World Wide Web (2022) 25:1129–1150
https://doi.org/10.1007/s11280-022-01025-2

Token replacement-based data augmentation methods for hate speech detection

Kosisochukwu Judith Madukwe[1] · **Xiaoying Gao**[1] · **Bing Xue**[1]

Received: 21 June 2021 / Revised: 17 December 2021 / Accepted: 9 February 2022 /
Published online: 1 March 2022

Abstract
Hate speech detection mostly involves the use of text data. This data, usually sourced from various social media platforms, have been known to be plagued with numerous issues that result in a reduction of its quality and hence, the quality of the trained models. Some of these issues are the lack of diversity and the diminutive class of interest in the dataset which results in overfitted models that do not generalize well on other or newly collected data. The different ways of handling these issues include augmenting the data with diverse samples, engineering non-redundant features or designing robust classification models. In this study, the focus is on the data augmentation aspect. Data augmentation is a popular method for improving the quality of existing datasets by generating synthetic samples that mimic the distribution of the original samples. There is a lack of extensive studies on how hate speech texts respond to varying textual data augmentation techniques and methods. Specifically, we provide further insight into the token replacement method of textual data augmentation by performing empirical studies that investigate which embedding method(s) is a robust source of synonym for replacement process, what effective method(s) can be used to select words to be replaced, and how to confirm if the label within each class is preserved. Our proposed methods, validated on two commonly used hate speech datasets affected by a known lack of diversity and diminutive class of interest issues, significantly improve classification performance and provides insights into token replacement methods.

Keywords Hate speech data · Data augmentation · Token substitution · Word replacement · Data generation · Text data

This article belongs to the Topical Collection: Special Issue on Web Intelligence = Artificial Intelligence in the Connected World
Guest Editors: Yuefeng Li, Amit Sheth, Athena Vakali, and Xiaohui Tao

✉ Kosisochukwu Judith Madukwe
kosisochukwu.madukwe@ecs.vuw.ac.nz

Xiaoying Gao
xiaoying.gao@ecs.vuw.ac.nz

Bing Xue
bing.xue@ecs.vuw.ac.nz

[1] School of Engineering and Computer Science, Victoria University of Wellington, PO Box 600, 6012 Wellington, New Zealand

1 Introduction

The research area of hate speech detection framed as a text classification task has come a long way in recent years. The outpour of research studies tackle different aspects of the problem and the resulting proposed solutions include the manual extraction of generic and task-specific features and the careful design of machine learning architectures [13, 20, 21, 23, 27]. Since most of the existing work have formulated the problem of detecting textual hate speech as a text classification task, it naturally requires access to a large source of clean, bias free, balanced textual data. However, this hasn't been completely feasible because of the following challenging reasons discussed more in depth in [19]:

- The percentage of existing hate samples are minute compared to the non-hateful samples, hence there is a data scarcity.
- The data annotation process is very expensive, time consuming and leads to very little output for the already minute hateful class.
- The sensitive nature of hate speech samples have the possibility of adversely affecting the mental health of the annotators exposed to it.
- The subjective nature of hate speech makes the label output questionable.

As a result of these, the existing proposed solutions are usually built on inadequate available data. This, inadvertently leads to several problems such as:

- The trained models not sufficiently exposed to hateful class.
- Overfitting.
- Low generalization ability of the trained models.
- The trained models not sufficiently exposed to the varying aspects/kinds of hate speech and hence biased towards only certain aspects.

Data augmentation has the potential to reduce overfitting in models. Also, it could increase the size of the class of interest, which for hate speech detection is the hate class, and improve the diversity of the training set thus improving the generalization ability of the trained models. Data augmentation is a set of semantically invariant transformations [4] used as a regularization technique [33]. These transformations generate new data samples that must preserve the class labels. Data augmentation has been applied to image classification [32], speech recognition [28] and machine translation [9] and more recently text classification [40]. In image classification, techniques such as rotating, mirroring, resizing or cropping are easily applied. In some non-English languages with a rich case marking system, these methods (cropping and rotation) have been shown to be beneficial [31]. Unfortunately, due to sequential nature of text, most of these methods do not directly relay to text data. It is pertinent that the semantic and syntactic information in the text are preserved. For example, a clothing store review *"The cardigan is made of a thick fabric good for winter"*, by cropping could give *"This cardigan is made"* or by mirroring gives *"retinw rof doog cribaf kciht a fo edam si nagidrac siht"*. None of these new sentences make sense or retain the original information. Hence, specific augmentation techniques are required for text data. Figure 1 shows a summary of some of the well known textual data augmentation techniques. The techniques are divided based on whether they only change a local region in the original sentence or whether they globally affect the whole sentence (more details in Section 2).

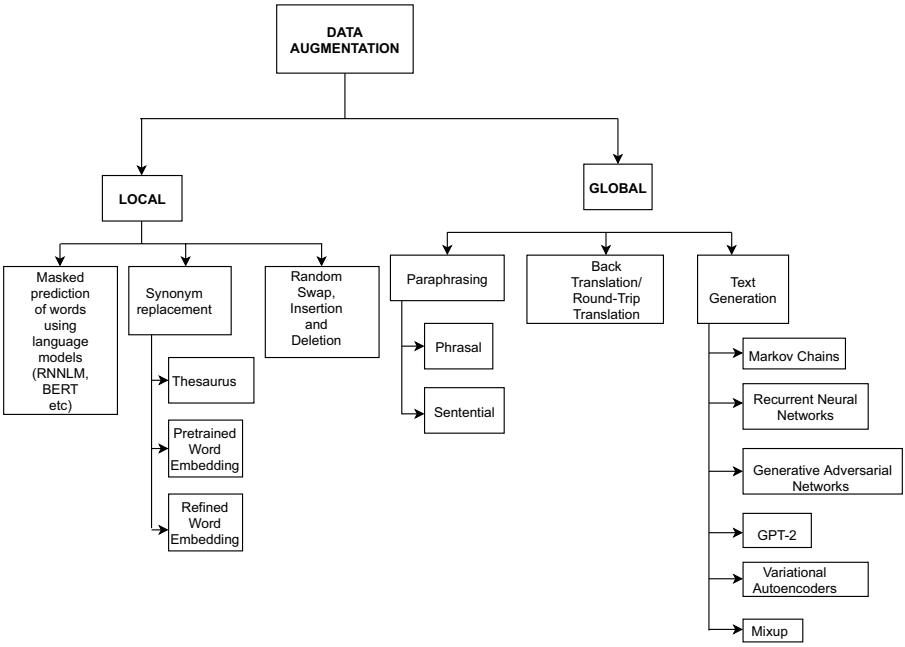

Fig. 1 Textual Data Augmentation Summary

For this study, we mostly focus on local data augmentation, specifically, token substitution/word replacement[1] data augmentation methods because it is one of the cost effective and easily accessible methods for textual data augmentation. Token substitution is a popular method which replaces a token in the sentence with its synonym. Albeit its popularity, it still has pertinent gaps especially in terms of hate speech data. Also, token replacement is the first step towards an incremental study on text data augmentation. Some of the gaps in this area that need to be further explored include:

1. **Source of Synonyms:** Word embedding methods such as Word2Vec [22] or GloVe [26], which are the widely used methods for producing synonyms, are limited by the learning method used to generate it. In this learning method, co-occurring words in the training documents end up in the same vicinity in the vector space. Hence, antonyms of a keyword could be in the same vicinity to that keyword because they usually co-occur in documents. For example, the word *"black"* could exist very closely to the word *"white"* and can be swapped for one another during a synonym replacement process. Thus, it is clear that using this embedding for token substitution/synonym replacement will generate misleading instances with higher chance of a label change. This is especially true for the hate speech detection task where using an antonym to replace a word is not considered a semantically invariant transformation [4]. *We will investigate what other embedding methods can produce better synonyms and how they perform in*

[1] For the rest of this study, token" and "word" will be used interchangeably. Also, "substitution" and "replacement" will be used interchangeably.

comparison with widely used embedding methods based solely on co-occurring words such as Word2Vec.

