Ensemble Based Machine Learning Models for Hate Speech and Offensive Content Identification

Asha Hegde, Mudoor Devadas Anusha and Hosahalli Lakshmaiah Shashirekha

Department of Computer Science, Mangalore University, Mangalore, India

Abstract

The rapid growth of internet and mobile technology has accelerated the spread of Hate Speech and Offensive Content (HASOC) to a larger extent. Identifying HASOC is a real challenge as the social media content may contain code-mixed text in two or more languages. Hence, filtering HASOC on social media to curb its further spread and the damage it is going to create is the need of the day. Automated tools are required to filter HASOC as doing it manually is labour intensive and error prone. In this paper, we, team MUM, describe the models submitted to the HASOC in English and Indo-Aryan Languages 2021 shared task in Forum for Information Retrieval Evaluation (FIRE) 2021. The shared task consists of Subtasks 1A and 1B for English, Hindi and Marathi languages and Subtask 2 for code-mixed text in English-Hindi language pair. The proposed models are devised as ensemble of three Machine Learning (ML) classifiers, namely: Random Forest (RF), Multi-Layer Perceptron (MLP) and Gradient Boosting (GB). These ensemble models are trained using a combination of the Term Frequency — Inverse Document Frequency (TF-IDF) of different features like word uni-gram, character n-grams, Hashtag vectors (HastagVec) followed by using the pre-trained embeddings: word2Vec and Emo2Vec. The proposed approaches obtained 43rd, 23rd, 18th, 10th, and 15th rank for English, Hindi and Marathi Subtask 1A and Subtask 1B respectively (Marathi language does not have Subtask 1B) and 11th rank in Subtask 2 for code-mixed English-Hindi tweets.

Keywords

Emoji2vec, HashTag, Ensemble, Voting Classifier, Code-mixing

1. Introduction

Social media have become increasingly popular since a decade, engaging people in more online activities such as online shopping, using dating apps, sharing messages on Facebook, WhatsApp, Instagram, Twitter etc. These platforms also have given users' the opportunity to hide their real identity (if required) which is being used by the miscreants in a negative way to spread malicious and aggressive hate speech content such as racist, xenophobic and many other forms of verbal aggression. Various forms of hate speech have been analyzed and described by linguists [1]. The act of hate speech generally refers to disparaging a person or group based on certain characteristics, including but not limited to: race, ethnicity, sexual orientation, gender, religion, and national origin [2]. Such content is often considered harmful for a rationale and constructive debate [3]. Many countries have defined increasingly specific rules to deal with HASOC [4].

Forum for Information Retrieval Evaluation, December 13-17, 2021, India

hegdekasha@gmail.com (A. Hegde); anugowda251@gmail.com (M. D. Anusha); hlsrekha@gmail.com (H. L. Shashirekha)

https://mangaloreuniversity.ac.in/dr-h-l-shashirekha (H. L. Shashirekha)

Political scientists and legal scholars are also exploring ways of regulating the platforms to handle the hate speech without restricting the freedom offered by the social media.

Code-mixing is a natural phenomenon of combining linguistic units such as phrases, words or morphemes of two or more languages in a text or speech. In a multi-lingual country like India, people usually will know to use more than one language. However, due to technological limitations of keys in smart phones or keyboards in laptops/desktops, instead of writing comments in their native script, people tend to write comments mixing English and native language words in Roman script leading to code-mixed text. Identifying HASOC in code-mixed text is quite challenging due to the increasing volume and complexity of the code-mixed texts [5].

Social media platforms have a growing number of users and uncontrollable number of posts/comments are shared every second, making it impossible to trace and control the content manually. Code-mixing content adds an additional complexity to this task. Hence, there is need for automatic tools and techniques to detect HASOC quickly to avoid it's spread further and the damage it is going to create [6]. HASOC identification in English and Indo-Aryan Languages 2021[1] shared task in FIRE 2021[2], invites researchers to develop models for HASOC identification in English and Indo-Aryan languages [7], specifically for English [8], Hindi [8] and Marathi [9] and also for code-mixed English-Hindi [10] tweets. In this paper, we, team MUM, describe the models submitted to HASOC 2021 shared task. The proposed models are devised as ensemble of three ML classifiers, namely: RF, MLP and GB. These ensemble models are trained using a combination of the TF-IDF of different features like word uni-gram, character n-grams, HashtagVec followed by using the pre-trained embeddings: word2Vec and Emo2Vec.

