Natural Language Processing (2024), 1–26
doi:10.1017/nlp.2024.44

CAMBRIDGE UNIVERSITY PRESS

ARTICLE

Constructing ensembles for hate speech detection

Izzet Emre Kucukkaya[1] and Cagri Toraman[2]

[1]School of Computation, Information and Technology, Technical University of Munich, Munich, Germany and [2]Computer Engineering Department, Middle East Technical University, Ankara, Turkey
Corresponding author: Cagri Toraman; Email: ctoraman@ceng.metu.edu.tr

(Received 19 April 2023; revised 23 July 2024; accepted 23 July 2024)

Abstract

Hate speech against individuals and groups with certain demographics is a major issue in social media. Supervised models for hate speech detection mostly utilize labeled data collections to understand textual semantics. However, hate speech detection is a complex task that involves several aspects, including topic and writing style. The complexity of hate speech can be represented by an ensemble of models learned from different aspects of data. Moreover, ensemble members or base models can be modified to give attention to particular aspects of hate speech. In this study, we extract different aspects of hate speech to construct ensembles, thereby improving the performance of hate speech detection by ensemble learning. We conduct detailed experiments on five datasets in multiple languages to generalize our observations. The experimental results, supported by statistical significance tests, show that the performance of hate speech detection can be improved by capturing multiple aspects of hate speech. Our ensemble construction approach outperforms the baselines in terms of the F1 score of the Hate class in 80% of the cases, and the Offensive class in 75% of the cases. We also compare our approach with state-of-the-art ensemble methods from shared tasks and find that our highest-performing method can improve the performance of the Hate class in two out of three datasets. We further discuss our approach and experimental results in terms of ensemble parameters and writing style among ensemble members.

Keywords: Hate speech detection; ensemble learning; text classification; online social networks; offensive content

1. Introduction

There is an increasing number of hateful messages in social media that target specific communities or individuals with real-life consequences (Byman 2021). In order to maintain a positive and inclusive environment, it is essential to identify and address hate speech. However, this task becomes increasingly difficult as the number of users continues to grow, making manual detection nearly impossible.

Machine learning and deep learning are state-of-the-art solutions for automatic hate speech detection using natural language processing (Schmidt and Wiegand 2017). The performance of these models can be sometimes inadequate since they depend highly on data distribution, weight initialization, and backbone classifiers. The outputs of various methods can be combined to improve the success of predictions, namely ensemble learning (Gomes *et al.* 2017).

Hate speech can exist in different styles, including the intention and writing style of the author. The ambiguity of the difference between hate, offensive, and normal speech is a key factor in understanding hate speech correctly (Kumaresan and Vidanage 2019). Furthermore, the existence of a target group is one of the foundations of hate speech. According to Mondal, Silva, and Benevenuto (2017), there are different types of hate speech based on the target individuals

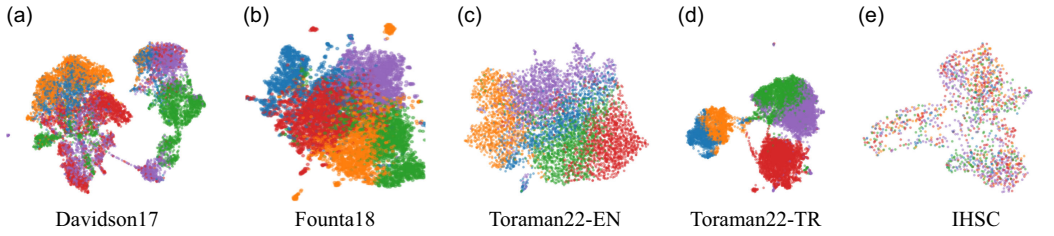

Figure 1. Ensemble members obtained by the Gaussian Mixture Model in five datasets. The hate speech datasets from (a) to (c) are in English (Davidson *et al.* 2017; Founta *et al.* 2018; Toraman, Sahinuç, and Yilmaz 2022a), (d) in Turkish (Toraman et al. 2022a), and (e) in Italian (Sanguinetti *et al.* 2018). The colors represent different parts of the hate speech datasets. UMAP (McInnes and Healy 2018) is used for dimension reduction. Best viewed in color. Gaussian Mixture Model is applied for clustering with several five clusters. The embedding vectors are derived using a Transformer-based deep learning model fine-tuned on hate speech detection datasets.

or groups. Topics and writing styles can be important aspects of defining and understanding the target of hate speech.

1.1 Motivation

Hate speech detection is a challenging task, as human annotators may not even agree with each other on whether a given text contains hate speech or not. Indeed, utilizing crowdsourcing with a larger number of people for the annotation of a hate speech dataset can lead to more reliable results (Founta *et al.* 2018). Expecting a single model to detect hate speech accurately can also be challenging.

In hate speech detection, we aim to make our models focus on different aspects of the dataset and contribute to the final result. Utilizing different aspects of hate speech exposed by different hateful groups with ensemble learning can improve the performance of hate speech detection, as well as modifying these aspects as needed (e.g. adding or removing an ensemble member as data changes). As an example of different aspects of hate speech detection, an ensemble can be constructed by focusing on different topics or target groups of hate speech in a given dataset.

We employ various methods for extracting different aspects of hate speech and constructing our ensembles. Conventional ensemble learning methods are bagging and fold ensembles (Dietterich 2000) that divide a given data into multiple splits without considering any particular data aspects. Using different topics or target groups in hate speech can be an alternative method that considers hate speech while splitting the data. For this purpose, we cluster train data by filtering topic keywords or applying the Gaussian Mixture Model (Reynolds 2009). We also construct our ensembles by dividing users according to their influential scores, since different writing styles provided by different degrees of influence can infer varying aspects of hate speech.

In Fig. 1, we illustrate this idea by clustering the training instances of five publicly available tweet datasets in English (Davidson *et al.* 2017; Founta *et al.* 2018; Toraman *et al.* 2022a) and other languages (Sanguinetti *et al.* 2018; Toraman *et al.* 2022a) in the literature. The clusters in the figure are obtained by the Gaussian Mixture Model, showing similar parts of the datasets. We use these clusters as ensemble members or base models to represent different aspects of hate speech datasets, thereby improving the performance of hate speech detection with ensemble learning.

1.2 Research questions

There are three main research questions in this study.

- **RQ-1** Does ensemble learning have benefits compared to a single classifier in terms of the performance of hate speech detection? To answer RQ-1, we compare our ensemble methods with three different single classifiers that do not employ ensemble learning at all.

https://doi.org/10.1017/nlp.2024.44 Published online by Cambridge University Press

- **RQ-2** Can we leverage different aspects of hate speech such as topics and authors to improve the performance of ensemble learning for hate speech detection? To answer RQ-2, we conduct a comprehensive experiment to compare six ensemble methods based on different aspects of hate speech with five baselines.
- **RQ-3** Is the performance of hate speech ensembles language-independent? To answer RQ-3, we conduct all experiments in three different languages (English, Italian, and Turkish).

1.3 Contributions

Our study mainly focuses on constructing an ensemble of classifiers that would reflect different aspects of hate speech in social media. The task is to improve the performance of hate speech detection by ensemble learning. The input data is social media texts, specifically tweets. The contributions of this study are as follows.

- We propose to extract different aspects of hate speech that are exposed by different hateful groups for ensemble learning. To do so, we focus on the diversity of the topics and authors of tweets.
- We publish the source code for the reproducibility of ensemble members and also lexicon resources used in the experiments at https://github.com/metunlp/hate-ensemble
- We conduct detailed experiments on multiple languages (i.e. English, Italian, and Turkish) to analyze the performance of ensemble learning for hate speech detection. The results show that the performance of hate speech detection can be improved by capturing different aspects of hate speech.

1.4 Bias statement

Following Kirk *et al.* (2022), this study discusses examples of harmful content (hate speech stereotypes). The authors do not support using harmful language or any of the harmful representations quoted in the study. Since we do not annotate tweets in this study, we rely on the definitions of hate speech that are followed in the existing datasets used in this study (Davidson *et al.* 2017; Founta *et al.* 2018; Sanguinetti *et al.* 2018; Toraman *et al.* 2022a).

1.5 Outline

The rest of the study is organized as follows. In the following section, we briefly summarize related work for ensemble learning and hate speech detection. We explain ensemble construction methods for hate speech detection in Section 3. We then present the experiments in Section 4, and a detailed discussion of the experimental results in Section 5. We conclude the study in the last section.

2. Related work

2.1 Ensemble learning

Ensemble learning aims to generate a set of classifiers (i.e. ensemble members or base models), and then merge their outputs to obtain the final prediction (Dietterich 2000). There are various ensemble strategies (Ganaie *et al.* 2022), including conventional classifier ensembles (Opitz and Maclin 1999) such as Bagging (Breiman 1996) and Boosting (Schapire 1990). Other methods include negative correlation learning (Liu and Yao 1999), ensemble pruning (Toraman and Can 2012), and explicit/implicit methods such as Dropconnect (Wan *et al.* 2013) and Dropout (Srivastava *et al.* 2014).

There are different techniques for ensemble construction. Stacking aims to construct multiple classifiers with different model types. On the other hand, other ensemble approaches employ the same model by learning different data subsets, for example Bagging (Breiman 1996) employs a

https://doi.org/10.1017/nlp.2024.44 Published online by Cambridge University Press

decision tree algorithm on different data subsets. In this study, we follow the latter approach, that is using the same model (i.e. homogeneous ensemble generation) but learning different data subsets, to exploit different characteristics of hate speech datasets.

The output of ensemble members is merged in fusion strategies, for example unweighted model averaging, majority voting, and super learner (Laan, Polley, and Hubbard 2007). In this study, we employ unweighted model averaging as our fusion strategy, and Transformer-based language models (Devlin *et al.* 2019; Sanh *et al.* 2019) as backbone classifier.

2.2 Text classification and hate speech detection

Automated hate speech detection is widely studied in the natural language processing community (Schmidt and Wiegand 2017). Text content is useful for extracting linguistic and syntactical features with the bag-of-words (Nobata *et al.* 2016; Waseem 2016; Burnap and Williams 2016; Davidson *et al.* 2017). User-based features, including meta-attributes and user profiles, can be good signals to detect hate speech (Waseem 2016; Chatzakou *et al.* 2017; Unsvåg and Gambäck 2018). Zhang *et al.* (2019) study feature engineering, specifically by adding lexico-semantic features and showing the improvements by various architectures such as CNNs, LSTMs, and SVM-based classifiers. Word embeddings (Pennington, Socher, and Manning 2014; Nobata *et al.* 2016; Mou, Ye, and Lee 2020) and deep neural networks (Hochreiter and Schmidhuber 1997; Kim 2014) are also employed to represent text semantics. Recently, BERT (Devlin *et al.* 2019) and similar encoder models based on Transformer (Vaswani *et al.* 2017) outperform previous methods for hate speech detection (Liu *et al.* 2019; Tekiroğlu, Chung, and Guerini 2020; Kennedy *et al.* 2020; Caselli *et al.* 2021; Mathew *et al.* 2021; Röttger *et al.* 2021; Toraman et al. 2022a).

Multi-task learning is leveraged by integrating hate speech detection with emotion detection to reflect the potential correlations between hate speech and certain negative emotion states (Min *et al.* 2023). Supervised contrastive learning is utilized for capturing span-level information for hate speech detection (Lu *et al.* 2023). The role of social stereotypes in hate speech detection is examined by the impact of social stereotypes on annotation and classification algorithms (Davani *et al.* 2023).

Furthermore, there are other efforts such as extracting hate spans from hateful tweets using Transformer-based language models (Khan, Ma, and Vosoughi 2021; Zhou *et al.* 2022), and dedicated shared tasks such as Toxic Spans Detection (Pavlopoulos *et al.* 2021; Ranasinghe and Zampieri 2021).

Instead of detecting hateful text segments, PAN21 is a shared task that profiles hate speech spreaders (Bevendorff *et al.* 2021). The dataset in this task consists of multiple tweets of the user. The participants have solutions to detect the overall tendency for spreading hate speech including Convolutional Neural Networks (CNN) (Siino *et al.* 2021). In this paper, we detect the existence of hate speech instead of hate speech spreaders. Yet, we utilize some author features such as the number of followers and followees in our ensemble methods.