2. **Substituted Word Selection:** Existing token substitution methods randomly choose word(s) in a sentence to be substituted [45] or replace words with a specific parts-of-speech (PoS) tag [4, 17]. It is clear that words in a sentence have different levels of importance in that sentence, hence, the replacement of a specific word will have an effect different from the replacement of another word. Therefore, naturally, we ask *"What alternative methods can be applied to effectively select the best words to be substituted during the token replacement process?"*
3. **Homogeneity vs Heterogeneity:** Token substitution mostly uses a pre-trained embedding to find the new word to be used as a substitute, calculated using a distance measure such as cosine similarity. The same pre-trained embedding is often used as the first layer of a deep classification model. *Would the performance deteriorate or remain unchanged if the pre-trained embeddings used are kept homogeneous or heterogeneous?*. For example, using Word2Vec for cosine similarity based synonym replacement and using GloVe as the first layer of the deep classification model is considered a heterogeneous setting i.e. it is mixed while a homogeneous setting will include using a Word2Vec for both replacement and classification i.e. it is matching. A homogeneous setting would be more cost effective and computationally efficient than a heterogeneous setting because it reduces the quantity of resources needed thus saving space and also reduces information loss as representation provided at the replacement stage is retained at the classification stage.
4. **Label Preservation:** Transformations as a result of data augmentation should be semantically invariant. In text classification, only transformations that preserve the labels while augmenting data are allowed. *Thus, we need to confirm that the augmentation process preserves the original labels.*

To the best of our knowledge, currently there are no studies addressing these issues especially for hate speech detection. Hence, we are motivated to contribute to filling the above gaps in literature. Therefore, the overall aim of this work is to perform empirical studies to investigate and provide answers to these pertinent questions highlighted above related to the token substitution method of data augmentation. Specifically,

- First, we intend to design experiments to replace the Word2Vec embedding with an alternative with the potential of providing a better substitute word.
- Secondly, we propose new methods for choosing which words to substitute in a sentence.
- The third specific objective will address the issue of homogeneity or heterogeneity of pre-trained embedding.
- And finally, try to confirm if the labels were preserved by the augmentation method of choice.

The rest of this work is organized as follows: In Section 2, we summarize existing hate speech data augmentation studies. In Section 3, we discuss the methods we proposed to address the highlighted problems. Section 4 contains the experiment design which includes the datasets and the experimental baselines. The results are presented in Section 5 while Sections 6, 7 and 8 describes some further analysis, our conclusions and the intended future work respectively.

2 Related studies

In this section we present a summary of data augmentation studies in relation to hate speech detection. In general, the existing studies have followed the summary in Figure 1. This groups them into local augmentation techniques and global augmentation techniques. The local techniques focus on a specific region (words or phrases) in a sentence and alter it to generate a new sentence. The global techniques focus on the whole sentence and alter it. For example, with this sentence *"the quick brown fox jumped over the lazy dog"*, when synonym replacement (which is a local technique) is applied, it could become *"the fast white fox walked over the idle dog"* while when paraphrasing (which is a global technique) is applied, it could become *"the lazy dog was jumped over by the quick brown fox"*. Other local techniques include using a language model to predict a masked word in a sentence and randomly swapping two words, inserting a new word or randomly deleting a word in a sentence. Apart from paraphrasing, the global techniques also include translating from one language to another then translating back to the original language and generating the text either from scratch or from a seed word/phrase.

Token substitution is a popular method which replaces a token in the sentence with its synonym. This synonym is usually gotten from a dictionary or thesaurus such as WordNet in [12] or calculated using the cosine similarities between the target word and words in a pre-trained word embedding such as Word2Vec in [30, 39] or Glove in [12]. Another well-known method is generation of new text using RNN [30] or GPT [12, 18, 36, 42] or GAN [3]. In [44], they applied dependency based embeddings for word substitution to generate text while leveraging textual membership queries. Some other studies leveraged solutions such as shifting the position of words in a zero padded representation [30], synthetic minority oversampling, random over- and under- sampling, and AdaSYN [29], adding common misspelling of words to data and collecting tweets that contain swear words in conjunction with positive adjectives or racial and religious tweets [38], adding tweets with disgust and anger emotions from suspended accounts to the data [1], bootstrapping from another dataset; embedding based [13] or sentiment polarity based [8] methods.

These existing studies are done independently. In our study, we holistically look at the problem, provide comparisons while keeping the experimental designs consistent. We propose new methods of providing synonyms and novel methods for selecting the token to be substituted. Moreover, we provide an insight into label preservation property of token replacement and also discuss how pre-trained embedding homogeneity and heterogeneity affects performance.

3 Proposed methods

In this section we discuss the methods proposed to address the limitations of the existing studies and achieve the objectives.

3.1 Objective 1: source of synonyms

The investigation into the source of synonyms for replacement is based on the most widely used method which is the original pre-trained Word2Vec (W2V). To achieve this objective, we propose the use of two relevant but different pre-trained embedding methods to provide

synonyms for replacement. In general these methods improve the standard Word2Vec from different perspectives and thus we postulate that that improvement can be beneficial to the synonym replacement process.

The first method is the counter-fitted (synonym and antonym) word embedding. This is adapted from the Mrkšićs' [24] model. The Mrkšićs' model aims to improve the semantic similarity inference capability of a vector space representation of words by introducing antonymy and synonymy constraints via an external lexicon into this vector space. It achieves this with use of a three-termed objective function which includes Antonym Repel (pushes antonymous word vectors away from each other), Synonym Attract (pushes known synonymous words pairs closer) and Vector Space Preservation (preserves original semantic information by bending the transformed vector space towards the original). This generates a modified embedding space where truly synonymous words are closer and their antonyms are further apart. More details on this can be found in [24]. For the rest of this work, this method will be denoted by **CF-W2V**.

The second method is the dependency-based embedding. Here, we utilize Levy and Goldberg's [16] skip-gram model (word2vecf[2]). In this model, while learning embeddings, the linear context words are replaced with dependency context from the dependency parse graph. An illustration of this idea is culled from [16]. In the text sequence, *'australian scientist discovers star with telescope'*, the linear context words from word2vec for *'discovers'* (with a window size of 2) are *'australian'*, *'scientist'*, *'star'*, and *'with'*. The word *'telescope'* was not captured as context for *'discovers'* because of the long distance between them. However, when using word2vecf, the dependency context for the target word *'discovers'* is now *'nsubj_scientist'*, *'obj_star'*, *'obl_telescope'*. It successfully adds the related words to the context information even though they are separated by a long distance. It has been shown that dependency-based embeddings outperforms its linear counterpart for question classification and relation identification tasks [15]. For the rest of this work, this method will be denoted by **DE-W2V**.

3.2 Objective 2: substituted word selection

We propose two methods for choosing words to be substituted instead of relying on Parts-of-Speech (PoS) tags or merely selecting all the words.