Rest of the paper is organized as follows: Section 2 throws light on the recent work related to identifying HASOC and Section 3 describes the proposed methodology. Experiments and results are described in Section 4 followed by conclusion and future work in section 5.

2. Related Work

The topic of hate speech detection is of immense importance and is attracting numerous researchers. Several workshops and shared tasks are focusing on the detection of HASOC. Prominent among them are the HASOC shared task organized by FIRE in 2019 and 2020 which focuses on detecting HASOC in Dravidian and Indo-European languages. A brief description of few of the recent works related to detecting HASOC are given below:

Fazlourrahman et al. [11] proposed Transfer Learning (TL) approach, ensemble of ML algorithms, and ML-TL - a hybrid combination of the first two, to identify HASOC in HASOC 2020 shared task. In TL-based model, they used Universal Language Model Fine-Tuning (ULMFiT) - a pre-trained Language Model (LM) that represents the general features of a language. They used word/char n-grams features to train an ensemble of ML estimators: Support Vector Machine (SVM), RF and Logistic Regression (LR) classifiers with majority voting. Further, in the ML-TL hybrid model, they used ULMFiT model in the ensemble model instead of RF classifier keeping other two classifiers and their parameters similar to the previous ensemble model. The hybrid combination of ML-TL approach obtained macro F1-scores of 0.4979 and 0.5182 securing 21st

[1] https://hasocfire.github.io/hasoc/2021/index.html

[2] https://hasocfire.github.io/hasoc/2021/call-for-participation.html

and 8th place for Subtask A in English and Hindi respectively. Further, for Subtask B in English, ULMFiT model achieved macro F1-scores of 0.2517 and obtained 5th rank. For Subtask A in German language, ensemble of ML classifiers obtained macro F1-scores of 0.5044 and secured 11th rank. Shashirekha et al. [12] concatenated CountVectorizer, TfidfVectorizer and additional text-based features to build an ensemble of GB, RF and XGBoost (XGB) classifiers with soft voting to detect HASOC in Indo-European languages in HASOC 2020 shared task. Their models obtained 5th, 15th, 7th, 3rd, 13th, and 6th ranks with F1-scores of 0.5046, 0.2595, 0.5033, 0.2595, 0.5033 and 0.2488 for English, German and Hindi Subtasks A and B respectively.

Fazlourrahman et al. [13] in their work submitted to Hate Speech Spreader Detection (HSSD) shared task[3] in PAN 2021[4], combined traditional char n-grams and word n-grams with syntactic n-grams to train a voting classifier of three ML estimators: SVM, LR, and RF with hard and soft voting. They achieved 73% and 83% accuracies for English and Spanish languages respectively. Multilingual LSTM based model proposed by Aluru et al. [14] used seven different NN-based architectures, namely: a pooled Gated Recurrent Unit (GRU), LSTM, GRU with attention, 2D Convolution with Pooling, GRU with Capsule architecture, LSTM based Capsule architecture with attention, and a fine-tuned BERT model with LSTM. Among these seven models, LSTM based fine-tuned BERT achieved the best macro F1-scores of 0.7143, 0.7662, and 0.7329 for English, German, and Spanish respectively. TF-IDF of classical word and character n-grams based ML models presented by Aditya et al. [15] classifies Tweets automatically into three categories: Hateful, Offensive, and None. The authors conducted several experiments by changing the range of n-grams for both word and character n-grams to train three ML algorithms: NB, LR and SVM and obtained 95.6% accuracy as the best results for LR classifier with an optimal n-gram range 1 to 3. Davidson et al. [16] proposed a set of multi-class classifiers: LR, NB, Decision Trees, RF, and Linear SVM to characterize Tweets into one of three classes: Hate speech, Offensive but not hate speech, and Neither offensive nor hate speech. They used Tweets containing lexicon terms in 34,588 Twitter accounts and obtained a precision of 0.91, recall of 0.90, and an F1 score of 0.90. Ghanghor et al. [17] presented Transformer-based models to identify offensive contents in Dravidian languages at DravidianLangTech 2021 workshop. They explored multilingual Bidirectional Encoder Representations from Transformers (mBERT) uncased, mBERT cased, Cross-lingual Learning Model (XLM) with Robustly Optimized BERT and IndicBERT. Their mBERT cased model outperformed other models with weighted F1-score 0.75, 0.95, 0.71 and obtained 3rd, 3rd and 4th ranks for Tamil, Malayalam and English respectively.

