Exploring N-gram, Word Embedding and Topic Models for Content-based Fake News Detection in FakeNewsNet Evaluation

Oluwafemi Oriola
Department of Computer Science
Adekunle Ajasin University
Akungba Akoko, Nigeria

ABSTRACT
FakeNewsNet is a repository of two novel datasets, PolitiFact and GossipCop, which are employed for evaluation of fake news detection techniques. Unlike other extensively studied benchmark fake news datasets, the FakeNewsNet datasets incorporate news content, social context, and dynamic information, which could be used to study fake news propagation, detection, and mitigation. Existing works on FakeNewsNet have focused on one-hot encoding, social contexts such as user-based models, and dynamic information such as news propagation model. However, n-gram, word embeddings, and topic models of news contents, which have been impressive in other contexts have not been explored. This paper therefore explores n-gram, word embeddings, and topic models of news contents for the evaluation of FakeNewsNet datasets. Unigram-based n-gram model, skip-gram word2vec-based word embeddings model and Latent Dirichlet Allocation-based topic model are extracted after preprocessing the datasets. The features are weighted by TFIDF to overcome the shortcomings of the individual models and analyzed using Logistic Regression. The evaluation of the models and their hybrids shows that n-gram model outperforms word embedding and topic models. Specifically, n-gram model records accuracy, precision, recall and F1-score of 0.80, 0.79, 0.78 and 0.79, respectively for PolitiFact and records 0.82, 0.75, 0.79 and 0.77, respectively for GossipCop. The comparison with benchmarks also shows that the performance of n-gram model is better.

General Terms
Machine Learning, Computational Linguistics

Keywords
Fake News Detection, FakeNewsNet, Classification, News Content Features, TFIDF

1. INTRODUCTION
The social media in the 21st century has been the game changing phenomenon within communication community[1]. It has bridged the gap of communication among people of different races, backgrounds, and social orientations. The media serves as an interactive means for disseminating and sharing information on the internet; hence, individuals and cooperate organizations have continued to rely on the information on the media in taken decisions.

However, the social media in recent times has been ravaged by falsehood, lies, rumour and distorted information, which has continued to rise uncontrollably in recent times[2].This is because of inadequate measure to check the truthfulness of fact and mitigate the propagation of fake information in social networks. Arising from this, many research efforts have gone into combating fake news for safe social media.

Big data and machine learning advances have been critical among the tools and techniques used by researchers. Notable publicly available datasets that have been used for research purposes include LIAR [3] , FNC-1[4] and BuzzFeed News[5], which have focused on news content features and few social engagement features.

Among the machine learning techniques, supervised machine learning techniques have been extensively used with impressive performances. For instance, logistic regression and support vector machine performed outstandingly in [6] [7] with surface level features including topic model, while deep learning models performed impressively with surface level features and word embeddings in [3] [8].

However, none of the existing datasets is as robust as FakeNewsNet which allows research into news content, social interaction, and news propagation. In the evaluation in [9], different models of the news contents, social contexts, and dynamic information have been explored. Nonetheless, news content models such as n-gram, word embedding and topic model, which have been impressive in previous datasets have not be evaluated. Therefore, this paper explores the models in comparison to previous works.

2. FAKENEWSNET
FakeNewsNet [9] was motivated by social engagement and behaviour aspect of consumers on social media as well as how to capture dynamic information related to fake news propagation, users' reactions to fake news and temporal patterns for early fake news detection and intervention. FakeNewsNet is a public multidimension data repository, which contains two datasets with news content, social context, and dynamic information. According to the developer, the sets of features provide opportunity for exploratory study of different approaches for better understanding of disinformation tactics.

The feature segments of the repository include news content, social context, and dynamic information.