The first method involves the use of a heuristic to choose words. We framed the problem of selecting replaceable words as a combinatorial optimization problem. Specifically, we use Particle Swarm Optimization (PSO) [14]. **PSO** is a population-based evolutionary computation technique where each solution can be represented as a particle in a swarm (flocks of birds or schools of fish). Each of these particles has a position (a possible solution) and a velocity (speed and direction of the particle) in the search space when initialized and they move around this space to search for the optimal solution(s). To move, each particle updates its position and its velocity based on its previous experience and the experience of the surrounding particles (population) using (1) and (2). This experience includes the particles' previous best position (*pbest*) and the previous best position of the population (*gbest*). The particle's position is evaluated based on a pre-defined fitness function.

$$v_{id}^{t+1} = w * v_{id}^{t} + c_1 * r_{1i} * (p_{id}^{t} - x_{id}^{t}) + c_2 * r_{2i} * (p_{gd}^{t} - x_{id}^{t}) \quad (1)$$

[2] https://bitbucket.org/yoavgo/word2vecf/src/default/

$$x_{id}^{t+1} = x_{id}^{t} + v_{id}^{t+1} \tag{2}$$

where v_{id}^{t} and x_{id}^{t} are the velocity and position respectively of the ith particle in dimension d at time t. w is the inertia weight representing the moving momentum of the particles. p_{id}^{t} and p_{gd}^{t} are *pbest* and *gbest* positions in dimension d at time t. c_1 and c_2 are acceleration constants, and $r1$ and $r2$ are random values uniformly distributed in [0, 1].

PSO has demonstrated its prowess for feature selection tasks [37, 43] which is a combinatorial optimization problem. We designed a binary representation for the particle positions where 0 represents an unreplaced word and 1 represents a to-be-replaced word. The search space for this task is directly proportional to the number of words in the dataset which could be approximately thousands or tens of thousands. This would cause a dimensionality problem which occupies a large memory space and requires a high computational cost. To avoid this problem, we employ Term Frequency-Inverse Document Frequency (TF-IDF) to select the top words without considering stop words. The selected words form a *bag-of-replaceable-words* and the particle size in the PSO will be the same as the size/length of this bag. Each individual/particle position will be used for all the sentences in the dataset. Hence, only words in a sentence that its position in the particle position is represented by a 1 will potentially be replaced by the replacement method of choice. When this word is selected, we also check if the synonym improves or degrades the predictive performance of the sentence. As described in Section 4.2, top N synonyms are chosen to generate more sentences. Therefore, when N synonyms for the potentially replaceable word are found, we calculate the fitness of this new sentence and only add it to the dataset if it improves the fitness over the original sentence. The fitness function is the average of all the prediction probability for the minority class. This average is what the algorithm maximizes. This prediction is from a classification model trained on the original train set and measured in terms of Macro-F1. If the prediction probability is greater than or equal to a set threshold, that individual/particle position will be selected. We empirically selected 60 particles in each swarm for 100 iterations for this experiment. For the rest of the parameters, we used the same as in [37].

The second method adapts the feature attribution method termed Integrated Gradient (IG) [34] to measure word importance. Feature attribution attempts to measure the importance of the input features based on the prediction of a deep neural network. It is one of the most popular methods that can extract insight about why a model makes a prediction. In literature, Integrated Gradient is used for explainable Artificial Intelligence (AI) [2, 35, 41] and generating adversarial samples [25]. **IG** produces a numeric score for each word in an input sentence. The magnitude of this score for each word corresponds to how important the model considers the word. For a text classification task, feature attribution can reveal which words are responsible, according to the model, for the predicted class assigned to a sentence. Attention based explanation is another popular attribution method in Natural Language Processing (NLP) for finding important words. Experiments carried out in [41] shows that attention might not explain a models prediction in some cases. Furthermore, [11] show that attention fails to provide a meaningful explanation therefore obfuscating the relationship between the attention scores and the model output. We chose IG for its ease of implementation and its superiority in literature over similar attribution methods. In [5], IG was used to calculate the feature attribution of each word in the input sentence. Then they test how fragile the classifier is by substituting words with high attribution scores (as determined by a selected threshold) with their synonyms and antonyms to check if the classifiers prediction changes. This falls under adversarial sample generation while our study falls under data augmentation.

Table 1 Homogeneous and Heterogeneous Settings

Homogeneous		Heterogeneous	
Replacement	Classification	Replacement	Classification
W2V	W2V	W2V	DE-W2V
		W2V	CF-W2V
CF-W2V	CF-W2V	CF-W2V	W2V
DE-W2V	DE-W2V	DE-W2V	W2V

In this study, we propose three settings. First, we synonym replace words considered **unimportant** (attribution score less than 0.000). In the second setting, we synonym replace words considered as **important** (attribution score greater than 0.000) and in the third setting, we **drop** words considered as unimportant from the sentence. The model under attribution is a one-layer Bi-directional Long Short Term Memory (BiLSTM) trained on the unbalanced original dataset. We empirically set the number of steps hyperparameter to 50. The augmented data for the **drop unimportant** words setting is much smaller than that of the other two settings since there is no synonym replacement occurring and hence no top N synonym selection to increase the dataset (intuition behind this is explained in Section 4.2). Hence, to increase the data size, we did a copy and paste (oversampling) to make the sizes equal with the other two settings. For the **replace important** and **replace unimportant** settings, there were sentences where the word to be replaced either does not occur in the pre-trained model or the word is considered neither important nor unimportant, thus generating the same original sentences. In cases like this, we ensure that duplicates are removed from the final dataset to avoid mimicking an oversampling effect.

These two proposed methods are different ways of solving the same substituted word selection problem. They both require a model trained on the original dataset and it might be a limitation. However, they provide a more high-level and automated method with added intelligence. The PSO method, apart from just selecting which words to be substituted, it also selects the best substitute word. This provides some sort of filtering which removes noise. The IG method interprets the model in order to identify the word importance in a sentence. Methods that have been widely used in literature for selecting the words to be substitute are a) selecting only adverbs, adjectives and nouns for replacement (AAN) and b) selecting all words in the sentence except stop words (AESW). They will serve as baselines for comparing our proposed methods.

3.3 Objective 3: Homogeneous and heterogeneous embedding

In this objective, we are trying to find out the effect on performance if the pre-trained embeddings used are kept homogeneous or heterogeneous. For the heterogeneous setting, W2V was used as the first classification layer when synonym replacement was done with CF-W2V or DE-W2V. W2V was used as replacement when DE-W2V and CF-W2V where used for classification. For the homogeneous setting, the same embedding was used for replacement and classification. Table 1 shows the different settings clearly. For this experiment, we investigate only the All Except Stop Words (AESW) method of selecting substituted word.

3.4 Objective 4: label preservation

As stated previously, the transformation for a valid data augmentation process should not change the original label of the instance. So far, for all the augmentation processes, we simply assume that any generated sentence will have the same label as its original counterpart. This assumption is not very valid because as we know, synonym replacement using W2V can replace a word with its antonym if the cosine similarities between the word and the antonym are high and there's a chance the meaning of the sentence will change and hence its label. Also in [5], they show that for difficult examples, word substitutions such as a change in name or synonym replacement changes the classifiers output, but for easier examples, they are ineffective in changing the classifier output. In order to determine if the labels were preserved, we carry out two different studies for two data augmentation processes (WordNet Synonym Replacement and MWP_BERT AESW) on the DAVIDSON dataset.

In the first study, we mimic the experiment carried out in [18]. The DAVIDSON Dataset has 3 classes with Hate class (Class 0) and Neither class (Class 2) being the minority classes. So for this analysis, we exclude Offensive class (Class 1) from the classification task as the original data for that class was not augmented. We randomly sample 100 instances from each minority class for both the original and newly generated data. Then from the original we sampled a small test set (70 for each of the two class, making it 140 in total. The size was determined by the size of the original data. We couldn't go higher because there's no more data). Then, we designed a binary classification experiment using only Class 0 and Class 2. We iteratively decrease the size of the original data and increase the size of the newly augmented data by 10%. Therefore, each time, the data size is the same, it just contains varying ratios of original and augmented data. The result is shown in Figure 3.

In the second study, we investigate how some methods preserve the labels by adapting the experiment done in [40]. The t-distributed stochastic neighbor embedding (t-SNE) of the last layers of the model trained on the combination of the original and augmented data is used to show if there is an overlap between the original and new instances. For this experiment we use only the Davidson Dataset. We visualize just the minority classes (Hate and Neutral Class). We select 1000 samples for both classes from the original and augmented data. This create a new data of size 4000 sentences. We train a new model, using the same architecture used for the rest of the experiments, on this data subset for visualization. The goal of this visualization is to show where the augmented data falls. The more it falls right on top of the original samples, the more we can claim that the labels were preserved.