2.3 Ensemble learning for hate speech detection

Ensemble learning is applied for hate speech detection. There are efforts to combine text features an ensemble of different classifiers with a voting scheme (Anusha and Shashirekha 2020; Hegde, Anusha, and Shashirekha 2021; Mutanga *et al.* 2022). Alsafari, Sadaoui, and Mouhoub (2020) create the ensemble of CNN and BiLSTM classifiers with contextual and non-contextual word-embedding models. Zimmerman, Kruschwitz, and Fox (2018) examine the ensemble of three CNN-based models whose weights are initialized randomly and trained on the same dataset. Tula *et al.* (2021) utilize an ensemble of different models such as distilmBERT (Sanh *et al.* 2019) and ULMFiT (Howard and Ruder 2018), and loss functions. Agarwal and Chowdary (2021) propose an ensemble learning-based adaptive model that works towards overcoming the strong user bias present in the dataset.

https://doi.org/10.1017/nlp.2024.44 Published online by Cambridge University Press

Recent advances in language models provided by the Transformer architecture (Vaswani *et al.* 2017) are also utilized with ensemble learning. Turban and Kruschwitz (2022) find that Transformer models are well-suited for ensemble learning by data augmentation techniques. Moreover, some participants in the Hate Speech Detection (HaSpeeDe) shared task at EVALITA (Sanguinetti *et al.* 2020) use ensemble techniques such as fine-tuning different Transformer-based language models on the same dataset, and then create an ensemble of these models to improve their results. Similarly, there are efforts to utilize an ensemble of different Transformer-based language models for Arabic hate speech detection (Mubarak, Al-Khalifa, and Al-Thubaity 2022; Magnossão de Paula *et al.* 2022).

There are several other shared tasks such as HASOC 2021 (Mandl *et al.* 2021) and OffensEval 2019 (Zampieri *et al.* 2019) and most of the participants propose methods based on ensemble learning. Contrary to our study, ensemble methods in such studies propose combining different classifiers and model architectures (Nikolov and Radivchev 2019; Kumar, Roy, and Saumya 2021).

Our main difference from the studies that utilize ensemble learning for hate speech detection is that we construct our ensembles by using different aspects of hate speech, such as topics and users. Our focus is not on the classification algorithm or voting scheme of ensemble learning but on the way of constructing ensemble members or base models by data splitting. That is, we investigate the effect of different aspects of the datasets by keeping the model architecture the same and modifying the approach of training and ensemble construction. One can employ different classification models for each split of the dataset, however, the improvement by ensemble construction might be affected by the choice of models. Moreover, we conduct detailed experiments to understand the performance of the proposed ensemble methods for hate speech detection in multiple languages.

3. Ensemble methods for hate speech detection

In this section, we first explain our main approach for constructing ensembles with different aspects of hate speech. We then explain six methods to generate ensemble members.

Our main approach is illustrated in Fig. 2. We divide a given data into five folds as in cross-validation, as shown in Fig. 2a. Each unique (non-overlapping) data split is 20% of the whole data and is used for testing or evaluating the performance of the ensemble methods. The remaining 80% of the data is used for both training and validation.

Different ensemble methods, which are explained in the following subsections, are applied on the 80% part, as represented in Fig. 2b. In the figure, we apply the Gaussian Mixture Model to split the data into five[a] coherent subsets, represented by different colors. We argue that these data splits can represent different aspects of hate speech. Therefore, we use each cluster as a train set and the remaining portion as a validation set, as represented by black and gray colors, respectively in Fig. 2c. Furthermore, due to the limited size of train splits, we introduce the Combination method that merges different data splits or aspects of hate speech. Fig. 2d depicts the members of a combination ensemble by merging multiple data splits into a single train data.

3.1 Random ensemble

As a baseline ensemble approach, we apply the bagging method (Breiman 1996), which randomly divides the original train set into 80% of the train and 20% of the validation set. The same process is repeated five times to obtain five ensemble members. We use the split ratio of 80-20 to have consistent training size with the main approach illustrated in Fig. 2. The best model is selected based on the validation loss for each member. We call this approach Random Ensemble

[a]We use five ensemble members in the experiments yet provide an analysis of varying numbers in the Discussion section.

https://doi.org/10.1017/nlp.2024.44 Published online by Cambridge University Press

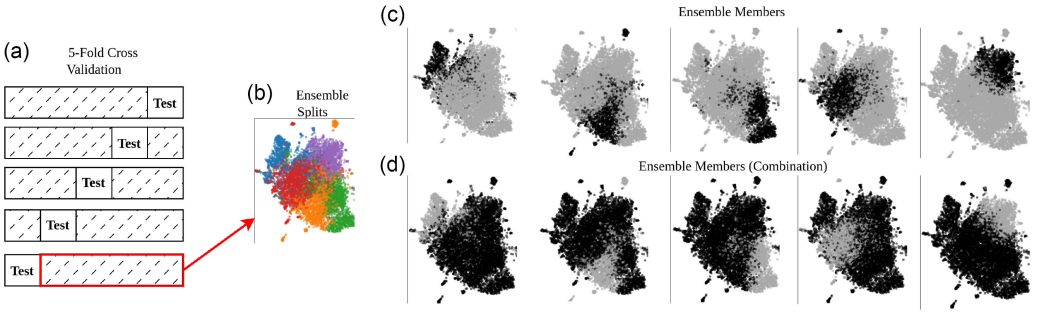

Figure 2. Illustration of our main approach for ensemble learning. (a) 5-fold cross-validation is applied to obtain non-overlapping test sets (20% of data). (b) The remaining 80% of the data is used for train and validation. As an example, the Gaussian Mixture Model is applied to obtain five different parts or aspects of data, represented by different colors. (c) Five ensemble members or base models are trained by using a single part of the data, represented by black color. The remaining four parts are used for validation. (d) In the Combination method, ensemble members are trained by four parts merged, represented by black color. The remaining part is used for validation.

since the instances in data splits are obtained randomly. The random ensemble is similar to the Combination methods in terms of train size.

3.2 Fold ensemble

Another baseline approach is the fold ensemble. In this method, the training set is stratified and divided into five folds. That is, each fold contains 20% of the train data, and there is no overlap among the fold splits. To construct an ensemble member, four of five data splits are merged (80% of the train data). This process is repeated five times with different four splits (i.e. three of five data splits). The remaining 20% of data is used for validation. In contrast to Random Ensemble, we ensure that the ratio of the classes is preserved, and each ensemble member has a different train set with 75% of overlap with other ensemble members. The fold ensemble is similar to the Combination method in terms of train size.

3.3 Topic ensemble

Random and fold ensembles do not necessarily represent different aspects of hate speech. In topic ensemble, the train set is divided according to specific hate topics such as race and gender. We rely on existing topic labels in the datasets. If topic labels do not exist, we filter the dataset by topic keywords[b] to obtain topic sets. That is, tweets are assigned to a particular topic label by the existence of a topic keyword. An instance can be assigned to multiple topics.

An example of topic splitting is illustrated in Fig. 3a. In this figure, we divide a dataset used in our experiments to show an example of how topic splits are represented in UMAP embedding space (McInnes and Healy 2018) using DistilBERT embeddings (Sanh *et al.* 2019). Although the instances belonging to particular topics are mainly clustered in the embeddings space, we observe no strict distinction among different topics, represented by different colors in the figure. This observation might show that there can be overlapping parts across topics (e.g. gender-based discussions in religions), and the Topic ensemble method can extract different aspects of hate speech exposed in different topics. Since topics are content features, we refer to the Topic ensemble method as a content-based ensemble method. Validation and test splits are generated as in the main approach illustrated in Fig. 2.

[b]This keyword list is generated by merging the topic keywords in existing datasets (Davidson *et al.* 2017; Basile *et al.* 2019; Toraman, Sahinuç, and Yilmaz 2022a). We publish the merged keyword list at https://github.com/metunlp/hate-ensemble

https://doi.org/10.1017/nlp.2024.44 Published online by Cambridge University Press

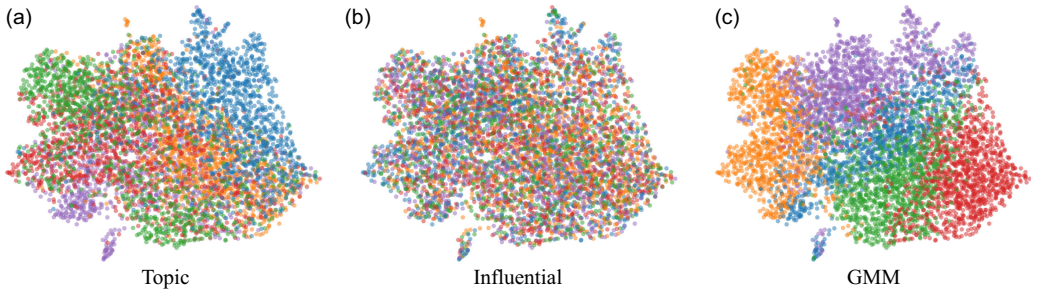

Figure 3. Data splits for different ensemble methods (Topic, Influential, and GMM) in the Toraman22-EN dataset by using UMAP representation (McInnes and Healy 2018). Best viewed in color.

3.4 Influential ensemble

The user's writing style can be an important factor in hate speech detection. We assume that users interacting with more people tend to have different writing styles (Martin, Sintsova, and Pu 2014). Such influential users can have different writing styles due to their popularity. Based on the tweet author's influential score, we divide the training set into five splits with an equal number of samples in each. We use the influential score, similar to (Toraman *et al.* 2022b), as $\log(n_{follower}/n_{followee})$ where $n_{follower}$ and $n_{followee}$ are the numbers of followers and followees, respectively. The tweets that are shared by the users are then used in training.

An example of influential splitting is illustrated in Figure 3b. We observe significant overlapping parts across different influential splits, represented by different colors in the figure. This observation might suggest that the difference in writing styles of different influential groups is not based on the context of tweets, and the Influential Ensemble method can extract different writing styles of tweet authors who have different user features, particularly being influential in this case. We refer to the Influential ensemble method as a user-based ensemble method. Validation and test splits are generated as in the main approach illustrated in Fig. 2.

3.5 GMM ensemble

Topic ensemble relies on topic labels or keywords. To extract different topics of hate speech in an unsupervised way without labels and keywords, we use Gaussian Mixture Modeling (GMM) (Reynolds 2009) that clusters the training set as a mixture of five Gaussian distributions. We choose Gaussian Mixture Modeling as the clustering method in this study for the following reasons. GMM can describe a complex set with a mixture of Gaussian distributions, which can fit the task of text clustering with pre-trained embeddings (Aharoni and Goldberg 2020). Moreover, we use pre-trained embeddings as feature vectors, and GMM outperforms other clustering methods, such as the k-means clustering (Lloyd 1982), when pre-trained embeddings are employed for representing topics in text (Sia, Dalmia, and Mielke 2020).

We use GMM to assign a sample to the most probable distribution to apply hard assignments. The GMM is initialized randomly using a fixed seed. An example of GMM splitting is illustrated in Fig. 3c. We observe a clear distinction of clusters or data splits, represented by different colors in the figure. The GMM ensemble can be useful for understanding the performance of hate ensembles when the contents in data splits are closely similar to each other. Since GMM is a probabilistic model that assumes that the instances are obtained from a mix of Gaussian distributions based on the input contents, we refer to the GMM ensemble method as a content-based ensemble method.

https://doi.org/10.1017/nlp.2024.44 Published online by Cambridge University Press

3.6 Combination ensemble

In contrast to using data folds as individual ensemble members, we use each fold as a validation set and combine the remaining data as a training set in the Combination method (i.e. 80% of data is for training and 20% of data is for validation), as represented in Fig. 2d. When the Combination method is not applied, training is based on a unique 20% of data, since there are five folds and we select a different fold each time for training. However, the Combination method has overlapping instances among five ensemble members since it merges four folds each time. The motivation is to increase the number of instances to be used in the training of each ensemble member while combining different aspects of hate speech.