The news contents are the texts of the tweets, which are labelled as fake news or true news using two fact-checking websites: PolitiFact and GossipCop. PolitiFact [10] is a website operated by Tampa Bay Times, where reporters and editors from the media fact-check the political news articles. It publishes the original statement of news articles and their complete fact-check evaluation results in their website. GossipCop [11] is a website for fact-checking entertainment stories. GossipCop analyses the information and provides

truth value for each story ranging from 0 to 10. The word clouds for the news content compositions of the datasets is presented in Figure 1.

Fig. 1. Word clouds for the news contents in FakeNewsNet

The social context consists post, second order user behaviour such as replies, re-posts and likes, which are collected using Twitter's Advanced Search API as well as the meta information for user profiles, user posts, and the social network information. The dynamic context includes the information that are used to track the updates such as timestamps of user engagements and topics of fake news.

3. STATE-OF-THE-ARTS IN FAKE NEWS DETECTION

Several models have been explored for detection of fake news using different evaluation datasets. The state-of-the-arts are presented as follows:

Wang [3] presented LIAR, a dataset for fake news detection. The 12.8K dataset collected from politifact.com[12] was manually labelled as pants-fire, false, barely-true, half-true, mostly true, and true based on human annotation. By applying surface linguistic features and speaker related data, it was found that support vector machine (SVM) and logistic regression (LogReg) with surface linguistic features outperformed convolutional neural networks(CNN), bidirectional long short-term memory recurrent neural network (BiLSTM) and hybrid CNN for cross validation while hybrid CNN model with combination of all features performed best for testing. Both LIAR and Buzzfeed News[5] were employed to evaluate user engagement features in [13]. The results showed that the unsupervised fake news detection algorithm was promising.

Thota *et al.*[8] evaluated fake news challenge dataset (FNC-1)[4], a stance-based fake news detection dataset using n-gram, bag of word and word2vec model weighted by TFIDF. By applying CNN, n-gram with unigram and bigram performed best with accuracy of 0.94 compared to bag of word of 0.89 and word2vec of 0.75 for binary classification. The FNC1 dataset was also applied in [14] to detect the stance of the title with respect to its article. By combining n-grams, word embeddings and cue words, macro F1-score of 0.596 was achieved with CNN.

Xu *et al.* [15] explored domain reputation features such as website registration behaviour, internet site ages, domain properties, probability of news disappearance and news content features such as TFIDF, LDA topic, Jaccard document similarity models for identifying fake and real news in BuzzFeed News[5]. They found that the fake and real news exhibited substantial differences based on reputations and domain features, while little differences were shown by TFIDF and topic models.

Topic agnostic and web markup models were fused in [7] to identify fake news in PoliticalNews corpus, which is not publicly available. The classification with SVM, KNN and RF showed that SVM recorded the highest accuracy of 0.83. The method was also impressive in other datasets with accuracy above 0.6. Aldwairi and Alwahedi[6] constructed a dataset, which is not publicly available but the methods can be utilized by users to detect and filter out sites containing false and misleading information. Based on combination of character and lexical count, LogReg was impressive in detecting fake news with accuracy, F1 and ROC of 99.4, 99.3 and 99.5, respectively compared to BayesNet, Random Tree and Naïve Bayes.

The FakeNewsNet evalution datasets, namely PolitiFact and GossipCop were evaluated based on one-hot encoding for news content, counts of social context features and dynamic information by the developer[9]. SVM, LogReg, NaiveBayes, CNN and LSTM were used as classifiers for the datasets which were divided into 80/20 train/test percentage split. By social article fusion model, LSTM performed best for PolitiFact with accuracy of 0.691 and F1-score of 0.706. LogReg with one hot encoding of news contents performed best for GossipCop with accuracy of 0.822 and F1-score of 0.799. Apart from this legacy works, no other work has focused on improvement of the benchmarks, which is the focus of this paper.

4. METHODS

4.1 Models

In this paper, news content features are explored to model n-gram, word embedding and topic models as the base models and their hybrids. The models are presented as follows.