4 Experiment design

4.1 Datasets

We make use of two commonly used and publicly available multi-class hate speech datasets with a severely unbalanced hate class.

1. Davidson[3]: This contains approximately 25k instances from Twitter in English with labels of Hate, Offensive, or Neither hateful nor offensive [6], more specifically, 5.77%

[3] https://github.com/t-davidson/hate-speech-and-offensive-language

Hate speech, 77.43% Offensive and 16.80% Neither. For this study, we consider the Hate (Class 0) and Neither (Class 2) classes as the minority classes while Offensive class (Class 1) is considered the majority class. Only the minority classes were augmented. We randomly split it into Train/Validation/Test sets with ratio 70:15:15.

2. Founta[4]: This contains approximately 80k instances from Twitter in English with labels Hateful, Abusive, or Normal and Spam [10], more specifically, 7.5% Hateful, 11% Abusive, 22.5% Spam and 59% None. We deleted the Spam class because it was redundant for this study. The minority classes in this study for this dataset are Class 0 (Hateful) and Class 1 (Abusive) while the majority class is Class 2 (None). Only the minority classes were augmented. We also randomly split the data into Train/Validation/Test sets with ratio 70:15:15.

The preprocessing pipeline was kept very basic as we did not want to over process and lose important information valid for data augmentation. We remove Twitter-centric noise such as usernames, urls and so on, then lowercase all the words.

4.2 Baselines and experiment settings

This study includes five baselines. The first baseline is the unaugmented, original dataset. The remaining four are re-implementation of existing data augmentation techniques in literature. The second baseline is the oversampling of the minority classes. Here, we make repeated copies of sentences in the minority class till its size becomes close to or equal to the size of the minority class. Then, there is the synonym replacement with **WordNet** and pre-trained Word2Vec (**W2V**) which are considered the third and the fourth baselines respectively. Here, words are replaced with their synonyms derived from **WordNet** and pre-trained Word2Vec (**W2V**). Finally, there's a masked word prediction using BERT (**MWP_BERT**) [7]. This is not a synonym replacement method but it is a token replacement method. Here, a word in a sentence is masked and BERT tries to predict what the masked word could be by taking the rest of the sentence into consideration. For all the synonym replacement using a large pre-trained model with the exception of BERT, a cosine similarity is used to measure the similarity or closeness of two word vectors in question. The cosine similarity between two vectors A and B of dimension n is the dot product between A and B divided by the product of the length of each vector. A larger dot product means that the vectors are closer to each other in the n-dimensional vector space and thus is a good candidate (synonym) to be used as a replacement. In order to avoid cases where vectors with a larger dimension are automatically more similar than others, the cosine distance measure scales the dot product by the length of each vector. Formally, the cosine similarity is defined as follows:

$$\cos(\mathbf{A}, \mathbf{B}) = \frac{\mathbf{AB}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} \mathbf{A}_i \mathbf{B}_i}{\sqrt{\sum_{i=1}^{n} (\mathbf{A}_i)^2} \sqrt{\sum_{i=1}^{n} (\mathbf{B}_i)^2}} \tag{3}$$

For the Davidson dataset, based on the discrepancy between the size of the classes, the Hate class (Class 0) would need to be increased 17x and the Neither class (Class 2) increased 5x to be roughly equal to the Offensive class (Class 1). At the onset of the experiments, we

[4] https://dataverse.mpi-sws.org/dataset.xhtml?persistentId=doi:10.5072/FK2/ZDTEMN

Table 2 F1 scores for baseline results on the DAVIDSON and FOUNTA Datasets

Baseline Settings	Davidson	Founta
Original Dataset	$0.646^{0.007}$	$0.570^{0.002}$
Oversampling	$\mathbf{0}.733^{0.003}$	$0.638^{0.003}$
WordNet SR	$\mathbf{0}.733^{0.002}$	$\mathbf{0}.646^{0.005}$
W2V AAN SR	$0.703^{0.007}$	$0.635^{0.005}$
W2V AESW SR	$0.707^{0.004}$	$0.635^{0.004}$
MWP_BERT	$0.724^{0.001}$	$0.602^{0.002}$

discovered that this will be almost impossible to achieve as token replacement methods will incorporate noise after certain limit. We empirically set this limit to be 5x for Hate class and 3x for Neither class. Therefore, for the oversampling baseline, we over sampled the Hate class (Class 0) 5x and the Neither class (Class 2) 3x. For the pre-trained embedding experiment, we chose top 5 synonyms for Hate class (Class 0) and top 3 for Neither class (Class 2). For MWP_BERT, in each sentence we mask non-stop words one at a time, where each mask is done on the original sentence. Therefore, the original sentence will produce the sentences equal to the number of non-stop words in the sentence. We experimented with another masked word prediction with BERT setting, where we iteratively mask the next word in the newly predicted sentence, therefore, one original sentence will produce exactly one augmented sentence. But then we discovered that the resulting sentence almost always had a completely different meaning from the original. We refrain from using this method. On the other hand, for the FOUNTA data, the Hate class (Class 0) would need to be increased 17x and the Abusive class (Class 1) increased 7x to be roughly equal to the None class (Class 2). We empirically set this limit to be 6x for Hate class (Class 0) for and 3x for Abusive class (Class 1). It is obvious that this would not balance the classes but hopefully increases it sufficiently enough to decrease overfitting and improve the generalization ability of the trained models.

The classification algorithm used in the classification task is a one layer Bi-directional LSTM. The network is initialized with the embedding layer, followed by the bidirectional layer and one dense layer. The loss was calculated with sparse categorical crossentropy, and the optimizer was 'Adagrad'. Early stopping and model checkpointing were used to avoid overfitting. We chose a very simple model in order to limit its effect on the outcome and thus gives a better measure of the augmentation methods. The test set was kept constant and unseen for all experiments.

5 Results

In this section, the classification results on the unseen test sets of the two datasets are presented using Macro F1 as a performance metric. The reported scores are an average of 10 runs with the standard deviation in superscript.

5.1 Baseline results

First, the classification results for the baselines are discussed. From Table 2, it is shown that all the baselines clearly outperform the original unaugmented data reiterating the

Table 3 F1 scores for source of synonyms on the Davidson and Founta Datasets

Source of Synonyms Settings	Davidson	Founta
Original Dataset	$0.646^{0.007}$	$0.570^{0.002}$
W2V AAN SR	$0.703^{0.007}$	$\mathbf{0}.635^{0.005}$
W2V AESW SR	$0.707^{0.004}$	$\mathbf{0}.635^{0.004}$
CF-W2V AAN SR	$\mathbf{0}.728^{0.002}$	$0.633^{0.002}$
CF-W2V AESW SR	$0.712^{0.002}$	$0.623^{0.004}$
DE-W2V AAN SR	$0.609^{0.004}$	$0.633^{0.005}$
DE-W2V AESW SR	$0.711^{0.003}$	$0.626^{0.003}$

advantages of data augmentation for supervised tasks. Oversampling and WordNet Synonym Replacement (SR) methods are the top methods clearly outperforming the W2V based methods for both Davidson and Founta. MWP_BERT when applied to the Davidson dataset strongly outperformed the original dataset but when applied on the Founta dataset is the least performing baseline. This could be because there are less noisy sentence structures in Davidson than in Founta and hence when a word is masked the rest of the sentence still contains enough information to predict a correct word. Also, we rarely expect results to be consistent for the two datasets. This is because in the Davidson data, our minority classes are the Hate and Neutral class while in the Founta, the minority classes are the Hate and Abusive classes. Thus, the Founta data might be a bit harder to improve.