We apply the Combination method to the Topic, Influential, and GMM methods. Note that the baseline methods, Random and Fold, have already 80% of data in the train. In other words, the normal versions of the Topic, Influential, and GMM methods have smaller training sizes than the baselines, which may cause a disadvantage in training. However, they can still challenge baselines as observed in the Experiments section.

4. Experiments

In this section, we first explain the datasets used in the experiments. We then report the experimental design and results.

4.1 Datasets

There are five datasets used in the experiments. Davidson17 (Davidson *et al.* 2017) and Founta18 (Founta *et al.* 2018) are two widely used datasets for the task of hate speech detection. Toraman22-EN (Toraman *et al.* 2022a) is a recent large-scale dataset with topic information, which is an important factor in creating ensemble members in this study. In order to generalize the results to multiple languages, we also use the datasets in Italian (IHSC) (Sanguinetti *et al.* 2018) and Turkish (Toraman22-TR) (Toraman *et al.* 2022a). The reason for selecting these languages is that our ensemble approach requires topic and user information, which are partly provided in these datasets. There are also other datasets for hate speech detection such as SOLID (Rosenthal *et al.* 2021) with 9 million samples annotated in a self-supervised fashion. However, in this study, we rely on manually annotated datasets.

We query Twitter API to get additional features for the tweet authors in Founta18, Toraman22, and IHSC such as the number of followers and accounts being followed, to utilize them in Influential Ensemble. Moreover, since only the Toraman22 datasets provide topic labels, we filter the other datasets to obtain the instances related to topics. The final distribution of class labels and topics used in the experiments are given for all datasets in Tables 1 and 2, respectively. The details of how we obtain the dataset versions used in the experiments are given in the following subsections for each dataset.

4.1.1 Davidson17

The Davidson17 dataset has 24,784 English tweets (Davidson *et al.* 2017). The dataset is annotated with three labels: Neutral, Offensive, and Hateful. For Topic Ensemble, topic labels are not provided, so we manually filter tweets by topic keywords. Out of 24,784 tweets, 15,632 were obtained for six topics. 3,352 of these samples pertain to more than one topic. While training, we use such samples in multiple topics. For Influential Ensemble, tweet IDs and user IDs are unavailable, preventing us from extracting user features and influential scores. The Influential ensemble is, therefore, not reported for the Davidson17 dataset in the experiments.

https://doi.org/10.1017/nlp.2024.44 Published online by Cambridge University Press

Table 1. Class distributions of the datasets used in the experiments

Dataset	Language	Normal	Offensive/Abusive	Spam	Hate	Total
Davidson17	English	1,240	13,223	0*	1,169	15,632
Founta18	English	5,134	2,592	875	1,073	9,674
Toraman22-EN	English	5,000	5,000	0*	5,000	15,000
IHSC	Italian	1,481	0**	0*	333	1,814
Toraman22-TR	Turkish	5,000	5,000	0*	5,000	15,000

*The dataset includes no sample in the Spam class.
**The IHSC dataset includes no sample in the Offensive or Abusive class.

Table 2. Topic distributions of the datasets used in the experiments

Dataset	Language	Politics	Sports	Race	Gender	Disability	Religion
Davidson17*	English	274	369	4,296	13,375	657	167
Founta18*	English	3,467	1,982	1,868	1,653	1,470	632
Toraman22-EN	English	3,000	3,000	3,000	3,000	0**	3,000
IHSC*	Italian	407	51	860	75	22	606
Toraman22-TR	Turkish	3,000	3,000	3,000	3,000	0**	3,000

*Some instances have multiple topic labels.
**The dataset includes no sample in the Disability topic.

4.1.2 Founta18

The Founta18 dataset has 99,799 English tweets (Founta *et al.* 2018), given as tweet IDs. We query them in the Twitter API to retrieve data, and 51,216 instances are obtained due to deleted users or tweets. The dataset is annotated with four labels: Normal, Abusive, Spam, and Hateful. For Topic Ensemble, topic labels are not provided, so we manually filter tweets by topic keywords. Out of 51,216 tweets, 9,674 were obtained for six topics. 1,295 of these samples pertain to more than one topic. While training, we use such samples in multiple topics.

4.1.3 Toraman22-EN

The Toraman22-EN dataset has 100,000 English tweets (Toraman *et al.* 2022a). The dataset is annotated with three labels: Normal, Offensive, and Hateful. For the Topic Ensemble, five topic labels are already provided in the dataset. We sample 3,000 tweets from each topic while preserving the balance of classes, resulting in 15,000 tweets to get a data size similar to the other datasets. For Influential Ensemble, we query user features by tweet IDs in the Twitter API.

4.1.4 IHSC

The Italian Hate Speech Corpus (IHSC) has 6,928 Italian tweets (Sanguinetti *et al.* 2018), published as a set of tweet IDs. The dataset is annotated with two labels: Normal and Hateful. We query them in the Twitter API to retrieve data and fetch 5,090 instances due to deleted users and tweets. For Topic Ensemble, topic labels are not provided, so we manually filter tweets by topic keywords. Out of 5,090 tweets, 1,814 were obtained for six topics. 202 of these samples pertain to more than one topic. While training, we use such samples in multiple topics.

https://doi.org/10.1017/nlp.2024.44 Published online by Cambridge University Press

4.1.5 Toraman22-TR

The Toraman22-TR dataset has 100,000 Turkish tweets (Toraman *et al.* 2022a). The dataset is annotated with three labels: Normal, Offensive, and Hateful. For the Topic Ensemble, five topic labels are already provided in the dataset. We sample 3,000 tweets from each topic while preserving the balance of classes, resulting in 15,000 tweets in order to get the data size similar to the other datasets. For Influential Ensemble, we query user features by tweet IDs in the Twitter API.

4.2 Experimental design

4.2.1 Experimental setup

We set the number of ensemble members to five since the number of labeled topics in the Toraman22 datasets is five and other datasets do not publish predetermined topic labels. We still provide an additional discussion on the varying numbers of ensemble members in Section 5.2. For preprocessing, we lowercase text and apply padding to the maximum input length of tokens with truncation.

As the backbone classifier for English, we fine-tune DistilBERT (Sanh *et al.* 2019), a distilled version of BERT (Devlin *et al.* 2019), for the sake of efficiency since the number of training runs is significantly increased by training multiple ensemble members. We need an efficient model in terms of training time considering our limited hardware budgets. DistilBERT is a good option for this purpose since it is 60% faster than base models while retaining 97% of its language understanding capabilities (Sanh *et al.* 2019).

For the baselines where we employ no ensembles (i.e. single classifiers based on varying features), we employ Autogluon (Shi *et al.* 2021) which is a well-established framework for training NLP models efficiently and robustly. For other languages, we use the Turkish version of BERT, called BERTurk,[c] and the Italian version of BERT, called Italian BERT[d]. After the CLS embedding at the last layer, we place a linear classification layer. We use the weighted cross-entropy loss, considering the number of instances per class. We choose to employ Transformer-based language models (i.e. BERT-like models)) since they outperform conventional bag-of-words and also other deep neural methods for hate speech detection (Caselli *et al.* 2021; Röttger *et al.* 2021; Toraman *et al.* 2022a).

4.2.2 Hyperparameters

We use the scikit-learn (Pedregosa *et al.* 2011) implementation of the Gaussian Mixture Model. The parameters of the GMM are set to default as follows. The covariance type is full, the number of components is five, the threshold is 0.01, the maximum number of iterations is 150, and the floor on the diagonal of the covariance matrix to prevent overfitting is 0.001.

We use distilbert-base-uncased for DistilBERT, bert-base-turkish-cased for BERTurk, and bert-base-italian-cased for Italian BERT with the help of the Transformers library by Huggingface (Wolf *et al.* 2020). Epoch size is set to 5, the learning rate is set to 5e-5, the learning rate scheduler is set to default (linear), the optimizer is set to default (adamw_hf), the batch size is set to 16, and the dropout rate is set to 0.4. We use the same hyperparameters for all methods. The values are determined in preliminary experiments by considering the task and hardware that we employ. The main idea is to keep all these parameters fixed for all methods since they are used for fine-tuning Transformer-based language models rather than ensemble learning. In these settings, training and evaluation take in between 50 and 60 minutes using three GPUs (Nvidia RTX 2080) for five ensemble members.

[c] BERTurk: https://huggingface.co/dbmdz/bert-base-turkish-cased
[d] Italian BERT: https://huggingface.co/dbmdz/bert-base-italian-cased

https://doi.org/10.1017/nlp.2024.44 Published online by Cambridge University Press

4.2.3 Evaluation metric

We use stratified 5-fold validation. For each fold, 20% of the data is set aside as the test set, while the remaining 80% is further split into a train set and validation set. Since the number of ensemble members is set to five, we split this 80% of data into five parts of equal size, and then obtain train and validation sets based on the ensemble methods explained in Section 3. The validation sets are used for finding the optimal models according to their validation losses during a number of epochs of training. Within each fold, we train five base models with the same backbone using different train-validation pairs derived from the 80% part of the whole data as explained in Section 3. The softmax probabilities of five models are unweighted-averaged to get the final prediction.

We evaluate models with the F1 score using scikit-learn (Pedregosa *et al.* 2011). The F1 score is calculated as follows.

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

where *Precision* is the ratio of the correct predictions over all predictions, and *Recall* is the ratio of the correct predictions over all true samples. We apply the weighted F1 score to get the average score over all classes since most of the datasets in the experiments are imbalanced.

We report the average of 5-fold results, along with their standard deviation. Moreover, we apply a paired t-test to find any statistically significant difference at a 95% interval in pairwise comparisons between two mean scores.

4.2.4 Baselines

Our focus in this study is the way of constructing ensemble members or base models by data splitting. We, therefore, compare our ensemble construction methods with the following baseline methods:

- **Single-Text:** We fine-tune a single model with the same size of train data. The motivation is to understand the performance when no ensemble method is applied.
- **Single-Text-Topic:** We fine-tune a single classifier that includes the text features and topic information. CLS text embeddings from the Transformer-based language models are concatenated with topic embedding (one-hot vector) that represents the tweet's topic. This concatenated embedding vector is then given to a linear classification layer.
- **Single-Text-Influential:** We fine-tune a single classifier that includes the text features and user features. CLS text embeddings are concatenated with influential features, which are follower and followee counts. The features are then given to a linear classification layer.
- **Ensemble-Bagging:** A baseline ensemble construction method is to create ensemble members by selecting data randomly, as explained in Section 3.
- **Ensemble-Fold:** A baseline ensemble construction method is to create ensemble members by splitting data discretely, as explained in Section 3.

We refer to our ensemble construction methods (i.e. Topic, GMM, Influential, and their Combination versions) as Ensemble-MethodName in the experimental results. For instance, Ensemble-Topic refers to the ensemble approach explained in Section 3.3.