3. Methodology

Several experiments were conducted on various learning models using various features and the ensemble of RF, MLP, and GB classifiers is proposed for all the subtasks. Figure 1 depicts the structure of the proposed ensemble model. The proposed model consists of the following steps:

[3]https://pan.webis.de/clef21/pan21-web/author-profiling.html
[4]https://pan.webis.de/clef21/pan21-web/

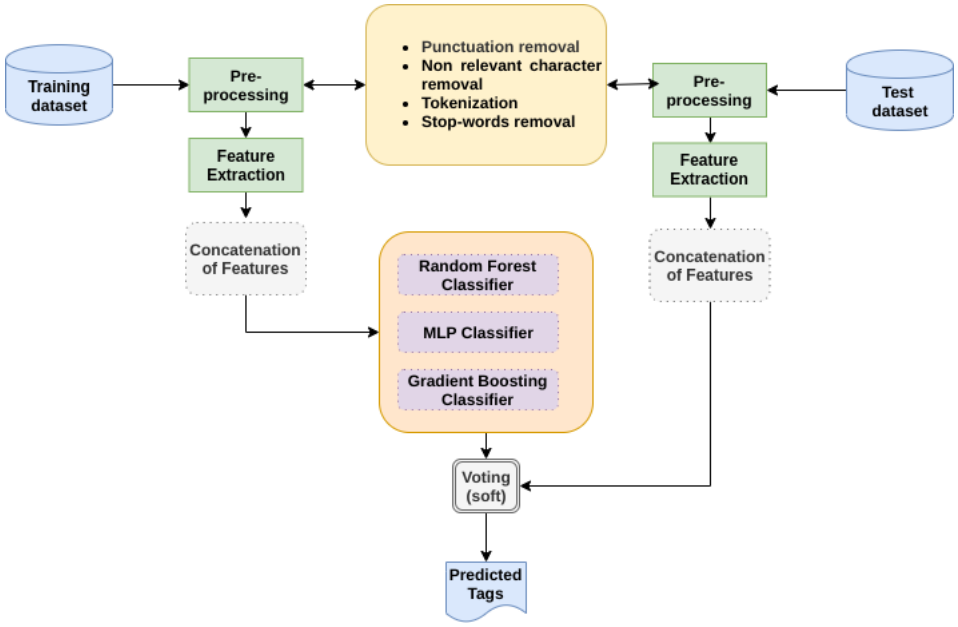

Figure 1: Structure of the proposed ensemble model

3.1. Pre-processing

Pre-processing is an essential approach to clean the textual content and transform it into a format that the ML algorithms can understand [18]. Emojis are a standardised set of small pictorial glyphs depicting everything from smiling faces to international flags. They are visual representations of emotions, objects and symbols that can be used individually or in groups and are powerful representations comparable to words. Hashtags are a combination of letters, numbers, and/or emoji preceded by the '#' symbol (e.g., #NoFilter) used to indicate the context or the core idea, which helps users to navigate the deluge of information. Micro-blogging services allow users to insert Hashtags into their posts to aid the gathering of relevant micro-blogs on a specific topic or event, as well as the diffusion of information [19]. Both Emojis and Hashtags have become popular on social media over the last decade. As most of the comments include Emojis and Hashtags, they are extracted and converted to vectors as discussed in Section 3.2.

A pipeline of pre-processing steps are used to remove the urls, punctuation, digits, unrelated characters and stopwords as these do not contribute to the task of any Text Classification (TC) in general. English stopwords list available in Natural Language Tool Kit (NLTK) and Hindi[5] and Marathi[6] stopwords lists available in github repository are used. Further, lemmmatization[7] is applied only for the English dataset to reduce the words to their root words and the pre-processed text is used to extract features.

[5]https://github.com/stopwords-iso/stopwords-hi
[6]https://github.com/stopwords-iso/stopwords-mr
[7]http://www.nltk.org/api/nltk.stem.html?highlight=lemmatizer

3.2. Feature Extraction

Feature extraction is the process of extracting features which are used to train the learning models. A combination of TF-IDF of words, char n-grams, pre-trained word embeddings followed by representing Emojis and Hashtags as vectors have exhibited good performance and hence are used in this work. The feature extraction steps are given below:

- **TF-IDF** expresses the relative importance between a word in the document and the entire corpus. For Subtask 1A and Subtask 1B: top 5,000 frequent character n-grams in the range 2 to 3 and all the word uni-grams are extracted from English, Hindi, and Marathi. Similarly, for Subtask 2 top 5000 frequent character n-grams in the range 2 to 3 and all the word uni-grams are extracted from code-mixed English-Hindi tweets. These n-grams are vectorized using the TFidfVectorizer[8].
- **Word2Vec** is a statistical method of learning word embeddings using the semantic relationship between the words such that similar words are placed together while dissimilar words are placed far apart in the representation [20]. fastText[9] pre-trained word vectors available at gensim library[10] are used to construct word embeddings for the Train and Test sets with a window size 5 in 300-dimensional space, for all the three languages for both Subtask 1A and 1B. Further, each document in the dataset is represented by the sum of the vector values of the words in that document.
- **Emo2Vec** are word-level representations that encode Emojis in Unicode standard into a fixed-sized, real-valued vectors embeddings. Emo2Vec[11] is an open-source representation for Emojis that maps Emojis into a 300-dimensional space similar to Google News word2Vec embeddings [21]. Since there are many Emojis, instead of removing them leading to loss of information they are extracted from the text and vectorized using pre-trained Emo2Vec.
- **Hashtag vector** (HashtagVec) is a customized vector representation that assign weights for Hashtags. Hashtags are extracted from the text and vectorized using Tf-IDFVectorizer.

The number of Emojis, Hashtags, and word uni-grams extracted from the Train and Test sets are given in Table 1. All the extracted features are concatenated together and are used to train the classifiers.

3.3. Classifier Construction

The performance of the model relies heavily on the features and the classifier used to carry out the classification. RF, MLP, and GB classifiers are used to identify HASOC. RF classifiers are well-suited to deal with high-dimensional noisy data [22]. This model is made up of a collection of decision trees, each of which is trained using a random subset of features and the prediction is obtained as a majority vote of all the trees in the forest. MLP classifiers are widely used in ML models because of their simplicity. This model is defined as a feed-forward NN that consists

[8] https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html
[9] https://fasttext.cc/docs/en/crawl-vectors.html
[10] https://pypi.org/project/gensim/
[11] https://github.com/uclnlp/emoji2vec/tree/master/pre-trained

Table 1
Details of Emojis and Hashtags extracted from HASOC 2021 dataset

Subtasks	Train Set	#Emojis	#Hash Tags	#Word uni-grams
Subtask 1	English	8,967	5,890	10,395
	Hindi	9,188	12,027	4,655
	Marathi	1,874	495	1,762
Subtask 2	Code-mixed English-Hindi tweets	4,158	2,330	10,797
Test Set				
Subtask 1	English	195	2,042	10,395
	Hindi	568	3,807	4,655
	Marathi	40	0	1,762
Subtask 2	Code-mixed English-Hindi tweets	1,161	847	10,797

of three types of layers: the input layer, the output layer, and one or more hidden layers [23]. The GB classifier is a powerful ML algorithm that has shown significant success in a variety of applications. The major benefit of using the GB algorithm is that it produces a robust classifier by converting a group of learners showing poor classification performance. This will benefit the regularisation methods that penalize different parts of the algorithm and improve the overall performance by reducing overfitting [24].

The performance of a classifier is also influenced by the dataset and no classifier produces good results for all datasets. As a result, no classifier is considered as the best classifier in general. Hence, an ensemble of classifiers, where the weakness of one classifier is overcome by the strength of another classifier produces better results compared to a single classifier [12]. The proposed ensemble model uses RF, MLP and GB - the three best performing classifiers and the prediction on the Test set is based on soft voting.

4. Experimental Setup and Results

HASOC 2021 shared task consists of two subtasks, Subtask 1 and Subtask 2. Subtask 1 is further divided into Subtask 1A and Subtask 1B for English, Hindi, and for Marathi (Marathi is not having Subtask 1B) where as Subtask 2 is for code-mixed English-Hindi tweets. It may be noted that the comments/posts labeled 'Offensive' only are used to train the models for Subtask 1B. The statistics of the dataset for Subtask 1 and Subtask 2 are given in Table 2.