5.2 Results on source of synonyms

Next in Table 3, we compare the W2V source of synonym and the Original dataset with the proposed methods for Objective 1 which are the CF-W2V and the DE-W2V sources of synonym. We present the result of each source under two settings: the AAN and the AESW selection methods. From this, we observe that for the Davidson dataset, the CF-W2V AAN improves the classification performance over the rest. The counter-fitted vectors outperforms the W2V counterparts. The DE-W2V AESW outperforms its W2V counterpart. For the Founta dataset, there is little significant difference between the sources of synonyms.

Furthermore, in order to better understand the performance of each word embedding used as a source of synonyms, we observed the most similar word vectors for randomly selected words from the dataset, which are shown in Table 4. As shown, the employed pre-trained embeddings produced different synonyms for the same words. Generally, W2V is noisier than the rest. CF-W2V and DE-W2V seem to produce more meaningful synonyms. It is important to note that there is large vocabulary size discrepancy among the pre-trained embedding source. The vocabulary sizes of W2V, CF-W2V, DE-W2V are 400000, 78763 and 174015 respectively. Additionally, W2V was used as the embedding layer in the classification network. The vocabulary size and the use of W2V for embedding would consequently affect the performance. To illustrate, if using DE-W2V to provide the synonym *"preposterous"* to replace *"idiotic"*, but in the W2V embedding layer, there is no representation for *"preposterous"* then that piece of information is lost and not presented to the neural network.

Table 4 Common words and their top 4 synonyms (as determined by cosine similarity) in different sources

Words	W2V	CF-W2V	DE-W2V
horror	Horror	gruesome	suspense
	FEARnet_branded_VOD	scary	sexploitation
	horror_flick	terrifying	fiction
	horror_flicks	horrific	science-fiction
redneck	hillbilly	hick	beatnik
	rednecks	kickin	yuppie
	hick	fuckin	skinhead
	hayseed	smelly	hipster
idiotic	stupid	foolish	preposterous
	moronic	silly	mean-spirited
	asinine	nonsensical	deceitful
	inane	unwise	trollish
fuck	fucking	fucking	bugger
	f_*_ck	fucked	fuck
	f_**_k	fuckin	shit
	shit	bitches	shit

Table 5 F1 scores for substituted word selection methods on the Davidson and Founta Datasets

Substituted Word Selection Settings	Davidson	Founta
AAN	$0.703^{0.007}$	$0.635^{0.005}$
AESW	$0.707^{0.004}$	$0.635^{0.004}$
PSO	$\mathbf{0}.723^{0.001}$	$0.639^{0.005}$
IG Replace Unimportant	$0.702^{0.004}$	$0.638^{0.004}$
IG Replace Important	$0.705^{0.004}$	$\mathbf{0}.641^{0.006}$
IG Drop Unimportant	$0.696^{0.006}$	$0.637^{0.004}$

(AAN: Adverb Adjective Noun, AESW: All Except Stop Words, PSO: Particle Swarm Optimization, IG: Integrated Gradient)

5.3 Results on substituted word selection

To show the results for Objective 2, we present the test scores of models trained on data derived from different word selection methods: AAN, AESW, PSO, and IG in Table 5. For these methods, W2V was used as the source of synonym. The results show that our proposed methods of selecting words to be substituted outperformed the AAN and AESW baselines in some settings for both datasets. For the Davidson dataset, PSO method successfully selected words for substitution that outperformed the other selection methods. For the IG method, synonym replacing important and unimportant performed better than dropping unimportant words. This could be because even though words are considered unimportant to the model, they are still not irrelevant to the classification task. This is supported by the fact that IG Drop Unimportant slightly performed lower than the baselines. For the Founta dataset, IG Replace Important words outperforms the rest. Analogous to the Davidson dataset, the IG Drop Unimportant setting had the least improvement in performance.

Analysing the results further, for the DAVIDSON data, the PSO selected an average of approximately 51% of the available words for replacement, while for the FOUNTA data, it selected an average of approximately 50% of the available words. This further demonstrates that PSO successfully selected words from the bag-of-replaceable words whose replacement and the quality of the subsequently chosen synonym will improve performance. In Figure 2, we depict, using random sentences culled from the DAVIDSON dataset, the way IG method attribute importance to words. The sentences are *"that band is white trash and only white trash would buy that album"*, *"we agree do fuck yes i do send those illegal wetback home"*, and *"they shot another monkey"*. When the same word occurs multiple times in the same sentence, the word will have the same attribution score. Also, words that a human might deem important could be considered unimportant by the model. For example, the word *"illegal"* in *"we agree do fuck yes i do send those illegal wetback home"*.

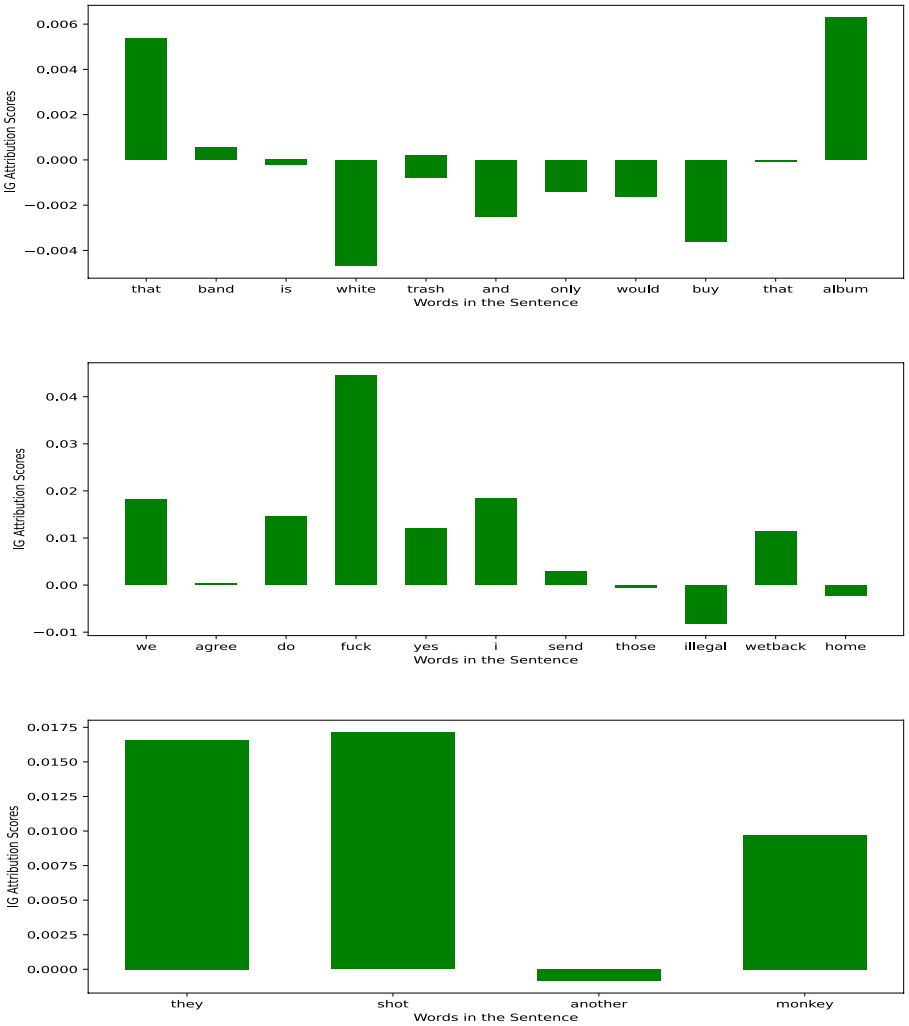

Fig. 2 Word Attribution/Importance via Integrated Gradient method

Table 6 Homogeneous and Heterogeneous Results

Homogeneous			Heterogeneous		
Repl	Clas	F1	Repl	Clas	F1
W2V	W2V	$\mathbf{0}.707^{0.004}$	W2V	DE-W2V	$0.701^{0.002}$
			W2V	CF-W2V	$0.700^{0.003}$
CF-W2V	CF-W2V	$\mathbf{0}.715^{0.002}$	CF-W2V	W2V	$0.712^{0.002}$
DE-W2V	DE-W2V	$\mathbf{0}.716^{0.002}$	DE-W2V	W2V	$0.711^{0.003}$

(**Repl**: The embedding used to provide synonym for replacement. **Clas**: The embedding used as the first layer in the neural network)

5.4 Results on Homogeneity vs. Heterogeneity

For Objective 3, the results are shown in terms of F1 scores for the test set for models trained on data generated via

- W2V synonym replacement with W2V embedding.
- CF-W2V synonym replacement with CF-W2V embedding.
- DE-W2V synonym replacement with DE-W2V embedding.
- W2V synonym replacement with DE-W2V embedding.
- W2V synonym replacement with CF-W2V embedding.
- CF-W2V synonym replacement with W2V embedding.
- DE-W2V synonym replacement with W2V embedding.