4.3 Experimental results

We present the results of ensemble methods for multi-class hate speech detection in all datasets in Table 3. We omit the results of a class label when the dataset includes no instances for this class label. Our observations that reflect the answers to the research questions of this study are given as follows.

https://doi.org/10.1017/nlp.2024.44 Published online by Cambridge University Press

Table 3. Comparison of ensemble methods for hate speech detection. An average of 5-fold is reported in terms of the F1 score. The highest scores for each class and dataset are given in bold. The symbol "*" indicates a statistically significant difference using a paired t-test at a 95% interval in pairwise comparisons between the highest-performing method and others

	Method	Normal	Offensive/Abusive	Spam	Hate	Weighted F1
Davidson17	Single-Text	0.845 ± 0.016	0.913 ± 0.017*	-	0.444 ± 0.044*	0.872 ± 0.018*
	Single-Text-Topic	0.818 ± 0.007*	**0.946 ± 0.005**	-	0.424 ± 0.036*	0.896 ± 0.005
	Ensemble-Bagging	0.860 ± 0.010	0.914 ± 0.011*	-	0.450 ± 0.029	0.875 ± 0.011*
	Ensemble-Fold	0.858 ± 0.006	0.934 ± 0.007*	-	0.496 ± 0.019	0.895 ± 0.007
	Ensemble-Topic	0.834 ± 0.017	0.945 ± 0.004	-	**0.497 ± 0.023**	**0.903 ± 0.005**
	Ensemble-GMM	0.828 ± 0.006*	0.919 ± 0.017*	-	0.450 ± 0.026	0.876 ± 0.016*
	Ensemble-Topic-Comb	0.861 ± 0.007*	0.929 ± 0.016	-	0.491 ± 0.033	0.891 ± 0.016
	Ensemble-GMM-Comb	**0.863 ± 0.009**	0.928 ± 0.007*	-	0.485 ± 0.022	0.890 ± 0.007
Founta18	Single-Text	0.744 ± 0.055*	0.800 ± 0.010*	0.505 ± 0.031	0.460 ± 0.017	0.706 ± 0.031*
	Single-Text-Topic	**0.848 ± 0.007**	0.806 ± 0.008*	0.444 ± 0.044*	0.388 ± 0.019*	**0.748 ± 0.007**
	Single-Text-Influential	0.842 ± 0.007	0.806 ± 0.011	0.448 ± 0.019*	0.378 ± 0.032*	0.746 ± 0.005
	Ensemble-Bagging	0.789 ± 0.013*	0.803 ± 0.013*	0.532 ± 0.006	0.475 ± 0.029	0.734 ± 0.007*
	Ensemble-Fold	0.798 ± 0.018*	0.809 ± 0.004*	0.532 ± 0.020	0.474 ± 0.033	0.741 ± 0.012
	Ensemble-Topic	0.812 ± 0.017*	0.810 ± 0.005	0.406 ± 0.020	0.406 ± 0.050*	0.739 ± 0.009
	Ensemble-Influential	0.783 ± 0.018*	0.805 ± 0.006*	0.516 ± 0.007*	0.444 ± 0.033*	0.727 ± 0.008*
	Ensemble-GMM	0.824 ± 0.034	**0.825 ± 0.006**	0.470 ± 0.056	0.399 ± 0.051*	0.745 ± 0.022
	Ensemble-Topic-Comb	0.800 ± 0.011*	0.808 ± 0.008*	**0.538 ± 0.013**	0.469 ± 0.023	0.742 ± 0.006
	Ensemble-Influential-Comb	0.799 ± 0.010*	0.811 ± 0.010*	0.535 ± 0.008	0.474 ± 0.033	0.742 ± 0.005
	Ensemble-GMM-Comb	0.781 ± 0.017*	0.810 ± 0.009*	0.527 ± 0.020	**0.477 ± 0.030**	0.732 ± 0.015
Toraman22-EN	Single-Text	0.696 ± 0.012	0.604 ± 0.020	-	0.626 ± 0.030	0.642 ± 0.013
	Single-Text-Topic	0.686 ± 0.005*	0.582 ± 0.012*	-	0.610 ± 0.019*	0.626 ± 0.012*
	Single-Text-Influential	0.684 ± 0.014*	0.576 ± 0.019*	-	0.600 ± 0.006*	0.622 ± 0.012*
	Ensemble-Bagging	0.714 ± 0.004	0.605 ± 0.013*	-	0.633 ± 0.012*	0.651 ± 0.007*
	Ensemble-Fold	0.714 ± 0.005	0.604 ± 0.009*	-	**0.647 ± 0.124**	**0.655 ± 0.007**
	Ensemble-Topic	0.680 ± 0.008*	0.593 ± 0.018*	-	0.579 ± 0.022*	0.617 ± 0.004*
	Ensemble-Influential	0.692 ± 0.009*	0.593 ± 0.010*	-	0.611 ± 0.016*	0.632 ± 0.007*
	Ensemble-GMM	0.685 ± 0.007*	0.589 ± 0.017*	-	0.610 ± 0.010*	0.628 ± 0.006*
	Ensemble-Topic-Comb	0.711 ± 0.003	**0.616 ± 0.008**	-	0.632 ± 0.015*	0.653 ± 0.007
	Ensemble-Influential-Comb	0.711 ± 0.006	**0.610 ± 0.010**	-	0.637 ± 0.015	0.653 ± 0.010
	Ensemble-GMM-Comb	**0.716 ± 0.010**	0.611 ± 0.014	-	0.633 ± 0.015*	0.654 ± 0.010
IHSC	Single-Text	0.861 ± 0.030*	-	-	0.548 ± 0.026*	0.804 ± 0.029*
	Single-Text-Topic	**0.910 ± 0.011**	-	-	0.562 ± 0.056	0.846 ± 0.014
	Single-Text-Influential	0.906 ± 0.008	-	-	0.502 ± 0.064*	0.832 ± 0.013*
	Ensemble-Bagging	0.898 ± 0.008	-	-	0.619 ± 0.018	0.847 ± 0.006
	Ensemble-Fold	0.908 ± 0.006	-	-	0.614 ± 0.030	**0.854 ± 0.009**
	Ensemble-Topic	0.879 ± 0.008*	-	-	0.438 ± 0.110*	0.798 ± 0.017*

https://doi.org/10.1017/nlp.2024.44 Published online by Cambridge University Press

Table 3. Continued

	Method	Normal	Offensive/Abusive	Spam	Hate	Weighted F1
	Ensemble-Influential	0.844 ± 0.021*	-	-	0.493 ± 0.014*	0.780 ± 0.019*
	Ensemble-GMM	0.828 ± 0.004*	-	-	0.490 ± 0.014	0.766 ± 0.004*
	Ensemble-Topic-Comb	0.907 ± 0.009	-	-	0.607 ± 0.041	0.852 ± 0.014
	Ensemble-Influential-Comb	0.887 ± 0.016	-	-	0.593 ± 0.022	0.833 ± 0.015*
	Ensemble-GMM-Comb	0.899 ± 0.009	-	-	**0.628 ± 0.026**	0.849 ± 0.011
Toraman22-TR	Single-Text	0.779 ± 0.004*	0.666 ± 0.017*	-	0.712 ± 0.008*	0.719 ± 0.006*
	Single-Text-Topic	0.776 ± 0.005*	0.656 ± 0.013*	-	0.716 ± 0.008*	0.716 ± 0.008*
	Single-Text-Influential	0.778 ± 0.004*	0.665 ± 0.015*	-	0.600 ± 0.006*	0.622 ± 0.012*
	Ensemble-Bagging	0.799 ± 0.004	0.690 ± 0.010	-	0.739 ± 0.012*	0.743 ± 0.004*
	Ensemble-Fold	0.792 ± 0.008*	0.688 ± 0.012	-	0.745 ± 0.011	0.741 ± 0.008
	Ensemble-Topic	0.759 ± 0.006*	0.642 ± 0.017*	-	0.711 ± 0.008*	0.704 ± 0.005*
	Ensemble-Influential	0.771 ± 0.007*	0.658 ± 0.026*	-	0.711 ± 0.015*	0.714 ± 0.015*
	Ensemble-GMM	0.768 ± 0.008*	0.659 ± 0.015*	-	0.706 ± 0.020*	0.711 ± 0.011*
	Ensemble-Topic-Comb	**0.799 ± 0.006**	**0.695 ± 0.009**	-	0.735 ± 0.010*	0.743 ± 0.006*
	Ensemble-Influential-Comb	0.794 ± 0.005*	0.685 ± 0.012*	-	0.732 ± 0.013*	0.737 ± 0.007*
	Ensemble-GMM-Comb	0.799 ± 0.008	0.690 ± 0.009*	-	**0.746 ± 0.012**	**0.745 ± 0.007**

RQ-1: Improvements by ensemble learning. We examine if ensemble learning provides any improvements to the performance of hate speech detection. Therefore, we pair the performance scores of a single model with an ensemble model. For instance, there are two single and six ensemble models in Davidson17, resulting in 12 pairs concerning Weighted F1 score. We observe that the highest-performing method in terms of the weighted F1 score is a type of ensemble learning, compared to the Single classifier, in all datasets. The ensemble methods yield better scores than the single methods in 67 of 108 (62%) comparisons in terms of the weighted F1 score. Furthermore, the Combination ensemble methods have better results than the single methods in 31 of 40 (78%) comparisons. The Combination method merges different splits to increase the number of instances and thereby improves the performance in most cases.

RQ-2: Benefits of ensemble methods that leverage different aspects of hate speech. When we compare our ensemble methods with the baseline ensemble approaches, we find that our proposed ensemble construction methods perform the highest in terms of the F1 score of the Hate class in four of five datasets (80% of the datasets) and in terms of the F1 score of the Offensive class in three of four datasets (75% of the datasets). We thereby argue that our ensemble construction methods can capture better different aspects of hate speech compared to the baseline ensemble construction methods. On the other hand, the Fold method performs the highest for the Hate class and weighted F1 score in Toraman22-EN. The reason could be the balanced distribution of both classes and topics in this dataset. Similarly, the Fold method performs highest in terms of the weighted F1 score in the IHSC dataset, however, our method has a better performance in the positive class (Hate), which is more important in this case since the task is binary classification (Normal vs. Hate) in this dataset.

Ensemble methods based on different user styles can be useful for reflecting user styles for hate speech. We demonstrate this idea with the influential feature of tweet authors. However, the Influential method is not the highest-performing method in any datasets, probably because the users in the same ensemble member do not necessarily have the same writing style of hate speech,

https://doi.org/10.1017/nlp.2024.44 Published online by Cambridge University Press

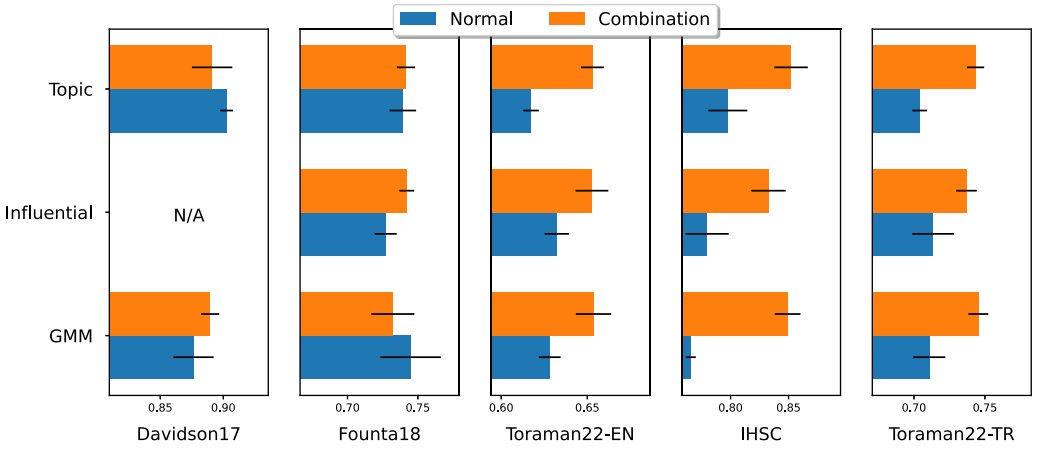

Figure 4. Performance comparison of our proposed ensemble methods (i.e. GMM, Topic, and Influential) with their Combination versions, as explained in Section 3, for all datasets in different subplots.

or due to possible noise in the ensemble members. Nevertheless, we argue that this method is still promising such that its combination version performs in the top-3 highest methods in Founta18 and Toraman22-EN. We argue that a deeper examination of data splits or user features might improve performance by reflecting different hate styles.

When we compare our normal and combination approach for constructing ensembles in Fig. 4, we observe that the Combination version outperforms in 12 of 14 (86%) comparisons. The better performance of the Combination method is probably due to the increased amount of samples in data splits used in training each ensemble member. The efficacy of the suggested models depends critically on the size of the training set while using deep learning methods (Bailly *et al.* 2022). Note that our normal method uses a single data split (20% of the whole data) to train an ensemble member, while the Combination method merges four splits (80% of the whole data) that result in a similar number of instances as in the baseline methods (Bagging and Fold). We argue that our main ensemble approach splits the data to obtain different aspects of hate speech for ensemble learning, and the combination of different aspects can further improve the performance of hate speech detection.

