Tweet Length Matters: A Comparative Analysis on Topic Detection in Microblogs

Furkan Şahinuç and Cagri Toraman(✉)

Aselsan Research Center, Ankara, Turkey
{fsahinuc,ctoraman}@aselsan.com.tr

Abstract. Microblogs are characterized as short and informal text; and therefore sparse and noisy. To understand topic semantics of short text, supervised and unsupervised methods are investigated, including traditional bag-of-words and deep learning-based models. However, the effectiveness of such methods are not together investigated in short-text topic detection. In this study, we provide a comparative analysis on topic detection in microblogs. We construct a tweet dataset based on the recent and important events worldwide, including the COVID-19 pandemic and BlackLivesMatter movement. We also analyze the effect of varying tweet length in both evaluation and training. Our results show that tweet length matters in terms of the effectiveness of a topic-detection method.

Keywords: Microblog · Short text · Topic detection · Tweet

1 Introduction

Online social networks, such as microblogs, are rich sources to share opinion and information, as well as collaborate with other users. Public discussion can be about various topics. Finding their topic labels can provide semantic basement and understanding for many applications; such as information filtering [2], new event detection and tracking [1], sentiment analysis [6], and opinion mining [10].

Microblogs are generally characterized as having short, informal, and noisy text. Tweets are one of the most popular example of microblogs. Finding their topics can be challenging due to the aforementioned characteristics. Given a set of microblogs, or tweets in this study, the task is to detect a single coarse-grained topic label for each one. We refer to this task as *tweet topic detection*.

Several methods are proposed for topic detection. Topic Detection and Tracking aims to monitor news stories not seen before, and group individual topics [1]. Topic modeling methods, such as LDA [4], discover thematic clusters of documents as mixture of probability distributions. Rather than finding topic groups in an unsupervised way, our task is a supervised classification. Traditional methods encode documents in the bag-of-words model, and employ state-of-the-art

D. Hiemstra et al. (Eds.): ECIR 2021, LNCS 12657, pp. 471–478, 2021.
https://doi.org/10.1007/978-3-030-72240-1_50

classifiers, such as SVM. However, such methods mostly rely on word occurrence, and thereby suffer from sparsity and vocabulary mismatch, which are likely to be observed in short text. With the recent developments in deep learning, documents can be encoded to capture advanced semantics with neural networks and word embeddings [17]. Words are not assumed to be independent as in bag-of-words. Text semantics are captured sequentially using word order and positions to get bidirectional contextual representations, as in the Transformer model [22].

There are many efforts to overcome sparsity and vocabulary mismatch in tweet classification. Tweet-specific features are extracted for classification [14]. Topic memory networks are employed for short text classification [25]. Topically enriched word embeddings are used for topic detection [14]. Neural models, such as RNN and LSTM, are employed to detect discrimination-related tweets [24], and CNN for Twitter sentiment analysis [12]. Transformer-based language models, such as BERT [9], are employed in disaster-related tweet detection [20].

Our contributions are the followings. (i) Although the existing studies cover various methods for tweet classification on different domains, there is still a lack of comparative analysis for tweet topic detection. We provide a comparative analysis of both traditional and recent methods for topic detection of short text, particularly tweets. (ii) Short text is mostly studied in terms of average length (number of words). We provide a detailed analysis for the effect of the length of short text in both evaluation and training. (iii) We construct a tweet dataset with topic labels related to recent and important events, including the COVID-19 pandemic and the BlackLivesMatter movement.

2 Topic Detection in Microblogs

In this section, we select and explain six methods related to tweet topic detection; namely, Boolean search [15], topic modeling [4], bag-of-words [15], word embeddings [5], neural network [13], and Transformer-based language model [9].

Boolean Search. Inverted index keeps a dictionary of words, and for each word, a list that holds the documents that words occur in [15]. Query keywords are searched efficiently on an inverted index by Boolean search operations. We assign topics to tweets based on any keyword match by the Boolean OR operator. We pre-determine five query keywords for each topic based on the most frequent hashtags. In case of matching more than one topic, we assign off-topic. To find more matches, words are stemmed with the Snowball stemmer for indexing.

Topic Modeling. Topic modeling is a probabilistic method that finds coherent topic distributions in the given documents in an unsupervised way. We use Latent Dirichlet Allocation (LDA) [4] for topic modeling. To utilize topic distributions for supervised topic detection, we use the topic distribution of a document as its feature vector. We then employ Support Vector Machines (SVM) for training.

Bag-of-Words. Bag-of-words is a document encoding method based on vector space model, where each document is represented in a fixed length of vectors [15]. Each vector consists of identifiers for terms in documents. We use TF-IDF

(Term Frequency-Inverse Document Frequency) term weighting [15]. We employ SVM for training. In this method, words are assumed to be independent, and grammar structure is not preserved.

Word Embeddings. Word embeddings are the encoded vector representations for words in an embedding space that projects semantical similarities [16]. Word embeddings are divided into contextual and non-contextual ones. Contextual embeddings have different vectors according to the text that they occur in, while non-contextual embeddings have static vectors regardless of context. This method considers non-contextual embeddings, while contextual ones are examined in Transformer-based language models. We use FastText [5], which is the successor of Word2Vec [16] and GloVe [19], but considers sub-word embeddings by n-grams. To obtain sentence embeddings for tweets, we get the average of word embeddings with L2 normalization, which divides the sum of embeddings by the length of a vector in the Euclidean space. We use a softmax layer to compute the probabilities for topic labels.