4.1.1 Base Models

The base modes include:

1. N-gram model combines sequential words into lists with size *n* to enumerate all the expressions of size *n* and count all occurrences. It is better than simple bag of words that relies on the frequency of words[16]. Specifically, unigram word n-gram also known as 1-gram is used and implemented in Python Scikit-Learn

0.22 Library[17]. The unigram features are weighted by term frequency inverse document frequency (TFIDF) due to its better performance in previous works[18][19].

The TFIDF for a given term t in a document d is given as:

$$\text{TF-IDF}(t, d) = \text{TF}(t, d) \times \text{IDF}(t) \quad (1)$$

when $\text{IDF}(t) = \log [\, n / (\text{DF}(t) + 1]$

n = total number of documents in the dataset

DF(t) = document frequency of t

Therefore,

The weighted n-gram is:

$$W(t,d) = \text{N-gram}(t,d) \times \text{TF-IDF}(t,d) \quad (2)$$

2. Word embedding is distributed representation of words in a vector space. Often, it is constructed using neural networks. The model is built on word and its contexts. In this study, skip-gram word2vec[20] is made use of to create a 300-dimensional dense vector space with word embeddings as features. The model is implemented using Gensim 0.8.6 [21] with minimum word count of 1, context window size of 10, workers of 8 and word vector dimensionality of 300 features.

3. Topic model is used to discover the abstract topic that occurs in each document. Latent Dirichlet Allocation (LDA) topic modelling technique[22] is used to extract important topics from the news contents. It represents documents in a corpus as a random mixture of latent topics, which are characterized by a probability distribution over vocabulary of words or terms extracted from all documents in the corpus. Topics between the range of 2 to 6 are tested in Python Scikit-Learn 0.22 Library[17] and the best is used in PolitiFact and GossipCop evaluations. The topic is weighted by TFIDF as given in (3)

$$W(t,d) = \text{Topic}(t,d) \times \text{TF-IDF}(t,d) \quad (3)$$

4.1.2 Hybrid Models

The base models are combined as follows:

1. *N-gram + Topic*: The TFIDF weighted n-gram model and TFIDF weighted topic model are stacked using Vertical Stack approach[23]
2. *N-gram + Word2Vec*: The TFIDF weighted n-gram matrix and word2vec matrix are fused by matrix multiplication using Python 3.6.4[23]
3. *Word2Vec + Topic*: The TFIDF weighted topic matrix and word2vec matrix are fused by matrix multiplication using Python 3.6.4[23]
4. *N-gram + Topic + Word2Vec*: The TFIDF weighted n-gram model and TFIDF weighted topic model that are stacked using Vertical Stack approach[23] are fused with word2vec matrix by matrix multiplication using Python 3.6.4[23]

4.2 Preprocessing

Before the datasets are used in the classification experiment, they are cleaned and formatted using the following steps:

1. tokenization using TweetTokenizer [24] to extract each word in the datasets.
2. stemming using WordNet Lemmatizer [24] to get the root words and their syntactic category.
3. removal of redundant instances
4. removal of punctuations
5. removal special characters and symbols
6. removal of hash symbols in hashtags
7. removal of English stop words.
8. change of all texts to lower case.

4.3 Classification

After preprocessing, the PolitiFact dataset consists a total of 968 instances, with 426 real news and 542 fake news instances, while the GossipCop consists a total of 20,796 instances, with 4,804 real news and 15,965 fake news instances. The distributions of the PolitiFact and GossipCop datasets for classification are presented in Table 1 and Table 2, respectively. By dividing each dataset into two samples with 80% as train sample and 20% as test sample, there are 774 instances in train sample and 194 instances in test sample of PolitiFact. Also, there are 16,615 instances in train sample and 4,154 instances in test sample.