In terms of class labels, we observe that the content-based models (i.e. GMM and Topic) perform the highest for the Hate class in four of five datasets. We argue that different writing styles can be captured by the content-based models since they cluster sentence embeddings, which are encoding vectors of the text contents.

Furthermore, we observe that the Topic method performs the highest for the Offensive/Abusive class in three of four datasets and for the Spam class in the Founta18 dataset. Some words are not necessarily offensive, abusive, or spam in another context. We thereby argue that the topic splits can better capture the differences in offensive, abusive, and spam classes.

RQ-3: Language-independent ensemble methods for hate speech detection. The good performance of ensemble methods is also valid for other languages, that is Italian and Turkish. In our experiments, the highest-performing ensemble method improves the performance of the single approach statistically significantly in both Italian and Turkish languages. Moreover, an ensemble method that leverages different topic aspects of hate speech, GMM Combination, has the highest performance in the Hate class in both Italian and Turkish. We argue that our observations support that the performance of hate speech ensembles is language-independent, though the experiments are still limited to three languages (English, Italian, and Turkish).

https://doi.org/10.1017/nlp.2024.44 Published online by Cambridge University Press

Table 4. Comparison of our highest scores with state-of-the-art ensemble methods. An average of 5-fold is reported in terms of the F1 score. The highest scores for each class and dataset are given in bold. Davidson17, Founta18, and Toraman22-EN are shortened as D17, F18, T22, respectively

	Method	Normal	Offensive/ Abusive	Spam	Hate	Weighted F1
D17	Glazkova et al. (2021)	**0.857 ± 0.006**	0.952 ± 0.003	-	0.467 ± 0.024	**0.908 ± 0.004**
	Wiedemann et al. (2020)	0.849 ± 0.007	**0.952 ± 0.002**	-	0.335 ± 0.081	0.898 ± 0.007
	Ensemble-Topic (Ours)	0.834 ± 0.017	0.945 ± 0.004	-	**0.497 ± 0.023**	0.903 ± 0.005
F18	Glazkova et al. (2021)	**0.859 ± 0.004**	0.816 ± 0.007	0.481 ± 0.040	0.469 ± 0.025	**0.770 ± 0.005**
	Wiedemann et al. (2020)	0.865 ± 0.007	**0.834 ± 0.007**	0.409 ± 0.058	0.402 ± 0.041	0.764 ± 0.003
	Ensemble-GMM-Comb (Ours)	0.781 ± 0.017	0.810 ± 0.009	**0.527 ± 0.020**	**0.477 ± 0.030**	0.732 ± 0.015
T22	Glazkova et al. (2021)	0.724 ± 0.006	**0.613 ± 0.009**	-	0.650 ± 0.013	0.662 ± 0.007
	Wiedemann et al. (2020)	**0.728 ± 0.011**	0.612 ± 0.016	-	**0.664 ± 0.012**	**0.668 ± 0.009**
	Ensemble-Influential-Comb (Ours)	0.711 ± 0.006	0.610 ± 0.010	-	0.637 ± 0.015	0.653 ± 0.010

5. Discussion

In this section, we provide a post-analysis of the ensemble approach and experimental results in terms of additional baselines, ensemble parameters, and writing style. We also include a brief discussion on the limitations of our study.

5.1 Comparison with state-of-the-art ensemble methods

Our approach emphasizes the utilization of different styles of hate speech for ensemble learning. We therefore compare our ensemble construction methods with baseline ensemble construction methods in Section 4. However, ensemble learning is a widely used method in hate speech detection especially in shared tasks and challenges, as mentioned in Section 2.1. We implement two of the state-of-the-art ensemble learning models which yielded the best results on their shared tasks (Wiedemann, Yimam, and Biemann 2020; Glazkova, Kadantsev, and Glazkov 2021). The results are reported in Table 4.

Glazkova *et al.* (2021) proposed an ensemble model for HASOC in FIRE 2021 (Mandl *et al.* 2021). The authors implemented a five-member ensemble with a soft-voting algorithm (i.e. averaging the prediction probabilities). They used Twitter-RoBERTa (Barbieri *et al.* 2020) as the backbone model. On the other hand, Wiedemann *et al.* (2020) proposed an ensemble method for the shared task SemEval-2020 (Zampieri *et al.* 2020). They implemented a 10-fold ensemble method, as similar to our combination fold method, using RoBERTa-large (Liu *et al.* 2019). We re-implement their method with a smaller model from the same model family, RoBERTa-base, since our limited hardware resources do not support RoBERTa-large while training ensemble learning.

Although the state-of-the-art methods have better weighted F1 scores, we observe that our highest-performing approach still outperforms the state-of-the-art methods in the Hate class of Davidson17 and Founta18. Note that our approach is ensemble construction for different hate speech styles but not optimizing performance score as in shared tasks by ensemble parameters (e.g. the backbone model and the number of ensemble members) and deep learning hyperparameters (e.g. learning rate). We argue that focusing on the optimization of both ensemble learning and different hate speech styles together can further improve the performance of hate speech detection.

https://doi.org/10.1017/nlp.2024.44 Published online by Cambridge University Press

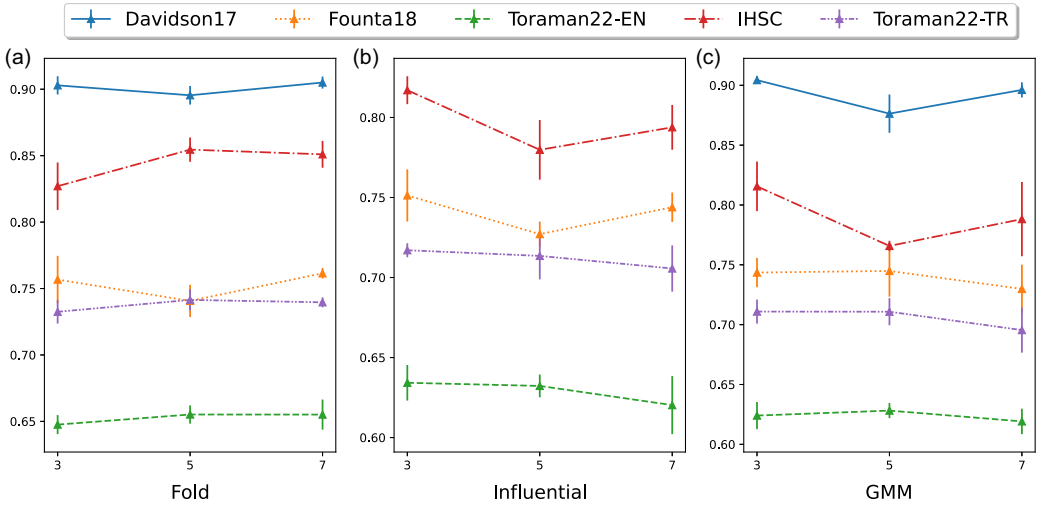

Figure 5. Performance comparison of a varying number of ensemble members in terms of weighted F1 score in all datasets for the Fold, Influential, and GMM ensemble methods from (a) to (c), respectively.

5.2 Varying number of ensemble members

The number of ensemble members or base models can be a challenging parameter such that varying the number of ensemble members can significantly change the performance results (Bornheim, Grieger, and Bialonski 2021). In Table 3, we set the number of ensemble members to five, except that the number of ensemble members of the Topic method varies from five to six topics as reported in Table 2. In this subsection, we compare the results of varying numbers of ensemble members from three to seven to understand its effect on the performance of our approach for hate speech detection. We report the performance in terms of weighted F1 score for each dataset in Fig. 5.

We apply the same hyperparameters as in Section 4, except for the number of ensemble members. We vary the number of ensemble members by using three, five, and seven members in this additional experiment. We select a baseline ensemble approach (i.e. Fold) and two proposed ensemble methods (i.e. GMM and Influential).

The performance does not deteriorate as the number of ensemble members increases in the majority of the cases for the Fold method. Since the Fold method splits the data and then merges a majority of the splits to train an ensemble member, the higher number of members, such as seven, can have more data instances, and therefore perform no worse than a smaller number of members.

In contrast to the results in the Fold method, the performance score of the GMM and Influential methods mostly decreases as the number of ensemble members increases. This observation supports the benefit of our ensemble approach that accounts for different aspects of hate speech. We argue that writing styles and topics are not captured well in seven members since the number of training instances might not be sufficient to cover a number of writing styles and topics inherently.

5.3 Writing style analysis

In our experiments, we use several features to analyze the intrinsic features of the writing style. Zangerle *et al.* (2023) have a shared task that aims to distinguish the authors in multi-author documents where no other information is in the corpus but only the raw text. The writing style is based on the stylistic fingerprints in the text such as lexical features (n-grams) (Stein *et al.* 2009; Schwartz *et al.* 2017), word frequencies (Holmes 1998), average word and sequence length (Zheng

https://doi.org/10.1017/nlp.2024.44 Published online by Cambridge University Press

et al. 2006; Schwartz *et al.* 2017), and structural features (Zheng *et al.* 2006). In our work, the motivation of this analysis is to understand the writing style of different ensemble members. We argue that our ensemble approach can reflect different aspects of hate speech, and writing style can be an important factor in this regard.

We analyze the results of a baseline ensemble method (i.e. Fold) and two of our proposed ensemble methods (i.e. Influential and GMM) in terms of word n-grams, specifically bigrams when n equals two. In Table 5, we concatenate the instances of each ensemble member and extract bigrams. We obtain a TF-IDF vector (Pedregosa *et al.* 2011) for each ensemble member using the most frequent 50 bigrams. We then calculate cosine similarity between TF-IDF vectors of two ensemble members and report the scores for all pairs of members in Table 5. The results show that the similarity scores between the pairs of ensemble members are much smaller in the Influential and GMM methods compared to the Fold method. In other words, the members in the Influential and GMM methods have much more variety in writing style compared to the members in the Fold method. Moreover, the GMM method has more variety in writing style compared to the Influential method, probably due to the fact that we separate GMM members based on embedding vectors.

Furthermore, we analyze the results of our proposed ensemble methods (i.e. GMM, Influential, and Topic) in terms of additional writing style features that may differ between the data splits used in training, such as the average number of words, the average number of URLs, and the average number of emojis. We also list the most frequent bigrams in each ensemble member. In this analysis, we report on the Topic, Influential, and GMM methods on the Davidson17 and Founta18 datasets. We select these datasets since the Topic method performs the highest in Davidson17, and GMM in Founta18, as observed in Table 3.

5.3.1 Analysis of Davidson17

The results are given in Tables 6 and 7 for the Topic and GMM methods, respectively.

In all ensemble methods, we observe that the average number of words, the average word length, the average number of URLs, and the average number of emojis differ among ensemble members. We argue that these differences could represent different user types or hate styles.

In terms of word n-grams, the same or similar bigrams are observed in the GMM method, compared to the Topic method. For instance, the "b*tch" and "*ss" words mostly exist in all ensemble members. That is, the splits of the GMM method seem to have similar writing styles in terms of bigrams. The worse performance of the GMM method in Table 3 can be attributed to this observation in this dataset.

5.3.2 Analysis of Founta18

The results are given in Tables 8, 9, and 10 for the Topic, Influential, and GMM methods, respectively.

In terms of word n-grams, we observe that all methods have a variety of bigrams in ensemble members. In contrast to the analysis of the Davidson17 dataset, there are no significant overlapping bigrams in the ensemble members of the GMM method. Note that the GMM method performs better than the Topic and Influential methods in the Founta18 dataset, while it performs poorly in the Davidson17 dataset. This analysis supports the performance results and may suggest that the performance of the GMM method depends on the dataset.