This experiment was done only on the AESW setting using the Davidson dataset. These results are shown in Table 6. The homogeneous setting outperformed its heterogeneous counterparts across board. The difference however is not significant. As pointed out in Section 5.2, W2V is 5x larger than CF-W2V and 2x larger than DE-W2V. Despite the disparate vocabulary sizes of these embeddings, the smaller embeddings (CF-W2V and DE-W2V) improved the performance. This further points to the strength of these smaller embedding and the improved quality of information they captured. For the homogeneous setting, all the synonyms that were inserted during augmentation are guaranteed to be represented in the neural network, hence that information is not lost. We also noticed that the CF-W2V and DE-W2V both outperformed the W2V counterpart when they were used as replacement in both the homogeneous and heterogeneous settings.

5.5 Results on label preservation

For objective 4, the results of the first experiment are presented in Figure 3. We plot on the y-axis the Macro-F1 scores of the models on the test set and on the x-axis we represent the different slices of the original and newly generated train data used for training the models. From the plot, we can see that the difference between the 90/10 and 10/90 setting for MWP_BERT is much lower than the difference for WordNet Synonym Replacement. Thus for MWP_BERT, the performance is relatively stable when the percentage of the new data is more than original data compared to when the percentage of original data is more than new data. On the other hand, for WordNet the difference in performance is

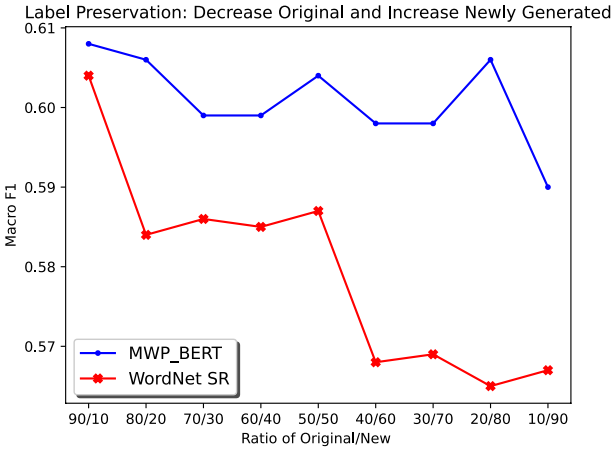

Fig. 3 Label Preservation on Davidson Data

much higher. This shows that the former, preserved the labels better. This is expected as BERT takes the whole context into consideration thereby retaining the original meaning of the sentence. While in WordNet, replacing individual words might not retain the same contextual information or meaning as the original sentence. This experiment can also be used to measure the quality of generated data. If the performance drops as more augmented data is added, then it could point to the fact that the quality of the data is not very high. This is the case for the results shown. There is a slight degradation in the quality of the data produced.

The results of the second experiment are shown as t-SNE figures in Figures 4 and 5 for Davidson Data. They are best viewed in color. The t-SNE of the last layer of the model trained on the combination of the original and MWP_BERT augmented data is shown in Figure 4. The figure shows that to a great extent, the original and augmented data fall in a similar spot (overlapping each other). There are some rouge instances however. In these cases, it seems to be more of instances from the two classes that are hard to distinguish. Consequently, their augmented counterparts are also hard to distinguish. For the WordNet augmentation method, the t-SNE in Figure 5 shows that there are only a few instances where the original and the augmented did not overlap. This occurs more in the neutral class. However, overall, there is a good enough preservation of labels for both methods with the MWP_BERT performing slightly better.

6 Further analysis

Now, let us look at the quality of sentences produced by the different augmentation methods that generated a new sentence (i.e all except Oversampling) for a particular given sentence. The example original sentences were culled from the Davidson dataset. Table 7 shows that generally the methods do not generate a sentence completely humanly indistinguishable from the original sentence. There are indications that some of the transformation could change the meaning and thus the label of the sentence. For instance this sentence *"they shot another monkey"*, which is labelled as hate speech in the dataset was transformed to *"they shot another hamster"*. The first sentence given the right context (dark skinned people of African descent have derogatorily been referred to with the word

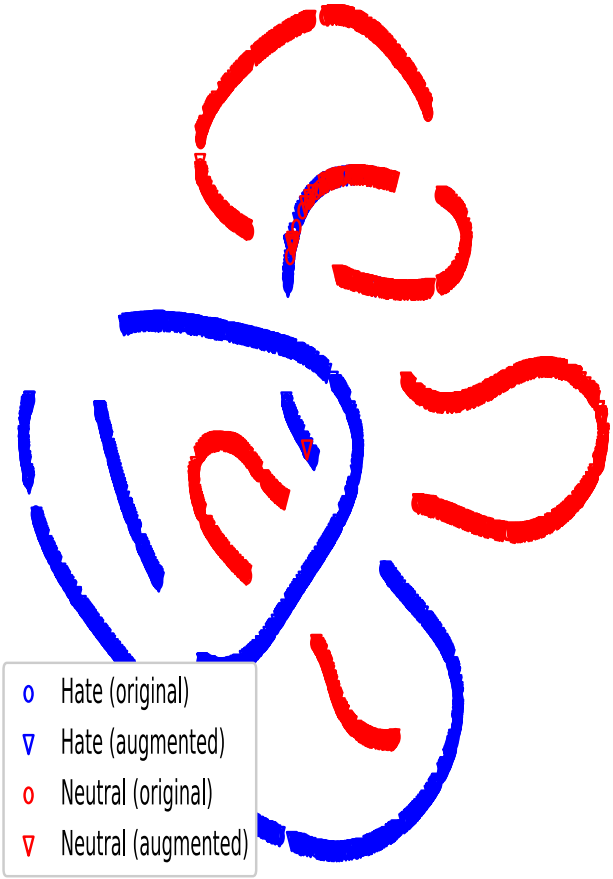

Fig. 4 MWP_BERT t-SNE

"monkey") is hateful but its transformation would not be considered hateful. Further, in the sentence *"that band is white trash and only white trash would buy that album"* the hateful phrase is *"white trash"*. However, the system interprets words individually hence *"white"* is been replaced with the names of other colors which do not carry the same meaning.