Several experiments were conducted by combining various features and classifiers. HASOC word2Vec is constructed by merging the Train and Test datasets of HASOC 2019[12], 2020[13] and 2021 (only training data) collected from the official websites. These word vectors are concatenated with Emo2Vec, word uni-grams and character n-grams for Subtask 1A in English

[12]https://hasocfire.github.io/hasoc/2019/dataset.html
[13]https://hasocfire.github.io/hasoc/2020/dataset.html

Table 2
Details of HASOC 2021 dataset

Subtasks	#Posts	Training set		
		English	Hindi	Marathi
Subtask 1A	HOF	2,501	3,161	669
	NOT	1,342	1,433	1,205
Subtask 1B	PRFN	1,196	213	-
	HATE	683	566	-
	OFFN	622	654	-
Code-mixed Hindi tweets				
Subtask 2	HOF	2,841		
	NOT	2,899		

Table 3
Features used for the subtasks

Subtasks	Languages	Features used
Subtask 1A	English	HASOC word2Vec + Emo2Vec + word uni-grams + char n-grams
	Hindi	(pre-trained) word2Vec + Emo2Vec + HashtagVec + word uni-grams + char n-grams
	Marathi	(pre-trained) word2Vec + Emo2Vec + HashtagVec + word uni-grams + char n-grams
Subtask 1B	English	(pre-trained) word2Vec + Emo2Vec + HashtagVec + word uni-grams + char n-grams
	Hindi	(pre-trained) word2Vec + Emo2Vec + HashtagVec + word uni-grams + char n-grams
Subtask 2	Code-mixed English-Hindi Tweets	Emo2Vec + HastagVec + word uni-grams + char n-grams

language. Further, pre-trained word2Vec (instead of HASOC word2Vec) of the respective language, Emo2Vec, HashtagVec, uni-grams and character n-grams are used for rest of the languages in Subtask 1A and all three languages in Subtask 1B. For Subtask 2, Emo2Vec, HashtagVec, uni-grams and character n-grams features are concatenated. These features were used to train the ensemble models and the predictions of these models on the Test set were submitted to the organizers of the shared task for final evaluation and ranking. Details of the features used for the subtasks are summarized in Table 3. The predictions of the subtasks are evaluated by the organizers using macro F1-score and the results are summarized in Table 4.

5. Conclusion and Future work

In this paper, we, team MUM, have presented the description of the proposed models submitted to HASOC 2021 shared task in FIRE 2021 to identify HASOC in English, Hindi, Marathi and code-mixed English-Hindi tweets. Several experiments were conducted with various combinations of features and classifiers to identify HASOC in the dataset provided by the organizers. Our team

Table 4
Results of the proposed model

Subtasks	English		Hindi		Marathi	
	F1-score	Rank	F1-score	Rank	F1-score	Rank
Subtask 1A	0.8251	43	0.6323	18	0.7830	15
Subtask 1B	0.5771	23	0.4952	10	-	-
Code-mixed Hindi tweets						
Subtasks	F1-score	Rank				
Subtask 2	0.6721	11				

achieved competitive results for both the subtasks using an ensemble of three classifiers: RF, GB and MLP with soft voting. In future, we intend to investigate different sets of features and feature selection models, as well as various learning approaches for identifying problematic content.

6. Acknowledgements

We would like to thank the organisers of the HASOC 2021 shared task for organising this interesting shared task and for responding promptly to all of our inquiries. We would also like to thank the anonymous reviewers for their insightful comments.

References

[1] S. Jaki, T. De Smedt, M. Gwóźdź, R. Panchal, A. Rossa, G. De Pauw, Online Hatred of Women in the Incels. ME Forum: Linguistic Analysis and Automatic Detection, John Benjamins, 2019, pp. 240–268.
[2] C. Blaya, Cyberhate: A review and content analysis of intervention strategies, Elsevier, 2019, pp. 163–172.
[3] J. S. Vedeler, T. Olsen, J. Eriksen, Hate speech harms: A Social Justice Discussion of Disabled Norwegians' Experiences, Taylor & Francis, 2019, pp. 368–383.
[4] N. Asogwa, C. Ezeibe, The State, Hate Speech Regulation and Sustainable Democracy in Africa: A Study of Nigeria and Kenya, Taylor & Francis, 2020, pp. 1–16.
[5] A. Bohra, D. Vijay, V. Singh, S. S. Akhtar, M. Shrivastava, A Dataset of Hindi-English Code-Mixed Social Media Text for Hate Speech Detection, in: Proceedings of the Second Workshop on Computational Modeling of People's Opinions, Personality, and Emotions in Social Media, 2018, pp. 36–41.
[6] Z. Waseem, D. Hovy, Hateful Symbols or Hateful People? Predictive Features for Hate Speech Detection on Twitter, in: Proceedings of the NAACL Student Research Workshop, 2016, pp. 88–93.
[7] S. Modha, T. Mandl, G. K. Shahi, H. Madhu, S. Satapara, T. Ranasinghe, M. Zampieri,

Overview of the hasoc subtrack at fire 2021: Hate speech and offensive content identification in english and indo-aryan languages and conversational hate speech, in: FIRE 2021: Forum for Information Retrieval Evaluation, Virtual Event, 13th-17th December 2021, 2021.