Another reason for the success of the GMM method could be the average statistics of writing style. Although the average statistics differ in all methods, the degree of this difference is larger in the GMM method, since the standard deviations of the average statistics are mostly higher in the GMM method. The standard deviations of the average number of words, word length, URLs, and emojis are 0.65, 0.20, 0.15, 0.18 in the Topic method; 0.31, 0.09, 0.11, 0.10 in the Influential method; and 0.68, 0.29, 0,33, and 0.17 in the GMM method, respectively.

https://doi.org/10.1017/nlp.2024.44 Published online by Cambridge University Press

https://doi.org/10.1017/nlp.2024.44 Published online by Cambridge University Press

Table 5. Writing style comparison in terms of cosine similarity scores between bigram TF-IDF vectors of each ensemble member. The table is colored according to the scores in each cell: The higher the similarity score, the color gets darker. The Davidson17 dataset has no influential data. The similarity matrices are symmetric

		Fold					Influential					GMM				
Dataset	Members	1	2	3	4	5	1	2	3	4	5	1	2	3	4	**5**
Davidson17	1	1.000	0.993	0.988	0.932	0.991	-	-	-	-	-	1.000	0.294	0.532	0.763	0.371
	2		1.000	0.987	0.929	0.990	-	-	-	-	-		1.000	0.446	0.337	0.359
	3			1.000	0.952	0.989	-	-	-	-	-			1.000	0.455	0.629
	4				1.000	0.944	-	-	-	-	-				1.000	0.391
	5					1.000	-	-	-	-	-					1.000
Founta18	1	1.000	0.985	0.957	0.891	0.866	1.000	0.256	0.175	0.140	0.015	1.000	0.002	0.012	0.006	0.459
	2		1.000	0.970	0.909	0.885		1.000	0.394	0.334	0.027		1.000	0.037	0.294	0.005
	3			1.000	0.965	0.947			1.000	0.791	0.038			1.000	0.032	0.027
	4				1.000	0.987				1.000	0.039				1.000	0.009
	5					1.000					1.000					1.000
Toraman22	1	1.000	0.987	0.984	0.985	0.988	1.000	0.757	0.571	0.404	0.622	1.000	0.142	0.157	0.522	0.345
	2		1.000	0.984	0.985	0.988		1.000	0.715	0.474	0.705		1.000	0.086	0.132	0.083
	3			1.000	0.985	0.985			1.000	0.416	0.596			1.000	0.094	0.052
	4				1.000	0.985				1.000	0.489				1.000	0.420
	5					1.000					1.000					1.000

Table 6. Writing style analysis for the Topic method in the Davidson17 dataset

	Member 1	Member 2	Member 3	Member 4	Member 5	Member 6
Most frequent bigrams	fuck wit	derek jeter	charlie crist	blessjesus amos	hoe ass	bad bitches
	can't fuck	new york	looks like	hoes need	look like	look like
	lil ass	relationship goals	rick scott	amos two	act like	bitches like
	ass fuckin'	yall' hoes'	eat lot	tow walk	pussy ass	yall bitches
	ass hoe	charlie strong	sorryimalex back	walk together	like bitches	ass bitches
Words per tweet	7.49	8.46	9.63	8.84	7.57	6.81
Avg. word length	5.75	6.12	6.51	6.33	5.81	5.80
URLs per tweet	0.09	0.20	0.27	0.20	0.11	0.10
Emojis per tweet	0.35	0.32	0.51	0.27	0.32	0.44

Table 7. Writing style analysis for the GMM method in the Davidson17 dataset

	Member 1	Member 2	Member 3	Member 4	Member 5
Most frequent bigrams	bitch ass	like bitch	bad bitches	bitch ass	like bitch
	ass bitch	bitch ass	like bitch	look like	bitch ass
	little bitch	ass bitch	bad bitch	ass nigga	ass nigga
	like bitch	bad bitches	fuck bitch	bad bitch	fuck bitch
	bad bitch	bad bitch	bitch ass	like bitch	ass bitch
Words per tweet	6.33	8.99	5.98	9.36	9.57
Avg. word length	6.18	5.83	5.16	6.05	5.11
URLs per tweet	0.08	0.16	0.00	0.23	0.01
Emojis per tweet	0.02	1.91	0.01	0.05	0.06

Table 8. Writing style analysis for the Founta18 dataset

	Member 1	Member 2	Member 3	Member 4	Member 5	Member 6
Most frequent bigrams	via c0nvey	even want	thebloodshow letting	parissaxo tired	huge weekend	Islamic state
	susan rice	cerromezone shit	letting know	tired feminist	weekend check	andyrichter sequel
	chemical attack	shit fuckin	know protesting	feminist bitches	check highlights	sequel looks
	efe efe	fuckin' cryinggggggg	protesting laws	bitches disgusting	new task	looks fucking
	abc news	cryinggggggg cuz	laws shoot	bad bitches	task unlocked	fucking terrifying
Words per tweet	10.43	10.68	9.83	9.47	10.58	9.91
Avg. word length	6.74	6.30	6.14	6.15	6.44	6.67
URLs per tweet	0.64	0.62	0.41	0.48	0.84	0.64
Emojis per tweet	0.13	0.48	0.48	0.61	0.24	0.16

https://doi.org/10.1017/nlp.2024.44 Published online by Cambridge University Press

Table 9. Writing style analysis for the Influential method in the Founta18 dataset

	Member 1	Member 2	Member 3	Member 4	Member 5
Most frequent bigrams	huge weekend	thebloodshow letting	stupid ass	abc news	via c0nvey
	weekend football	letting know	ass stupid	via c0nvey	efe efe
	football check	know protesting	thebloodshow letting	news reports	Donald Trump
	check highlights	protesting laws	letting know	brianklaas abc	white house
	new task	laws shoot	know protesting	reports syria	andyrichter jesus
Words per tweet	10.44	9.90	10.57	10.79	10.72
Avg. word length	6.40	6.26	6.44	6.50	6.50
URLs per tweet	0.86	0.58	0.62	0.58	0.54
Emojis per tweet	0.23	0.41	0.43	0.28	0.18

Table 10. Writing style analysis for the GMM method in the Founta18 dataset

	Member 1	Member 2	Member 3	Member 4	Member 5
Most frequent bigrams	via c0nvey	ass stupid	stupid ass	parissaxo tired	via c0nvey
	white house	thebloodshow letting	fucking stupid	tired feminist	white house
	chemical attack	letting know	bad bitch	feminist bitches	Donald Trump
	Donald Trump	know protesting	really fucking	bitches disgusting	chemical attack
	efe efe	bad bitch	protesting laws	new task	run idiot
Words per tweet	11.08	10.17	10.28	9.31	11.17
Avg. word length	6.35	6.68	6.08	6.92	6.40
URLs per tweet	0.97	0.69	0.14	1.07	0.88
Emojis per tweet	0.06	0.50	0.28	0.27	0.05

In the Influential method, ensemble members are obtained by splitting the data according to the influential scores. The influential score decreases as the member number increases in this table. That is, the users in the data portion of the first ensemble member (i.e. Member 1 in the table) have a higher influential score compared to the influential scores of the last ensemble member. We observe that the average numbers of words and word length are longer in less influential groups. On the other hand, the average number of URLs and emojis is higher in the first member (more influential) compared to the last one (less influential).

5.4 Selection of backbone model

HateBERT (Caselli *et al.* 2021) and fBERT (Sarkar *et al.* 2021) are two instances of Transformer-based models that are fine-tuned for hate speech detection. However, we use a smaller and faster model, DistilBERT (Sanh *et al.* 2019), as a backbone model in our experiments, since the number of training models is very high due to ensemble learning and 5-fold splitting.

https://doi.org/10.1017/nlp.2024.44 Published online by Cambridge University Press

Table 11. Comparison of Backbone Models. An average of 5-fold is reported in terms of the F1 score. The highest scores are given in bold. Davidson17, Founta18, and Toraman22-EN are shortened as D17, F18, and T22, respectively

	Method	Normal	Offensive/Abusive	Spam	Hate	Weighted F1
D17	Single-Text-DistilBERT	**0.845 ± 0.016**	0.913 ± 0.017	-	0.444 ± 0.044	0.872 ± 0.018
	Single-Text-HateBERT	0.822 ± 0.007	0.940 ± 0.000	-	**0.460 ± 0.023**	0.896 ± 0.005
	Single-Text-fBERT	0.844 ± 0.010	**0.946 ± 0.005**	-	0.460 ± 0.033	**0.900 ± 0.006**
F18	Single-Text-DistilBERT	0.744 ± 0.055	0.800 ± 0.010	**0.505 ± 0.031**	**0.460 ± 0.017**	0.706 ± 0.031
	Single-Text-HateBERT	0.842 ± 0.007	0.802 ± 0.012	0.456 ± 0.031	0.438 ± 0.021	0.752 ± 0.007
	Single-Text-fBERT	**0.848 ± 0.004**	**0.814 ± 0.008**	0.444 ± 0.023	0.448 ± 0.043	**0.758 ± 0.007**
T22	Single-Text-DistilBERT	0.696 ± 0.012	**0.604 ± 0.020**	-	0.626 ± 0.030	0.642 ± 0.013
	Single-Text-HateBERT	0.694 ± 0.012	0.582 ± 0.007	-	0.618 ± 0.004	0.632 ± 0.007
	Single-Text-fBERT	**0.704 ± 0.010**	0.594 ± 0.010	-	**0.634 ± 0.010**	**0.644 ± 0.005**

In Table 11, we validate our selection for the backbone model by comparing the performance of DistilBERT with those of fBERT and HateBERT. In this experiment, we use the Single-Text method to understand solely the performance of the backbone model rather than ensemble learning. The results show that DistilBERT has close performance (e.g. the Weighted F1 score in Toraman22) and sometimes better performance (e.g. the Hate class in Founta18) than HateBERT and fBERT, though DistilBERT is smaller in terms of the number of model parameters. We thereby argue that our selection of the backbone model does not have a significant impact on the model performance while comparing ensemble learning methods.

5.5 Limitations

We acknowledge a set of limitations to our study. Our experiments demonstrate ensemble learning for hate speech in the English, Turkish, and Italian languages. More experiments in different languages can be conducted to generalize the results to other languages. Similarly, different ensemble techniques can be employed to understand how they can capture different aspects of hate speech. In addition, there are other voting schemes in ensemble learning but we apply unweighted-average of softmax probabilities. For instance, one can use a dedicated weighting schema to give more importance to the predictions of particular ensemble members when test instances with the same style are processed.

Furthermore, we use a modified version of the datasets reported in the experiments. We observe that some of the performance scores are lower than those reported in the studies that use the whole dataset for experiments. For instance, Malik, Pang, and van den Hengel (2022) achieve the weighted F1 scores of 0.91 and 0.79 using Transformer-based models for the Davidson17 and Founta18 datasets, respectively. The Single-Text method based on the DistilBERT architecture yields weighted F1 scores of 0.87 and 0.71, respectively for the datasets in this study. Also, Toraman *et al.* (2022a) achieve the weighted scores of 0.83 and 0.78 for Toraman22-EN and TR, while we obtain the weighted F1 scores of 0.64 and 0.72, respectively. The reason for lower performance scores in our study is most probably related to the size of training data since we use a smaller part of these datasets after filtering the samples that do not include the topic keywords. In addition, ensemble learning is based on fine-tuning DistilBERT (Sanh *et al.* 2019). One can further train other language models or feature extraction methods to encode text data before applying ensemble methods.

https://doi.org/10.1017/nlp.2024.44 Published online by Cambridge University Press

6. Conclusion

We examine various ensemble learning techniques for hate speech detection. The motivation is to utilize multiple classifiers that reflect different aspects of hate speech. We conduct detailed experiments on five datasets covering multiple languages. Our experimental results, supported by statistical significance tests, show that the performance of hate speech detection is improved by capturing multiple aspects of hate speech in the majority of the datasets used in the experiments. We also provide a discussion on the number of ensemble members and writing styles exposed by different ensemble members.

In future work, we plan to generalize the experiments to other datasets and languages. Since cross-lingual transfer learning is a promising approach for hate speech detection (Bigoulaeva, Hangya, and Fraser 2021), one can further examine the opportunity for cross-lingual transfer learning with ensemble learning. We also plan to have a deeper analysis of ensemble members in order to understand how they reflect different aspects of hate speech. Alternative methods for capturing hate styles can also be studied, such as other user features rather than popularity for writing styles.

Acknowledgements. The work was partially done at Aselsan, Ankara, Turkey. We thank to Umitcan Sahin for providing the list of topic keywords used in Section 3.3.