Table 1. Data distribution for Classification of PolitiFact

Classification	Real News	Fake News	Total
Train	349	425	774
Test	77	117	194

Table 2. Data distribution for Classification of GossipCop

Classification	Real News	Fake News	Total
Train	3,861	12,754	16,615
Test	943	3,211	4,154

With the aid of Python Scikit-Learn 0.22 Library[17], the imbalanced train samples are first oversampled using synthetic minority oversampling technique (SMOTE)[25] before LogReg classifier is modelled. In the classification model, the LogReg, with its default parameter setting is used to learn from the train samples and predict the class of the instances based on the test sample.

4.4 Performance Evaluation

The performances of the features were compared based on accuracy, precision, recall and F-score as presented in equations (4) to (7). The equations rely on the true positive (TP), which is the number of real news that are correctly predicted; true negative (TN), which is the number of fake news that are correctly predicted; false positive (FP), which is the number of instances of fake news that have been incorrectly predicted as real news; false negative (FN), which is the number of instances of real news that have been incorrectly predicted as belonging to fake news.

$$\text{Accuracy (Acc)} = \frac{TP+TN}{TP+FP+FN+TN} \quad (4)$$

$$\text{Precision (P)} = \frac{\text{TP}}{\text{TP+FP}} \quad (5)$$

$$\text{Recall (R)} = \frac{\text{TP}}{\text{TP+FN}} \quad (6)$$

$$\text{F-Score (F1)} = \frac{2X(\text{Recall X Precison})}{(\text{Recall+Precision})} \quad (7)$$

5. RESULTS

The results of the performance of the models are presented in Table 3 and Table 4. Table 3 presents the results of performance of the models for PolitiFact, while Table 4 presents the results of the performance of the models for GossipCop.

The results in Table 3 show that for the base models, n-gram features record the highest accuracy of 0.80, precision of 0.79, recall of 0.78 and F1-score of 0.78 followed by word2vec, which records accuracy of 0.73, precision of 0.73, recall of 0.74 and F1-score of 0.73. The topic model records the worst performance, which shows that surface level features might not always be better than embedding features. For the hybrid models, the combination of n-gram and topic models records the highest accuracy of 0.77, precision of 0.76, recall of 0.76 and F1-score of 0.76 followed by combination of n-gram and word2vec with accuracy of 0.72, precision of 0.72, recall of 0.73 and F1-score of 0.72. The results contradict the outcomes of the base model in that despite the worst performance of topic model in the base model, its combination with n-gram model is better than the combination of n-gram and word2vec.

The results in Table 4 show that for the base models, n-gram features record the highest accuracy of 0.82, precision of 0.75, recall of 0.79 and F1-score of 0.77 followed by word2vec, which records accuracy of 0.78, precision of 0.71, recall of 0.76 and F1-score of 0.72. The topic model records the worst performance, which affirms that surface level features are not always better than embedding features. For the hybrid models, the combination of n-gram and topic models records the highest accuracy of 0.82, precision of 0.75, recall of 0.78 and F1-score of 0.76 followed by combination of n-gram and word2vec with accuracy of 0.78, precision of 0.71, recall of 0.76 and F1-score of 0.72. The results confirm that despite the worst performance of topic model in the base model, its combination with n-gram model is better than the combination of n-gram and word2vec.

Summarily, word n-gram features record the best performances in the classification of PolitiFact and GossipCop datasets followed by combination of word n-gram and topic model.

Table 3. Performance of Features for PolitiFact

Type	Model	Acc	P	R	F1
Base	N-gram	0.80	0.79	0.78	0.78
	Topic	0.60	0.55	0.53	0.51
	Word2Vec	0.73	0.73	0.74	0.73
Hybrid	N-gram+ Topic	0.77	0.76	0.76	0.76
	N-gram + Word2Vec	0.72	0.72	0.73	0.72
	Topic + Word2Vec	0.42	0.49	0.49	0.39
	N-gram + Topic + Word2Vec	0.40	0.45	0.48	0.36