[8] T. Mandl, S. Modha, G. K. Shahi, H. Madhu, S. Satapara, P. Majumder, J. Schäfer, T. Ranasinghe, M. Zampieri, D. Nandini, A. K. Jaiswal, Overview of the hasoc subtrack at fire 2021: Hate speech and offensive content identification in english and indo-aryan languages, Working Notes of FIRE (2021).

[9] S. Gaikwad, T. Ranasinghe, M. Zampieri, C. M. Homan, Cross-lingual Offensive Language Identification for Low Resource Languages: The Case of Marathi, 2021.

[10] S. Satapara, S. Modha, T. Mandl, H. Madhu, P. Majumder, Overview of the HASOC Subtrack at FIRE 2021: Conversational Hate Speech Detection in Code-mixed Language, in: FIRE (Working Notes), 2021.

[11] F. Balouchzahi, H. L. Shashirekha, LAs for HASOC-Learning Approaches for Hate Speech and Offensive Content Identification., in: FIRE (Working Notes), 2020, pp. 145–151.

[12] M. D. Anusha, H. L. Shashirekha, An Ensemble Model for Hate Speech and Offensive Content Identification in Indo-European Languages., in: FIRE (Working Notes), 2020, pp. 253–259.

[13] F. Balouchzahi, H. L. Shashirekha, G. Sidorov, HSSD: Hate Speech Spreader Detection using N-grams and Voting Classifier, in: CEUR Workshop Proceedings, 2021, pp. 1829–1836. URL: http://ceur-ws.org/Vol-2936/paper-156.pdf.

[14] S. S. Aluru, B. Mathew, P. Saha, A. Mukherjee, Deep Learning Models for Multilingual Hate Speech Detection, 2020.

[15] A. Gaydhani, V. Doma, S. Kendre, L. Bhagwat, Detecting Hate Speech and Offensive Language on Twitter using Machine Learning: An n-gram and tfidf Based Approach, 2018.

[16] T. Davidson, D. Warmsley, M. W. Macy, I. Weber, Automated Hate Speech Detection and the Problem of Offensive Language, in: Proceedings of the Eleventh International Conference on Web and Social Media, ICWSM 2017, AAAI Press, 2017, pp. 512–515. URL: https://aaai.org/ocs/index.php/ICWSM/ICWSM17/paper/view/15665.

[17] N. Ghanghor, P. Krishnamurthy, S. Thavareesan, R. Priyadharshini, B. R. Chakravarthi, IIITK@DravidianLangTech-EACL2021: Offensive language identification and meme classification in Tamil, Malayalam and Kannada, in: Proceedings of the First Workshop on Speech and Language Technologies for Dravidian Languages, Association for Computational Linguistics, Kyiv, 2021, pp. 222–229. URL: https://aclanthology.org/2021.dravidianlangtech-1.30.

[18] D. Kumar, R. Cohen, L. Golab, Online Abuse Detection: The Value of Preprocessing and Neural Attention Models, in: Proceedings of the Tenth Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis, 2019, pp. 16–24.

[19] A. Alsini, A. Datta, J. Li, D. Huynh, Empirical Analysis of Factors Influencing Twitter Hashtag Recommendation on Detected Communities, in: International Conference on Advanced Data Mining and Applications, Springer, 2017, pp. 119–131.

[20] T. Mikolov, W.-t. Yih, G. Zweig, Linguistic Regularities in Continuous Space Word Representations, in: Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2013, pp.

746–751.
[21] B. Eisner, T. Rocktäschel, I. Augenstein, M. Bošnjak, S. Riedel, emoji2vec: Learning emoji representations from their description, 2016.
[22] M. Z. Islam, J. Liu, J. Li, L. Liu, W. Kang, A Semantics Aware Random Forest for Text Classification, in: Proceedings of the 28th ACM International Conference on Information and Knowledge Management, 2019, pp. 1061–1070.
[23] M. Bounabi, K. El Moutaouakil, K. Satori, A Probabilistic Vector Representation and Neural Network for Text Classification, in: International Conference on Big Data, Cloud and Applications, Springer, 2018, pp. 343–355.
[24] V. Athanasiou, M. Maragoudakis, A Novel, Gradient Boosting Framework for Sentiment Analysis in Languages where NLP Resources are Not Plentiful: A Case Study for Modern Greek, Multidisciplinary Digital Publishing Institute, 2017, p. 34.