7 Conclusions

This study presented an in-depth investigation into the token replacement methods for increasing the minority classes in hate speech detection datasets. We have proposed the use of novel methods for providing synonyms and also for selecting the candidate word to be replaced in a sentence. Our results show that the proposed methods provide varying synonyms that are better than or comparable to existing synonym sources. Moreover, our results also show that the proposed methods of selecting candidate words are superior to the baseline methods. Additionally, we have shown the effect of heterogeneity and homogeneity when carrying out synonym replacement for classification tasks and also probed the label preservation strength

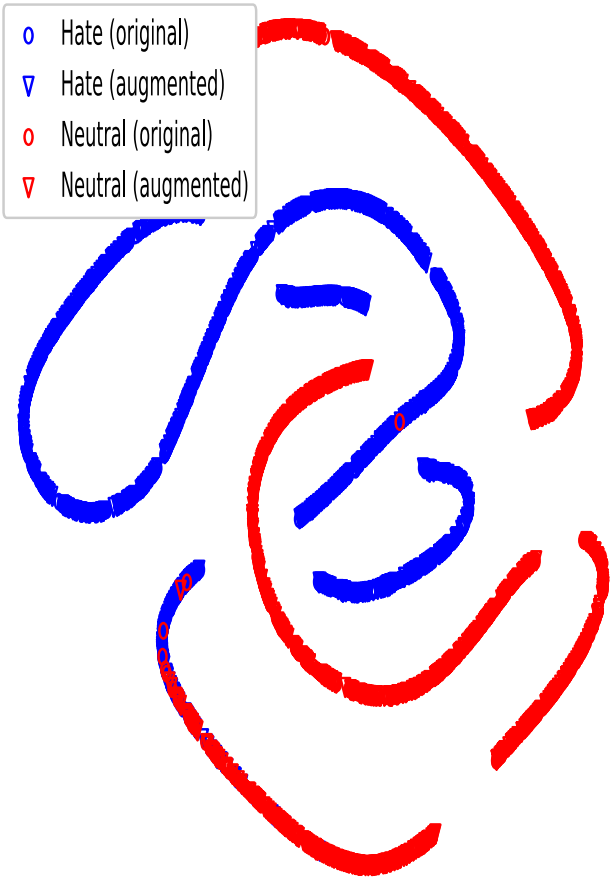

Fig. 5 WordNet t-SNE

of these methods. Our results showed that the homogeneous setting better served the task and also showed how some of the methods preserved the original labels.

We have uncovered some limitations that plagues the token replacement methods. First, because it is primarily a local word replacement method, it limits the diversity of the generated sentence since only one word is changed. Also, when a word has more than one meaning, it could have different synonyms. The vocabulary of pre-trained word embeddings are limited and hence some of the candidate words to be replaced do not occur in the embedding therefore that it will have no replacements thus infringing on the diversity.

8 Future work

Building on the findings of this study, we would like to further explore methods of conducting synonym replacement for homonyms. Hence the meaning/context of the word will assist in determining which synonym is more appropriate. Additionally, we intend to explore methods of solving the limited embedding vocabulary problems, thereby improving the chances that a candidate word and its synonym existing in the vocabulary.

Table 7 Different Generated Sentences for different augmentation methods

Settings	Sentence
Original	**that band is white trash and only white trash would buy that album**
WordNet	that dance band is white trash and only white trash would buy that album
W2V_AAN	that bands is black garbage and one black garbage would buy that album
W2V_AESW	that bands is black garbage and only black garbage could sell that album
MWP_BERT	that band is white trash and only white trash can buy that album
CF-W2V_AAN	that band is brown rubbish and solely brown rubbish would buy that album
CF-W2V_AESW	that banding is brown rubbish and only brown rubbish could hold that album
DE-W2V_AAN	that boyband is black garbage and measly black garbgae would buy that mini-album
DE-W2V_AESW	that boyband is black garbage and only black garbage whould sell that album.
PSO Method	that band is white trash and only white trash would buy that album
IG_Unimp	that band is black trash and only white trash would buy that album
IG_Imp	it bands is white trash and only white trash would buy that album
IG_DropUnimp	that band trash album
Original	**we agree do fuck yes i do! send those illegal wetback home**
WordNet	we agree do fuck yes i do send those illegal wetback national
W2V_AAN	we agree do fuckiveng yes ive do! send those ivellegal wetback house
W2V_AESW	we agrees do f_**_k mso_style_noshow i do! sends those illicit spics homes
MWP_BERT	we agree do fuck yes i do! send those illegal aliens home
CF-W2V_AAN	we agree do fucked yes i do! send those illegal wetback homing
CF-W2V_AESW	we unhappy do fucking yeah i do! sent those illicit pinche habitation
DE-W2V_AAN	we agree do fuck yes i do! send those illegal wetback residence
DE-W2V_AESW	we concur do bugger yeah i do! sent those unlawful wetback homes
PSO	we agree do fuck yes i do send those illegal wetback home
IG_Unimp	we agree do fuck yes i do send those illicit wetback homes
IG_Imp	we disagree want fucking mso_style_qformat i do send those illegal wetback home
IG_DropUnimp	we agree do fuck yes i do send wetback
Original	**they shot another monkey**
WordNet	they shot some other monkey
W2V_AAN	they shot another monkeys
W2V_AESW	they shots another monkeys
MWP_BERT	they killed another monkey
CF-W2V_AAN	they shot another curious
CF-W2V_AESW	they pulls deliriously curious
DE-W2V_AAN	they shot another hamster
DE-W2V_AESW	they stabbed a hamster
PSO	they shot another monkey
IG_Unimp	they shot another monkey
IG_Imp	we shots another monkeys
IG_DropUnimp	they shot monkey

Funding Open Access funding enabled and organized by CAUL and its Member Institutions.

Data availability Not applicable

Code availability Not applicable

Compliance with ethical standards

Conflicts of interest The authors declare that they have no conflict of interest.

References

1. Alorainy, W., Burnap, P., Liu, H., Javed, A., Williams, M.L.: Suspended accounts: A source of tweets with disgust and anger emotions for augmenting hate speech data sample. In: 2018 International Conference on Machine Learning and Cybernetics (ICMLC), vol. 2, pp. 581–586 (2018). https://doi.org/10.1109/ICMLC.2018.8527001
2. Bodria, F., Panisson, A., Perotti, A., Piaggesi, S.: Explainability methods for natural language processing: applications to sentiment analysis. In: SEBD (2020)
3. Cao, R., Lee, R.K.W.: HateGAN: Adversarial generative-based data augmentation for hate speech detection. In: Proceedings of the 28th International Conference on Computational Linguistics, pp. 6327–6338. International Committee on Computational Linguistics, Barcelona, Spain (Online) (2020). https://doi.org/10.18653/v1/2020.coling-main.557
4. Coulombe, C.: Text data augmentation made simple by leveraging NLP cloud apis (2018). arxiv:1812.04718
5. Datta, D., Kumar, S., Barnes, L., Fletcher, T.: Geometry matters: Exploring language examples at the decision boundary (2020). arxiv:2010.07212
6. Davidson, T., Warmsley, D., Macy, M., Weber, I.: 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 (2017)
7. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: 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, pp. 4171–4186. Association for Computational Linguistics, Minneapolis, Minnesota (2019). https://doi.org/10.18653/v1/N19-1423
8. Elmadany, A., Zhang, C., Abdul-Mageed, M., Hashemi, A.: Leveraging affective bidirectional transformers for offensive language detection. In: Proceedings of the 4th Workshop on Open-source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection, pp. 102–108. European Language Resource Association, Marseille, France (2020). https://www.aclweb.org/anthology/2020.osact-1.17
9. Fadaee, M., Bisazza, A., Monz, C.: Data augmentation for low-resource neural machine translation. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pp. 567–573. Association for Computational Linguistics, Vancouver, Canada (2017). https://doi.org/10.18653/v1/P17-2090
10. Founta, A.M., Djouvas, C., Chatzakou, D., Leontiadis, I., Blackburn, J., Stringhini, G., Vakali, A., Sirivianos, M., Kourtellis, N.: Large scale crowdsourcing and characterization of twitter abusive behavior. In: Proceedings of the Twelfth International AAAI Conference on Web and Social Media (ICWSM 2018) (2018). arxiv:1802.00393