Table 4. Performance of Features for GossipCop

Type	Model	Acc	P	R	F1
Base	N-gram	0.82	0.75	0.79	0.77
	Topic	0.51	0.51	0.51	0.47
	Word2Vec	0.78	0.71	0.76	0.72
Hybrid	N-gram+ Topic	0.82	0.75	0.78	0.76
	N-gram + Word2Vec	0.78	0.71	0.76	0.72
	Topic + Word2Vec	0.63	0.60	0.64	0.60
	N-gram + Topic + Word2Vec	0.58	0.57	0.60	0.54

The performance of the models are compared with the benchmarks for detection of fake news in FakeNewsNet such as logistic regression (LogReg) and Social Article Fusion models [9] in Table 5. The values in bold face are the benchmarks. From the results of PolitiFact, the n-gram model records accuracy of 0.80, precision of 0.79, recall of 0.78 and F1-score of 0.78, while Social Article Fusion model[9] records accuracy of 0.69, precision of 0.64, recall of 0.79 and F1-sore of 0.71. These show that the n-gram model outperforms Social Article Fusion model [9] in terms of accuracy, precision and F1-score. For GossipCop, the n-gram model records accuracy of 0.82, precision of 0.75, recall of 0.79 and F1-score of 0.77, while LogReg model [9] records accuracy of 0.82, precision of 0.90, recall of 0.72 and F1-sore of 0.80. These show that the n-gram model outperforms LogReg model[9] in terms of recall. However, the precision and F1-score of n-gram is lower than LogReg [9]. The performance of the n-gram model is more consistent for PolitiFact and GossipCop datasets.

Table 5. Comparison of the evaluated Features and the Benchmarks

Model	PolitiFact				GossipCop			
	Acc	P	R	F1	Acc	P	R	F1
LogReg [9]	0.64	0.76	0.54	0.63	**0.82**	**0.90**	**0.72**	**0.80**
Social Article Fusion [9]	**0.69**	**0.64**	**0.79**	**0.71**	0.80	0.82	0.75	0.79
N-gram	**0.80**	**0.79**	**0.78**	**0.78**	**0.82**	**0.75**	**0.79**	**0.77**
Topic	0.60	0.55	0.53	0.51	0.51	0.51	0.51	0.47

Word2Vec	0.73	0.73	0.74	0.73	0.78	0.71	0.76	0.72
N-gram+ Topic	0.77	0.76	0.76	0.76	0.82	0.75	0.78	0.76
N-gram + Word2Vec	0.72	0.72	0.73	0.72	0.78	0.71	0.76	0.72
Topic + Word2Vec	0.42	0.49	0.49	0.39	0.63	0.60	0.64	0.60
N-gram + Topic + Word2Vec	0.40	0.45	0.48	0.36	0.58	0.57	0.60	0.54

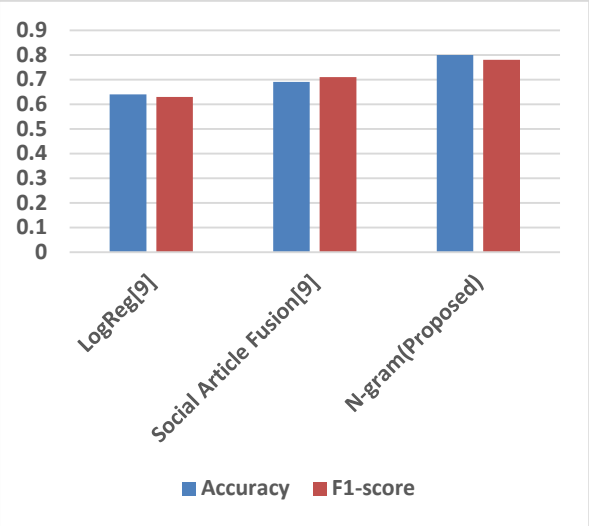

Fig. 2. Comparison of the performance of the proposed N-gram and the Benchmarks for PolitiFact

From Figure 2 and Figure 4 showing the charts for the comparison of the performances of the proposed n-gram and the benchmarks for PolitiFact and GossipCop, respectively, N-gram model outperforms both social article fusion and LogReg[9] for accuracy and F1 except for GossipCop in which N-gram model performs worst in terms of F1. An observation of the confusion matrices in Figure 3 shows that this is due to poor prediction of the real news because of bias towards majority class, which improved sampling techniques and more advanced machine learning techniques could solve.

