includes the commonly used patterns and special terms for the target groups' historical events can improve the model in terms of false positives.

REFERENCES

1. Schmidt, A. and M. Wiegand, "A Survey on Hate Speech Detection using Natural Language Processing", *Proceedings of the Fifth International Workshop on Natural Language Processing for Social Media*, pp. 1–10, Association for Computational Linguistics, Valencia, Spain, 2019.

2. Mayda, I., Y. E. Demir, T. Dalyan and B. Diri, "Hate Speech Dataset from Turkish Tweets", *Innovations in Intelligent Systems and Applications Conference (ASYU)*, pp. 1–6, Elazığ, Turkey, 2021.

3. Devlin, J., M.-W. Chang, K. Lee and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding", *North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4171–4186, Association for Computational Linguistics, Minneapolis, Minnesota, 2019.

4. Mozafari, M., R. Farahbakhsh and N. Crespi, "A BERT-based Transfer Learning Approach for Hate Speech Detection in Online Social Media", *International Conference on Complex Networks and Their Applications*, pp. 928–940, Springer, Lisbon, Portugal, 2019.

5. Gupta, S., S. Lakra and M. Kaur, "Study on BERT Model for Hate Speech Detection", *International Conference on Electronics, Communication and Aerospace Technology (ICECA)*, pp. 1–8, Tamil Nadu, India, 2020.

6. Mozafari, M., R. Farahbakhsh and N. Crespi, "Hate Speech Detection and Racial Bias Mitigation in Social Media Based on BERT Model", *PloS One*, Vol. 15, No. 8, p. e0237861, 2020.

7. Caselli, T., V. Basile, J. Mitrović and M. Granitzer, "HateBERT: Retraining BERT

for Abusive Language Detection in English", *Proceedings of the 5th Workshop on Online Abuse and Harms*, pp. 17–25, Association for Computational Linguistics, Online, 2021.

8. Perifanos, K. and D. Goutsos, "Multimodal Hate Speech Detection in Greek Social Media", *Multimodal Technologies and Interaction*, Vol. 5, No. 7, p. 34, 2021.

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

10. Wiegand, M., J. Ruppenhofer, A. Schmidt and C. Greenberg, "Inducing a Lexicon of Abusive Words – a Feature-Based Approach", *North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 1046–1056, Association for Computational Linguistics, New Orleans, Louisiana, 2018.

11. Koufakou, A., E. W. Pamungkas, V. Basile and V. Patti, "HurtBERT: Incorporating Lexical Features with BERT for the Detection of Abusive Language", *Proceedings of the Fourth Workshop on Online Abuse and Harms*, pp. 34–43, Association for Computational Linguistics, Online, 2020.

12. Hüsünbeyi, Z. M., *Detecting Hate Speech in Turkish Texts*, Master's Thesis, Boğaziçi University, 2020.

13. LeCun, Y., B. Boser, J. Denker, D. Henderson, R. Howard, W. Hubbard and L. Jackel, "Handwritten Digit Recognition with a Back-Propagation Network", *Advances in Neural Information Processing Systems*, Vol. 2, 1989.

14. Hochreiter, S. and J. Schmidhuber, "Long Short-Term Memory", *Neural Computation*, Vol. 9, No. 8, pp. 1735–1780, 1997.

15. Cho, K., B. van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk

and Y. Bengio, "Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation", *Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1724–1734, Association for Computational Linguistics, Doha, Qatar, 2014.

16. Vaswani, A., N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser and I. Polosukhin, "Attention Is All You Need", *Advances in Neural Information Processing Systems*, Vol. 30, 2017.

17. Gambäck, B. and U. K. Sikdar, "Using Convolutional Neural Networks to Classify Hate-Speech", *Proceedings of the First Workshop on Abusive Language Online*, pp. 85–90, Association for Computational Linguistics, Vancouver, BC, Canada, 2017.

18. Mikolov, T., I. Sutskever, K. Chen, G. S. Corrado and J. Dean, "Distributed Representations of Words and Phrases and Their Compositionality", *Advances in Neural Information Processing Systems*, Vol. 26, 2013.