References

Agarwal S. and Chowdary C.R. (2021). Combating hate speech using an adaptive ensemble learning model with a case study on COVID-19. *Expert Systems with Applications* **185**, 115632.

Aharoni R. and Goldberg Y. (2020). Unsupervised domain clusters in pretrained language models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, pp. 7747–7763. Online.

Alsafari S., Sadaoui S. and Mouhoub M. (2020). Deep learning ensembles for hate speech detection. In *IEEE 32nd International Conference on Tools with Artificial Intelligence (ICTAI)*, pp. 526–531.

Anusha M.D. and Shashirekha H.L. (2020). An ensemble model for hate speech and offensive content identification in Indo-European languages. In *Working Notes of FIRE. 2020 - Forum for Information Retrieval Evaluation*, December 16-20, 2020, Hyderabad, India, vol. **2826**, pp. 253–259. CEUR Workshop Proceedings.

Bailly A., Blanc C., Francis É., Guillotin T., Jamal F., Wakim B. and Roy P. (2022). Effects of dataset size and interactions on the prediction performance of logistic regression and deep learning models. *Computer Methods and Programs in Biomedicine* **213**, 106504.

Barbieri F., Camacho-Collados J., Anke L.E. and Neves L. (2020). Tweeteval: unified benchmark and comparative evaluation for tweet classification. In *Findings of the Association for Computational Linguistics: EMNLP 2020, Online Event, Volume EMNLP 2020 of Findings of ACL*, 16-20 November 2020. Association for Computational Linguistics, pp. 1644–1650.

Basile V., Bosco C., Fersini E., Nozza D., Patti V., Rangel Pardo F.M., Rosso P. and Sanguinetti M. (2019). SemEval-2019 task 5: multilingual detection of hate speech against immigrants and women in Twitter. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, Minneapolis, Minnesota, USA. Association for Computational Linguistics, pp. 54–63.

Bevendorff J., Chulvi B., Sarracén G.L.D.L.P., Kestemont M., Manjavacas E., Markov I., Mayerl M., Potthast M., Rangel F., Rosso P., Stamatatos E., Stein B., Wiegmann M., Wolska M. and Zangerle E. (2021). Overview of PAN 2021: authorship verification, profiling hate speech spreaders on twitter, and style change detection. In *12th International Conference of the CLEF Association (CLEF 2021)*. Springer.

Bigoulaeva I., Hangya V. and Fraser A. (2021). Cross-lingual transfer learning for hate speech detection. In *Proceedings of the First Workshop on Language Technology for Equality, Diversity and Inclusion*, pp. 15–25.

Bornheim T., Grieger N. and Bialonski S. (2021). FHAC at GermEval 2021: identifying German toxic, engaging, and fact-claiming comments with ensemble learning. In *Proceedings of the GermEval 2021 Shared Task on the Identification of Toxic, Engaging, and Fact-Claiming Comments*, Duesseldorf, Germany. Association for Computational Linguistics, pp. 105–111.

Breiman L. (1996). Bagging predictors. *Machine Learning* **24**(2), 123–140.

Burnap P. and Williams M.L. (2016). Us and them: identifying cyber hate on Twitter across multiple protected characteristics. *EPJ Data Science* **5**(1), 11.

Byman D.L. (2021). How Hateful Rhetoric Connects to Real-World Violence. Available at https://www.brookings.edu/blog/order-from-chaos/2021/04/09/how-hateful-rhetoric-connects-to-real-world-violence/ (accessed: 19 April 2023)

https://doi.org/10.1017/nlp.2024.44 Published online by Cambridge University Press

Caselli T., Basile V., Mitrović J. and Granitzer M. (2021). HateBERT: retraining BERT for abusive language detection in English. In *Proceedings of the 5th Workshop on Online Abuse and Harms (WOAH 2021)*. Association for Computational Linguistics, pp. 17–25. Online.

Chatzakou D., Kourtellis N., Blackburn J., Cristofaro E.D., Stringhini G. and Vakali A. (2017). Mean birds: detecting aggression and bullying on Twitter. In *Proceedings of the 2017 ACM on Web Science Conference, WebSci 2017*, June 25 - 28, 2017, Troy, NY, USA. ACM, pp. 13–22.

Davani A.M., Atari M., Kennedy B. and Dehghani M. (2023). Hate speech classifiers learn normative social stereotypes. *Transactions of the Association for Computational Linguistics* **11**, 300–319.

Davidson T., Warmsley D., Macy M. and Weber I. (2017). Automated hate speech detection and the problem of offensive language. In *Proceedings of the 11th International AAAI Conference on Web and Social Media, ICWSM'17*, pp. 512–515.

Devlin J., Chang M.-W., Lee K. and Toutanova K. (2019). BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies„ Volume 1 (Long and Short Papers)*, Minneapolis, Minnesota. Association for Computational Linguistics, pp. 4171–4186.

Dietterich T.G. (2000). Ensemble methods in machine learning. In *Multiple Classifier Systems, MCS 2000*, Berlin, Heidelberg. Springer Berlin Heidelberg, pp. 1–15.

Founta A., Djouvas C., Chatzakou D., Leontiadis I., Blackburn J., Stringhini G., Vakali A., Sirivianos M. and Kourtellis N. (2018). Large scale crowdsourcing and characterization of Twitter abusive behavior. In *Proceedings of the Twelfth International Conference on Web and Social Media, ICWSM 2018*, June 25-28, 2018, Stanford, California, USA. AAAI Press, pp. 491–500.

Ganaie M.A., Hu M., Malik A.K., Tanveer M. and Suganthan P.N. (2022). Ensemble deep learning: a review. *Engineering Applications of Artificial Intelligence* **115**, 105151.

Glazkova A., Kadantsev M. and Glazkov M. (2021). Fine-tuning of pre-trained transformers for hate offensive and profane content detection in english and marathi. In *Working Notes of FIRE. 2021 - Forum for Information Retrieval Evaluation*, December 13-17, 2021, Gandhinagar, India. CEUR-WS.org, vol. **3159**, pp. 52–62. CEUR Workshop Proceedings.

Gomes H.M., Barddal J.P., Enembreck F. and Bifet A. (2017). A survey on ensemble learning for data stream classification. *ACM Computing Surveys* **50**(2), 23:1–23:36.

Hegde A., Anusha M.D. and Shashirekha H.L. (2021). Ensemble based machine learning models for hate speech and offensive content identification. In *Working Notes of FIRE. 2021 - Forum for Information Retrieval Evaluation*, December 13-17, 2021, Gandhinagar, India, vol. **3159**, pp. 132–141. CEUR Workshop Proceedings.

Hochreiter S. and Schmidhuber J. (1997). Long short-term memory. *Neural Computation* **9**(8), 1735–1780.

Holmes D.I. (1998). The evolution of stylometry in humanities scholarship. *Literary and Linguistic Computing* **13**(3), 111–117.

Howard J. and Ruder S. (2018). Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018*, July 15-20, 2018, Melbourne, Australia. Association for Computational Linguistics, vol. **1**, pp. 328–339.

Kennedy B., Jin X., Mostafazadeh Davani A., Dehghani M. and Ren X. (2020). Contextualizing hate speech classifiers with post-hoc explanation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, pp. 5435–5442. Online.

Khan Y., Ma W. and Vosoughi S. (2021). Lone pine at SemEval-2021 Task 5: fine-grained detection of hate speech using BERToxic. In *Proceedings of the 15th International Workshop on Semantic Evaluation, SemEval@ACL/IJCNLP 2021*, August 5-6, 2021, Virtual Event/Bangkok, Thailand. Association for Computational Linguistics, pp. 967–973.

Kim Y. (2014). Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014: A Meeting of SIGDAT, a Special Interest Group of the ACL*, October 25-29, 2014, Doha, Qatar. ACL, pp. 1746–1751.

Kirk H.R., Birhane A., Vidgen B. and Derczynski L. (2022). Handling and presenting harmful text in NLP research. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics, pp. 497–510.

Kumar A., Roy P.K. and Saumya S. (2021). An ensemble approach for hate and offensive language identification in english and indo-aryan languages. In *Working Notes of FIRE. 2021 - Forum for Information Retrieval Evaluation*, December 13-17, 2021, Gandhinagar, India. CEUR-WS.org, vol. **3159**, pp. 439–445. CEUR Workshop Proceedings.

Kumaresan K. and Vidanage K. (2019). HateSense: tackling ambiguity in hate speech detection. In *2019 National Information Technology Conference (NITC)*, pp. 20–26.

Laan M., Polley E. and Hubbard A. (2007). Super learner. *Statistical Applications in Genetics and Molecular Biology* **6**(1), Article25.

Liu Y., Ott M., Goyal N., Du J., Joshi M., Chen D., Levy O., Lewis M., Zettlemoyer L. and Stoyanov V. (2019). Roberta: a robustly optimized BERT pretraining approach. CoRR, abs/1907.11692.

Liu Y. and Yao X. (1999). Ensemble learning via negative correlation. *Neural Networks* **12**(10), 1399–1404.

Lloyd S. (1982). Least squares quantization in PCM. *IEEE Transactions n Information Theory* **28**(2), 129–137.

https://doi.org/10.1017/nlp.2024.44 Published online by Cambridge University Press

Lu J., **Lin H.**, **Zhang X.**, **Li Z.**, **Zhang T.**, **Zong L.**, **Ma F. and Xu B.** (2023). Hate speech detection via dual contrastive learning. *IEEE/ACM Transactions on Audio, Speech, and Language Processing* **31**, 2787–2795.

Magnossão de Paula A.F., **Rosso P.**, **Bensalem I. and Zaghouani W.** (2022). UPV at the Arabic hate speech 2022 shared task: offensive language and hate speech detection using transformers and ensemble models. In *Proceedings of the 5th Workshop on Open-Source Arabic Corpora and Processing Tools with Shared Tasks on Qur'an QA and Fine-Grained Hate Speech Detection*, Marseille, France. European Language Resources Association, pp. 181–185.

Malik J.S., **Pang G. and van den Hengel A.** (2022). Deep learning for hate speech detection: a comparative study. CoRR, abs/2202.09517.

Mandl T., **Modha S.**, **Shahi G.K.**, **Madhu H.**, **Satapara S.**, **Majumder P.**, **Schäfer J.**, **Ranasinghe T.**, **Zampieri M.**, **Nandini D. and Jaiswal A.K.** (2021). Overview of the HASOC subtrack at FIRE 2021: hatespeech and offensive content identification in english and indo-aryan languages. In *Working Notes of FIRE. 2021 - Forum for Information Retrieval Evaluation*, December 13-17, 2021, Gandhinagar, India, vol. **3159**, pp. 1–19. CEUR Workshop Proceedings.

Martin L., **Sintsova V. and Pu P.** (2014). Are influential writers more objective?: an analysis of emotionality in review comments. In *23rd International World Wide Web Conference, WWW'14, Companion Volume*, April 7-11, 2014, Seoul, Republic of Korea. ACM, 799–804.

Mathew B., **Saha P.**, **Yimam S.M.**, **Biemann C.**, **Goyal P. and Mukherjee A.** (2021). HateXplain: a benchmark dataset for explainable hate speech detection. *Proceedings of the AAAI Conference on Artificial Intelligence* **35**(17), 14867–14875.

McInnes L. and Healy J. (2018). UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. CoRR, abs/1802.03426.

Min C., **Lin H.**, **Li X.**, **Zhao H.**, **Lu J.**, **Yang L. and Xu B.** (2023). Finding hate speech with auxiliary emotion detection from self-training multi-label learning perspective. *Information Fusion* **96**(C), 214–223.

Mondal M., **Silva L.A. and Benevenuto F.** (2017). A measurement study of hate speech in social media. In *Proceedings of the 28th ACM Conference on Hypertext and Social Media, HT'17*, New York, NY, USA. Association for Computing Machinery, pp. 85–94.

Mou G., **Ye P. and Lee K.** (2020). SWE2: SubWord enriched and significant word emphasized framework for hate speech detection. In *CIKM'20: The 29th ACM International Conference on Information and Knowledge Management, Virtual Event*, October 19-23, 2020, Ireland. ACM, pp. 1145–1154.