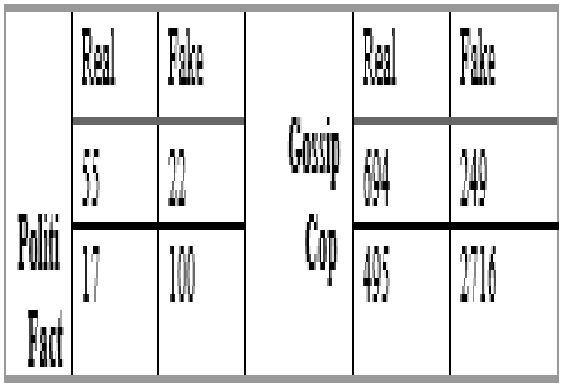

Fig. 3. Confusion Matrices for PolitiFact and GossipCop

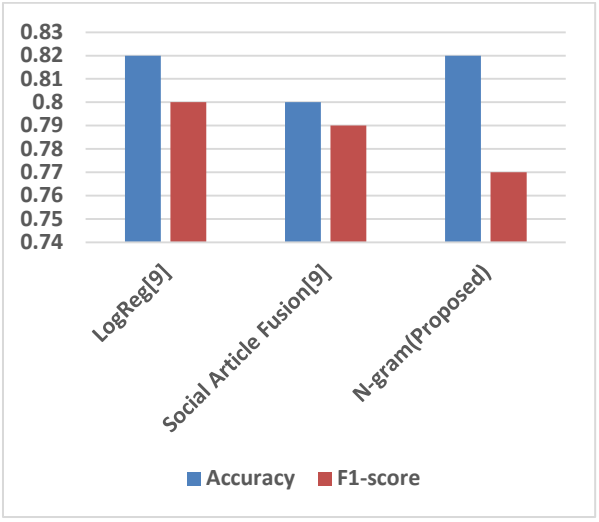

Fig. 4. Comparison of the performance of the proposed N-gram and the Benchmarks for GossipCop

6. CONCLUSION

This paper focuses on exploring n-gram, word embedding and topic models for content-based fake news detection in FakeNewsNet evaluation datasets. The work argues the need for evaluation of the datasets beyond the legacy work [9] in comparison with other datasets, which have been more broadly studied for content-based fake news detection. Based on the use of base models and their hybrids, n-gram is found to be the best among the models with accuracy of 0.80, precision of 0.79, recall of 0.78 and F1-score of 0.78 for PolitiFact; accuracy of 0.82, precision of 0.75, recall of 0.79 and F1-score of 0.77 for GossipCop. On the overall, the n-gram results are better than the benchmarks in terms of performance and consistency.

In future, bigram and trigram word n-gram and character n-gram features will be explored to improve the results of n-gram model. Also, deep learning algorithms such as CNN and LSTM together with more sampling techniques will be employed for the evaluation.

7. ACKNOWLEDGMENTS

The author thanks Kai Shu, Deepak Mahudeswaran, Suhang Wang, Dongwon Lee, and Huan Liu for free access to FakeNewsNet repository.

8. REFERENCES

[1] S. Vosoughi, M. N. E. O. Mohsenvand, and D. E. B. Roy, “Rumor Gauge : Predicting the Veracity of Rumors on Twitter r r,” ACM Trans. Knowl. Discov. Data, vol. 11, no. 4, 2017.

[2] R. Yan, Y. I. Li, W. Wu, D. Li, and Y. Wang, "Rumor Blocking through Online Link Deletion," ACM Trans. Knowl. Discov. Data, vol. 13, no. 2, 2019.