19. Pavlopoulos, J., P. Malakasiotis and I. Androutsopoulos, "Deeper Attention to Abusive User Content Moderation", *Empirical Methods in Natural Language Processing*, pp. 1125–1135, Association for Computational Linguistics, Copenhagen, Denmark, 2017.

20. Kim, Y., "Convolutional Neural Networks for Sentence Classification", *Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1746–1751, Association for Computational Linguistics, Doha, Qatar, 2014.

21. Zhang, Z., D. Robinson and J. Tepper, "Detecting Hate Speech on Twitter using a Convolution-Gru Based Deep Neural Network", *European Semantic Web Conference*, pp. 745–760, Springer, Crete, Greece, 2018.

22. Park, J. H. and P. Fung, "One-step and Two-step Classification for Abusive Language Detection on Twitter", *Proceedings of the First Workshop on Abusive Lan-*

guage Online, pp. 41–45, Association for Computational Linguistics, Vancouver, BC, Canada, 2017.

23. Pitsilis, G. K., H. Ramampiaro and H. Langseth, "Effective Hate-Speech Detection in Twitter Data using Recurrent Neural Networks", *Applied Intelligence*, Vol. 48, No. 12, pp. 4730–4742, 2018.

24. Founta, A. M., D. Chatzakou, N. Kourtellis, J. Blackburn, A. Vakali and I. Leontiadis, "A Unified Deep Learning Architecture for Abuse Detection", *Proceedings of the 10th ACM Conference on Web Science*, pp. 105–114, Boston, Massachusetts, USA, 2019.

25. Pérez, J. M. and F. M. Luque, "Atalaya at SemEval 2019 Task 5: Robust Embeddings for Tweet Classification", *Proceedings of the 13th International Workshop on Semantic Evaluation*, pp. 64–69, Association for Computational Linguistics, Minneapolis, Minnesota, USA, 2019.

26. Sarzynska-Wawer, J., A. Wawer, A. Pawlak, J. Szymanowska, I. Stefaniak, M. Jarkiewicz and L. Okruszek, "Detecting Formal Thought Disorder by Deep Contextualized Word Representations", *Psychiatry Research*, Vol. 304, p. 114135, 2021.

27. Indurthi, V., B. Syed, M. Shrivastava, N. Chakravartula, M. Gupta and V. Varma, "FERMI at SemEval-2019 Task 5: Using Sentence Embeddings to Identify Hate Speech Against Immigrants and Women in Twitter", *Proceedings of the 13th International Workshop on Semantic Evaluation*, pp. 70–74, Association for Computational Linguistics, Minneapolis, Minnesota, USA, 2019.

28. Antoun, W., F. Baly and H. Hajj, "AraBERT: Transformer-Based Model for Arabic Language Understanding", *Workshop Language Resources and Evaluation Conference*, p. 9, Marseille, France, 2020.

29. Schweter, S., "BERTurk - BERT Models for Turkish", 2020, `https://github.com/stefan-it/turkish-bert`, accessed on September 17, 2021.

30. Bahdanau, D., K. Cho and Y. Bengio, "Neural Machine Translation by Jointly Learning to Align and Translate", Y. Bengio and Y. LeCun (Editors), *3rd International Conference on Learning Representations*, San Diego, CA, USA, 2015.

31. Luong, T., H. Pham and C. D. Manning, "Effective Approaches to Attention-based Neural Machine Translation", *Empirical Methods in Natural Language Processing*, pp. 1412–1421, Association for Computational Linguistics, Lisbon, Portugal, 2015.

32. Yang, Z., D. Yang, C. Dyer, X. He, A. Smola and E. Hovy, "Hierarchical Attention Networks for Document Classification", *North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 1480–1489, Association for Computational Linguistics, San Diego, California, 2016.

33. Mehdad, Y. and J. Tetreault, "Do Characters Abuse More than Words?", *Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pp. 299–303, Los Angeles, USA, 2016.

34. Brown, P. F., V. J. Della Pietra, P. V. deSouza, J. C. Lai and R. L. Mercer, "Class-Based n-gram Models of Natural Language", *Computational Linguistics*, Vol. 18, No. 4, pp. 467–480, 1992.