Mubarak H., **Al-Khalifa H. and Al-Thubaity A.** (2022). Overview of OSACT5 shared task on Arabic offensive language and hate speech detection. In *Proceedings of the 5th Workshop on Open-Source Arabic Corpora and Processing Tools with Shared Tasks on Qur'an QA and Fine-Grained Hate Speech Detection*, Marseille, France. European Language Resources Association, pp. 162–166.

Mutanga R.T., **Naicker N. and Olugbara O.O.** (2022). Detecting hate speech on twitter network using ensemble machine learning. *International Journal of Advanced Computer Science and Applications* **13**(3), 331–339.

Nikolov A. and Radivchev V. (2019). Nikolov-radivchev at semeval-2019 task 6: offensive tweet classification with BERT and ensembles. In *Proceedings of the 13th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT 2019*, June 6-7, 2019, Minneapolis, MN, USA. Association for Computational Linguistics, pp. 691–695.

Nobata C., **Tetreault J.R.**, **Thomas A.O.**, **Mehdad Y. and Chang Y.** (2016). Abusive language detection in online user content. In *Proceedings of the 25th International Conference on World Wide Web*.

Opitz D.W. and Maclin R. (1999). Popular ensemble methods: an empirical study. *Journal of Artificial Intelligence Research* **11**, 169–198.

Pavlopoulos J., **Sorensen J.**, **Laugier L. and Androutsopoulos I.** (2021). Semeval-2021 task 5: toxic spans detection. In *Proceedings of the 15th International Workshop on Semantic Evaluation, SemEval@ACL/IJCNLP 2021*, August 5-6, 2021, Virtual Event/Bangkok, Thailand. Association for Computational Linguistics, pp. 59–69.

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

Pennington J., **Socher R. and Manning C.D.** (2014). Glove: global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014: A Meeting of SIGDAT, a Special Interest Group of the ACL*, October 25-29, 2014, Doha, Qatar. ACL, pp. 1532–1543.

Ranasinghe T. and Zampieri M. (2021). MUDES: multilingual detection of offensive spans. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Demonstrations, NAACL-HLT 2021, Online*, June 6-11, 2021. Association for Computational Linguistics. pp. 144–152. Online.

Reynolds D.A. (2009). Gaussian mixture models. In *Encyclopedia of Biometrics*. Springer US, pp. 659–663.

Rosenthal S., **Atanasova P.**, **Karadzhov G.**, **Zampieri M. and Nakov P.** (2021). SOLID: a large-scale semi-supervised dataset for offensive language identification. In *Findings of the Association for Computational Linguistics: ACL/IJCNLP 2021, Online Event, August 1-6, 2021, volume ACL/IJCNLP 2021 of Findings of ACL* . Association for Computational Linguistics, pp. 915–928.

https://doi.org/10.1017/nlp.2024.44 Published online by Cambridge University Press

Röttger P., **Vidgen B.**, **Nguyen D.**, **Waseem Z.**, **Margetts H. and Pierrehumbert J.** (2021). HateCheck: functional tests for hate speech detection models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Association for Computational Linguistics, pp. 41–58. Online.

Sanguinetti M., **Comandini G.**, **Nuovo E.D.**, **Frenda S.**, **Stranisci M.**, **Bosco C.**, **Caselli T.**, **Patti V. and Russo I.** (2020). HaSpeeDe 2 @ EVALITA2020: overview of the EVALITA, hate speech detection task. In *Proceedings of the Seventh Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2020), Online event*, December 17th, 2020. CEUR-WS.org, vol. **2765**, CEUR Workshop Proceedings, Online event.

Sanguinetti M., **Poletto F.**, **Bosco C.**, **Patti V. and Stranisci M.** (2018). An Italian Twitter Corpus of hate speech against immigrants. In *Proceedings of the 11th Conference on Language Resources and Evaluation (LREC2018)*, Miyazaki, Japan, pp. 2798–2895.

Sanh V., **Debut L.**, **Chaumond J. and Wolf T.** (2019). DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. CoRR, abs/1910.01108.

Sarkar D., **Zampieri M.**, **Ranasinghe T. and Ororbia II A. G.** (2021). fbert: a neural transformer for identifying offensive content. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, 16-20 November, 2021, Virtual Event/Punta Cana, Dominican Republic. Association for Computational Linguistics, pp. 1792–1798

Schapire R.E. (1990). The strength of weak learnability. *Machine Learning* **5**(2), 197–227.

Schmidt A. and Wiegand M. (2017). A survey on hate speech detection using natural language processing. In *Proceedings of the Fifth International Workshop on Natural Language Processing for Social Media*, Valencia, Spain. Association for Computational Linguistics, pp. 1–10.

Schwartz R., **Sap M.**, **Konstas I.**, **Zilles L.**, **Choi Y. and Smith N.A.** (2017). The effect of different writing tasks on linguistic style: a case study of the ROC story cloze task. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, Vancouver, Canada. Association for Computational Linguistics, pp. 15–25.

Shi X., **Mueller J.**, **Erickson N.**, **Li M. and Smola A.** (2021). Multimodal automl on structured tables with text fields. In *8th ICML Workshop on Automated Machine Learning (AutoML)*.

Sia S., **Dalmia A. and Mielke S.J.** (2020). Tired of topic models? Clusters of pretrained word embeddings make for fast and good topics too!. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, pp. 1728–1736. Online.

Siino M., **Di Nuovo E.**, **Tinnirello I. and La Cascia M.** (2021). Detection of hate speech spreaders using convolutional neural networks—Notebook for PAN at CLEF 2021. In *CLEF. 2021 Labs and Workshops, Notebook Papers*. CEUR-WS.org.

Srivastava N., **Hinton G.**, **Krizhevsky A.**, **Sutskever I. and Salakhutdinov R.** (2014). Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research* **15**(56), 1929–1958.

Stein B., **Rosso P.**, **Stamatatos E.**, **Koppel M. and Agirre E.** (2009). 3rd pan workshop on uncovering plagiarism, authorship and social software misuse. In *25th Annual Conference of the Spanish Society for Natural Language Processing (SEPLN)*, pp. 1–77.

Tekiroğlu S.S., **Chung Y.-L. and Guerini M.** (2020). Generating counter narratives against online hate speech: data and strategies. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, pp. 1177–1190. Online.

Toraman C. and Can F. (2012). Squeezing the ensemble pruning: faster and more accurate categorization for news portals. In *European Conference on Information Retrieval*. Springer, pp. 508–511.

Toraman C., **Sahinuç F. and Yilmaz E.H.** (2022a). Large-scale hate speech detection with cross-domain transfer. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference, LREC 2022*, 20-25 June 2022, Marseille, France. European Language Resources Association, pp. 2215–2225.

Toraman C., **Sahinuç F.**, **Yilmaz E.H. and Akkaya I.B.** (2022b). Understanding social engagements: a comparative analysis of user and text features in Twitter. *Social Network Analysis and Mining* **12**(1), 47.

Tula D., **Potluri P.**, **Ms S.**, **Doddapaneni S.**, **Sahu P.**, **Sukumaran R. and Patwa P.** (2021). Bitions@DravidianLangTech-EACL2021: ensemble of multilingual language models with pseudo labeling for offence detection in Dravidian languages. In *Proceedings of the First Workshop on Speech and Language Technologies for Dravidian Languages*, Kyiv. Association for Computational Linguistics, pp. 291–299.

Turban C. and Kruschwitz U. (2022). Tackling irony detection using ensemble classifiers. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, Marseille, France. European Language Resources Association, pp. 6976–6984.

Unsvåg E.F. and Gambäck B. (2018). The effects of user features on Twitter hate speech detection. In *Proceedings of the 2nd Workshop on Abusive Language Online, ALW@EMNLP 2018*, October 31, 2018, Brussels, Belgium. Association for Computational Linguistics, pp. 75–85.

Vaswani A., **Shazeer N.**, **Parmar N.**, **Uszkoreit J.**, **Jones L.**, **Gomez A.N.**, **Kaiser L.u. and Polosukhin I.** (2017). Attention is all you need. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., vol. **30**.

Wan L., **Zeiler M.D.**, **Zhang S.**, **LeCun Y. and Fergus R.** (2013). Regularization of neural networks using DropConnect. In *Proceedings of the 30th International Conference on Machine Learning, ICML 2013*, 16-21 June 2013, Atlanta, GA, USA, vol. **28**, pp. 1058–1066. JMLR Workshop and Conference Proceedings.

https://doi.org/10.1017/nlp.2024.44 Published online by Cambridge University Press

Waseem Z. (2016). Are you a racist or am I seeing things? annotator influence on hate speech detection on Twitter. In *Proceedings of the First Workshop on NLP and Computational Social Science, NLP+CSS@EMNLP 2016*, November 5, 2016, Austin, TX, USA. Association for Computational Linguistics, pp. 138–142.

Wiedemann G., **Yimam S.M. and Biemann C.** (2020). UHH-LT at semeval-2020 task 12: fine-tuning of pre-trained transformer networks for offensive language detection. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation, SemEval@COLING 2020*, December 12-13, 2020, Barcelona (online). International Committee for Computational Linguistics, pp. 1638–1644.

Wolf T., **Debut L.**, **Sanh V.**, **Chaumond J.**, **Delangue C.**, **Moi A.**, **Cistac P.**, **Rault T.**, **Louf R.**, **Funtowicz M.**, **Davison J.**, **Shleifer S.**, **von Platen P.**, **Ma C.**, **Jernite Y.**, **Plu J.**, **Xu C.**, **Le Scao T.**, **Gugger S.**, **Drame M.**, **Lhoest Q. and Rush A.** (2020). Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Association for Computational Linguistics, pp. 38–45. Online.

Zampieri M., **Malmasi S.**, **Nakov P.**, **Rosenthal S.**, **Farra N. and Kumar R.** (2019). Semeval-2019 task 6: identifying and categorizing offensive language in social media (offenseval). In *Proceedings of the 13th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT 2019*, June 6-7, 2019, Minneapolis, MN, USA. Association for Computational Linguistics, pp. 75–86.

Zampieri M., **Nakov P.**, **Rosenthal S.**, **Atanasova P.**, **Karadzhov G.**, **Mubarak H.**, **Derczynski L.**, **Pitenis Z. and Çöltekin Ç.** (2020). Semeval-2020 task 12: multilingual offensive language identification in social media (offenseval 2020). In *Proceedings of the Fourteenth Workshop on Semantic Evaluation, SemEval@COLING 2020*, December 12-13, 2020, Barcelona (online). International Committee for Computational Linguistics, pp. 1425–1447.

Zangerle E., **Mayerl M.**, **Potthast M. and Stein B.** (2023). Overview of the multi-author writing style analysis task at PAN *2023*. In *Working Notes of the Conference and Labs of the Evaluation Forum (CLEF 2023)*, Thessaloniki, Greece. CEUR-WS.org, vol. **3497**, pp. 2513–2522. CEUR Workshop Proceedings.

Zhang H., **Wojatzki M.**, **Horsmann T. and Zesch T.** (2019). ltl.uni-due at SemEval-2019 Task 5: simple but effective lexico-semantic features for detecting hate speech in Twitter. In *Proceedings of the 13th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT 2019*, June 6-7, 2019, Minneapolis, MN, USA. Association for Computational Linguistics, pp. 441–446.

Zheng R., **Li J.**, **Chen H. and Huang Z.** (2006). A framework for authorship identification of online messages: writing-style features and classification techniques. *Journal of the Association for Information Science and Technology* **57**(3), 378–393.

Zhou L., **Caines A.**, **Pete I. and Hutchings A.** (2022). Automated hate speech detection and span extraction in underground hacking and extremist forums. *Natural Language Engineering* 29(5), 1247–1274.

Zimmerman S., **Kruschwitz U. and Fox C.** (2018). Improving hate speech detection with deep learning ensembles. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation, LREC 2018*, May 7-12, 2018, Miyazaki, Japan. European Language Resources Association.

Cite this article: Kucukkaya IE and Toraman C. Constructing ensembles for hate speech detection. *Natural Language Processing* https://doi.org/10.1017/nlp.2024.44