[3] W. Y. Wang, "' Liar , Liar Pants on Fire ': A New Benchmark Dataset for Fake News Detection," 2016.

[4] B. Andreas Hanselowski, Avinesh PVS and and F. C. Schiller, "Team Athene on the Fake News Challenge.," 2017. [Online]. Available: https://medium.com/@andre134679/%0Ateam-athene-on-the-fake-news-/%0Achallenge-28a5cf5e017b.

[5] BuzzFeedNews, "BuzzFeedNews," 2016. [Online]. Available: https://github.com/BuzzFeedNews/2016-10-facebook-factcheck/%0Ablob/master/data.

[6] M. Aldwairi and A. Alwahedi, "Detecting Fake News in Social Media Networks," Procedia Comput. Sci., vol. 141, pp. 215–222, 2018.

[7] S. Castelo, E. Nakamura, and J. Freire, "A Topic-Agnostic Approach for Identifying Fake News Pages," in Companion Proceedings of the 2019 World Wide Web Conference (WWW '19 Companion), p. 6pages.

[8] A. Thota, "Fake News Detection : A Deep Learning Approach," SMU Data Sci. Rev., vol. 1, no. 3, 2018.

[9] K. Shu, D. Mahudeswaran, S. Wang, D. Lee, and H. Liu, "FakeNewsNet : A Data Repository with News Content , Social Context and Dynamic Information for Studying Fake News on Social Media," Assoc. Adv. Artif. Intell., 2017.

[10] TampaBayTimes, "PolitiFact," Tampa Bay Times. [Online]. Available: https://www.politifact.com/.

[11] "GossipCop." [Online]. Available: https://www.gossipcop.com/.

[12] US, "PolitiFact." [Online]. Available: http://politifact.com.

[13] S. Yang, K. Shu, S. Wang, R. Gu, F. Wu, and H. Liu, "Unsupervised Fake News Detection on Social Media : A Generative Approach," 2019.

[14] B. Ghanem, P. Rosso, and F. Rangel, "Stance Detection in Fake News : A Combined Feature Representation," in Proceedings of the First Workshop on Fact Extraction and VERification (FEVER), 2018, pp. 66–71.

[15] K. Xu, F. Wang, H. Wang, and B. Yang, "Detecting Fake News Over Online Social Media via Domain Reputations and Content Understanding," TSINGHUA Sci. Technol., vol. 25, no. 1, pp. 20–27, 2020.

[16] P. Fortuna and S. Nunes, "A Survey on Automatic Detection of Hate Speech in Text," ACM Comput. Surv. 51, 4, vol. 51, no. 4, 2018.

[17] G. O. and D. E. Pedregosa F., Varoquaux G., Gramfort A., Michel V., Thirion B., "Scikit-learn: Machine Learning in Python," J. Mach. Learn. Res., vol. 12, pp. 2825–2830, 2011.

[18] S. Malmasi and M. Zampieri, "Challenges in Discriminating Profanity from Hate Speech," pp. 1–16, 2011.

[19] A. Gaydhani, V. Doma, S. Kendre, and L. Bhagwat, "Detecting Hate Speech and Offensive Language on Twitter using Machine Learning : An N-gram and TFIDF based Approach."

[20] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Distributed Representations of Words and Phrases and their Compositionality," pp. 1–9.R. Reh, "gensim Documentation," 2017. and M. I. J. D. M. Blei, A. Y. Ng, "Latent Dirichlet allocation," J. Mach. Learn. Res., vol. 3, no, no. 3, pp. 993– 1022, 2003.

[21] PythonTM, "Python 3.6.4." 2017.

[22] E. L. Steven Bird, Ewan Kliein, Analyzing Texts with Natural Language Toolkit: Natural Language Processing with Python, First. O'Reilly, 2009.

[23] N. V Chawla, K. W. Bowyer, and L. O. Hall, "SMOTE : Synthetic Minority Over-sampling Technique," vol. 16, pp. 321–357, 2002.