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

36. Razavi, A. H., D. Inkpen, S. Uritsky and S. Matwin, "Offensive Language Detection using Multi-Level Classification", *Canadian Conference on Artificial Intelligence*, pp. 16–27, Springer, Ottawa, ON, Canada, 2010.

37. Xu, J.-M., K.-S. Jun, X. Zhu and A. Bellmore, "Learning from Bullying Traces

in Social Media", *North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 656–666, Association for Computational Linguistics, Montréal, Canada, 2012.

38. Chen, Y., Y. Zhou, S. Zhu and H. Xu, "Detecting Offensive Language in Social Media to Protect Adolescent Online Safety", *International Conference on Privacy, Security, Risk and Trust and International Conference on Social Computing*, pp. 71–80, USA, 2012.

39. Burnap, P. and M. L. Williams, "Cyber Hate Speech on Twitter: An Application of Machine Classification and Statistical Modeling for Policy and Decision Making", *Policy & Internet*, Vol. 7, No. 2, pp. 223–242, 2015.

40. Burnap, P. and M. L. Williams, "Us and Them: Identifying Cyber Hate on Twitter across Multiple Protected Characteristics", *EPJ Data Science*, Vol. 5, pp. 1–15, 2016.

41. ElSherief, M., V. Kulkarni, D. Nguyen, W. Y. Wang and E. Belding, "Hate Lingo: a Target-Based Linguistic Analysis of Hate Speech in Social Media", *Proceedings of the International AAAI Conference on Web and Social Media*, Vol. 12, California, USA, 2018.

42. ElSherief, M., C. Ziems, D. Muchlinski, V. Anupindi, J. Seybolt, M. De Choudhury and D. Yang, "Latent Hatred: a Benchmark for Understanding Implicit Hate Speech", *Empirical Methods in Natural Language Processing*, pp. 345–363, Association for Computational Linguistics, Online and Punta Cana, Dominican Republic, 2021.

43. Zampieri, M., S. Malmasi, P. Nakov, S. Rosenthal, N. Farra and R. Kumar, "Predicting the Type and Target of Offensive Posts in Social Media", *North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 1415–1420, Association for

Computational Linguistics, Minneapolis, Minnesota, 2019.

44. Basile, V., C. Bosco, E. Fersini, D. Nozza, V. Patti, F. M. Rangel Pardo, P. Rosso and M. Sanguinetti, "SemEval-2019 Task 5: Multilingual Detection of Hate Speech Against Immigrants and Women in Twitter", *Proceedings of the 13th International Workshop on Semantic Evaluation*, pp. 54–63, Association for Computational Linguistics, Minneapolis, Minnesota, USA, 2019.

45. Sap, M., S. Gabriel, L. Qin, D. Jurafsky, N. A. Smith and Y. Choi, "Social Bias Frames: Reasoning about Social and Power Implications of Language", *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 5477–5490, Association for Computational Linguistics, Online, 2020.

46. Mollas, I., Z. Chrysopoulou, S. Karlos and G. Tsoumakas, "ETHOS: a Multi-Label Hate Speech Detection Dataset", *Complex & Intelligent Systems*, 2022.

47. Mathew, B., P. Saha, S. M. Yimam, C. Biemann, P. Goyal and A. Mukherjee, "HateXplain: a Benchmark Dataset for Explainable Hate Speech Detection", *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35, pp. 14867–14875, Online, 2021.

48. Choi, G., S. Oh and H. Kim, "Improving Document-Level Sentiment Classification using Importance of Sentences", *Entropy*, Vol. 22, No. 12, p. 1336, 2020.

49. Loshchilov, I. and F. Hutter, "Decoupled Weight Decay Regularization", *International Conference on Learning Representations*, New Orleans, Louisiana, United States, 2019.

50. Srivastava, N., G. Hinton, A. Krizhevsky, I. Sutskever and R. Salakhutdinov, "Dropout: a Simple Way to Prevent Neural Networks from Overfitting", *The Journal of Machine Learning Research*, Vol. 15, No. 1, pp. 1929–1958, 2014.

51. HDF Publications, H. D. F., "Hate Speech and Discriminatory Discourse in Me-

dia", 2019, https://hrantdink.org/attachments/article/2727/Hate-Speech-and-Discriminatory-Discourse-in-Media-2019.pdf, accessed on September 13, 2021.