11. Jain, S., Wallace, B.C.: Attention is not Explanation. 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), pp. 3543–3556. Association for Computational Linguistics, Minneapolis, Minnesota (2019). https://doi.org/10.18653/v1/N19-1357
12. Juuti, M., Gröndahl, T., Flanagan, A., Asokan, N.: A little goes a long way: Improving toxic language classification despite data scarcity. In: Findings of the Association for Computational Linguistics: EMNLP (2020)
13. Kebriaei, E., Karimi, S., Sabri, N., Shakery, A.: Emad at SemEval-2019 task 6: Offensive language identification using traditional machine learning and deep learning approaches. In: Proceedings of the 13th International Workshop on Semantic Evaluation, pp. 600–603. Association for Computational Linguistics, Minneapolis, Minnesota, USA (2019). https://doi.org/10.18653/v1/S19-2107
14. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: Proceedings of ICNN'95 - International Conference on Neural Networks, vol. 4, pp. 1942–1948 vol.4 (1995). https://doi.org/10.1109/ICNN.1995.488968
15. Komninos, A., Manandhar, S.: Dependency based embeddings for sentence classification tasks. In: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 1490–1500. Association for Computational Linguistics, San Diego, California (2016). https://www.aclweb.org/anthology/N16-1175
16. Levy, O., Goldberg, Y.: Dependency-based word embeddings. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pp. 302–308. Association for Computational Linguistics, Baltimore, Maryland (2014). https://www.aclweb.org/anthology/P14-2050
17. Li, Y., Cohn, T., Baldwin, T.: Robust training under linguistic adversity. In: Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers, pp. 21–27. Association for Computational Linguistics, Valencia, Spain (2017). https://www.aclweb.org/anthology/E17-2004
18. Liu, R., Xu, G., Vosoughi, S.: Enhanced offensive language detection through data augmentation (2020)
19. Madukwe, K., Gao, X., Xue, B.: In data we trust: A critical analysis of hate speech detection datasets. In: Proceedings of the Fourth Workshop on Online Abuse and Harms, pp. 150–161. Association for Computational Linguistics, Online (2020). https://doi.org/10.18653/v1/2020.alw-1.18
20. Madukwe, K.J., Gao, X.: The thin line between hate and profanity. In: Liu, J., Bailey, J. (eds.) AI 2019: Advances in Artificial Intelligence, pp. 344–356. Springer International Publishing, Cham (2019)
21. Madukwe, K.J., Gao, X., Xue, B.: A ga-based approach to fine-tuning bert for hate speech detection. In: 2020 IEEE Symposium Series on Computational Intelligence (SSCI), pp. 2821–2828 (2020). https://doi.org/10.1109/SSCI47803.2020.9308419
22. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space (2013)
23. Mozafari, M., Farahbakhsh, R., Crespi, N.: A bert-based transfer learning approach for hate speech detection in online social media (2019)
24. Mrkšić, N., Ó Séaghdha, D., Thomson, B., Gašić, M., Rojas-Barahona, L.M., Su, P.H., Vandyke, D., Wen, T.H., Young, S.: Counter-fitting word vectors to linguistic constraints. In: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 142–148. Association for Computational Linguistics, San Diego, California (2016). https://www.aclweb.org/anthology/N16-1018
25. Mudrakarta, P.K., Taly, A., Sundararajan, M., Dhamdhere, K.: Did the model understand the question? In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 1896–1906. Association for Computational Linguistics, Melbourne, Australia (2018). https://doi.org/10.18653/v1/P18-1176
26. Pennington, J., Socher, R., Manning, C.: GloVe: Global vectors for word representation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1532–1543. Association for Computational Linguistics, Doha, Qatar (2014). https://doi.org/10.3115/v1/D14-1162
27. Pitsilis, G., Ramampiaro, H., Langseth, H.: Effective hate-speech detection in twitter data using recurrent neural networks. Applied Intelligence **48**, in press. (2018). https://doi.org/10.1007/s10489-018-1242-y
28. Ramirez, J.M., Montalvo, A., Calvo, J.R.: A survey of the effects of data augmentation for automatic speech recognition systems. In: Nyström, I., Hernández Heredia, Y., Milián Núñez V. (eds.)

Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications, pp. 669–678. Springer International Publishing, Cham (2019)
29. Rathpisey, H., Adji, T.B.: Handling imbalance issue in hate speech classification using sampling-based methods. In: 2019 5th International Conference on Science in Information Technology (ICSITech), pp. 193–198 (2019). https://doi.org/10.1109/ICSITech46713.2019.8987500
30. Rizos, G., Hemker, K., Schuller, B.: Augment to prevent: short-text data augmentation in deep learning for hate-speech classification, p. 991-1000. Association for Computing Machinery, New York, NY, USA (2019). https://doi.org/10.1145/3357384.3358040
31. Şahin, G.G., Steedman, M.: Data augmentation via dependency tree morphing for low-resource languages. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pp. 5004–5009. Association for Computational Linguistics, Brussels, Belgium (2018). https://doi.org/10.18653/v1/D18-1545
32. Shorten, C., Khoshgoftaar, T.: A survey on image data augmentation for deep learning. Journal of Big Data **6**, 1–48 (2019)
33. Simard, P.Y., LeCun, Y.A., Denker, J.S., Victorri, B.: Transformation Invariance in Pattern Recognition — Tangent Distance and Tangent Propagation, pp. 239–274. Springer, Berlin (1998). https://doi.org/10.1007/3-540-49430-8_13
34. Sundararajan, M., Taly, A., Yan, Q.: Axiomatic attribution for deep networks. In: Proceedings of the 34th International Conference on Machine Learning - Volume 70, ICML'17, p. 3319-3328. JMLR.org (2017)
35. Tang, X., Shen, X., Wang, Y., Yang, Y.: Categorizing offensive language in social networks: A chinese corpus, systems and an explanation tool. In: CNCL (2020)
36. Tekiroglu, S.S., Chung, Y.L., Guerini, M.: Generating counter narratives against online hate speech: Data and strategies (2020)
37. Tran, B., Xue, B., Zhang, M.: Variable-length particle swarm optimization for feature selection on high-dimensional classification. IEEE Transactions on Evolutionary Computation **23**(3), 473–487 (2019). https://doi.org/10.1109/TEVC.2018.2869405
38. Vijayaraghavan, P., Larochelle, H., Roy, D.: Interpretable multi-modal hate speech detection (2021)
39. Wang, W.Y., Yang, D.: That's so annoying!!!: A lexical and frame-semantic embedding based data augmentation approach to automatic categorization of annoying behaviors using #petpeeve tweets. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pp. 2557–2563. Association for Computational Linguistics, Lisbon, Portugal (2015). https://doi.org/10.18653/v1/D15-1306
40. Wei, J., Zou, K.: Eda: Easy data augmentation techniques for boosting performance on text classification tasks (2019)
41. Wright, A.P., Shaikh, O., Park, H., Epperson, W., Ahmed, M., Pinel, S., Chau, D.H., Yang, D.: RECAST: enabling user recourse and interpretability of toxicity detection models with interactive visualization (2021) arxiv:2102.04427
42. Wullach, T., Adler, A., Minkov, E.: Towards hate speech detection at large via deep generative modeling. IEEE Internet Computing pp. 1–1 (2020). https://doi.org/10.1109/MIC.2020.3033161
43. Xue, B., Zhang, M., Browne, W.N.: Particle swarm optimisation for feature selection in classification: Novel initialisation and updating mechanisms. Applied Soft Computing **18**, 261–276 (2014). https://doi.org/10.1016/j.asoc.2013.09.018
44. Zarecki, J., Markovitch, S.: Textual membership queries (2018). arxiv:1805.04609
45. Zhang, X., Zhao, J., LeCun, Y.: Character-level convolutional networks for text classification. In: Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1, NIPS'15, p. 649-657. MIT Press, Cambridge, MA, USA (2015)