Neural Networks. Artificial neural networks have significant interest in the last decade, such as Recurrent Neural Network (RNN) and Convolutional Neural Networks (CNN), to process text sequentially and get neural embeddings. We select CNN that leverages local features in hidden layers of networks with convolving filters. CNN achieves remarkable results in natural language processing tasks [12,13]. Based on [13], we train CNN for sentence classification with one layer of convolution on randomly initialized word embeddings.

Transformer-Based Language Models. Transformer is a deep learning-based architecture that uses self-attention for each token over all tokens [22]. Similar to RNN and CNN, text order is preserved; but Transformer processes text sequence without recurrent neural structures, instead with self-attention that keeps positional embeddings. We select BERT [9], which is a deep learning-based language model built on bidirectional contextual representations of words by considering word positions and context with Transformer. To fine-tune BERT, we add a softmax layer with the cross-entropy loss function. The CLS sentence embeddings provided by the last layer of BERT are given as input to this additional layer.

3 Experiments

3.1 Experimental Setup

Dataset Construction. We collect around 100 million tweets in English, 21% of which have at least one hashtag, from Twitter API between April 07, 2020 and June 15, 2020. We select six important topics that occur in the top-100 most frequently used hashtags. The topics are the COVID-19 pandemic, “Black Lives Matter” (BLM) movement, Korean popular music (K-Pop), Bollywood movies and series, gaming consoles, and U.S. politics. We notice that many hashtags belong to the same topic. We assign topics to tweets according to the

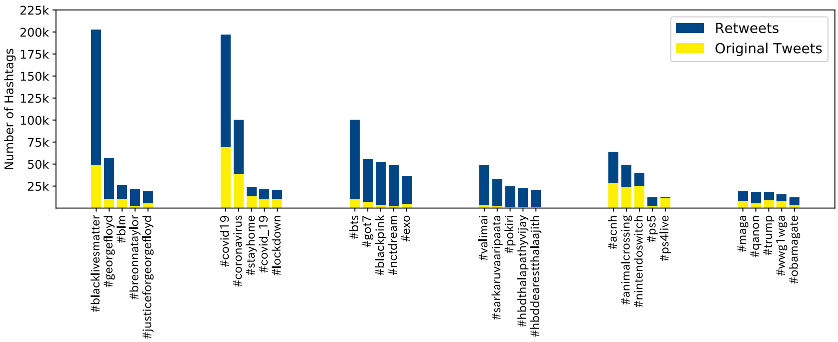


Fig. 1. The most frequently used five hashtags for each topic (BLM, COVID-19, K-pop, Bollywood movies, gaming, and U.S. politics, respectively).

predetermined set of hashtags for each topic. Figure 1 displays the top-5 most frequently used hashtags for each topic. A significant part of the hashtags are observed in retweets for K-Pop and Bollywood.

We apply the following cleaning steps to construct our dataset. (i) We exclude retweets, since duplicate contents would cause bias in results. (ii) We ignore the tweets with multiple hashtags from different topics, and the tweets with less than three words. (iii) We remove the words with less than three and more than 15 characters; as well as hashtags, mentions, and URLs. We keep words with only alpha-numeric characters. Words are lowercased. The NLTK lemmatization [3] is applied. (iv) We randomly select out-of-topic tweets that contain no related hashtag to our topics, which makes seven classes. The size of out-of-topic is chosen to be approximately 10% of the size of whole dataset. The final version of our dataset has 354,310 tweets[1]. The average length (number of words) is 13.4. The total numbers of tweets by topics are given in Table 1.

Table 1. The total number of tweets in our topic-detection dataset.

BLM	COVID-19	K-Pop	Bollywood	Gaming	U.S. Politics	Out-of-Topic	**Total**
61,672	139,036	45,817	9,661	36,613	26,373	35,138	**354,310**

Methodology. We use scikit-learn [18] for bag-of-words and topic modeling. We limit the vector size to 10,000 features, and remove the English stop words provided by scikit-learn. For LDA, we choose the number of topics as 50, based on the preliminary experiments. We use Linear SVC with one-vs-rest multi-classification for both models. For word embeddings, we use FastText's classification module [11] by choosing the vector dimension as 100. For neural networks, we follow CNN-based sentence classification [13], and use TensorFlow[2] with default parameters.

[1] The dataset can be accessed in https://github.com/avaapm/ECIR2021.

[2] https://github.com/dennybritz/cnn-text-classification-tf.

For Transformer-based models, we use DistilBERT [21] uncased model by HuggingFace [23] for the sake of efficiency.

We design two experiments: (i) We compare six important topic-detection methods, by applying 10-fold cross validation and reporting the weighted F1 score to evaluate effectiveness. The pairwise differences between the methods are statistically validated by using the two-tailed paired t-test at a 95% interval with Bonferroni correction. (ii) We analyze effectiveness for varying tweet lengths from 4 to 40 words to understand the behavior of the topic-detection methods.

3.2 Experimental Results

Comparison of Topic-Detection Methods. The comparison results are given in Table 2. We observe that (i) Boolean search has a poor performance, possibly due to dynamic dictionary in tweets. (ii) CNN-based topic detection statistically significantly outperforms other methods in short text, except BERT-based topic detection (we also validate that BERT statistically significantly outperforms others too, except CNN). We fine-tune BERT to provide a classification layer, but one can pre-train BERT for short and informal text to improve its effectiveness. (iii) Bag-of-words and topic modeling have lower scores, compared to CNN and BERT, possibly due to the sparsity of short text. (iv) FastText performs poor, possibly due to the fact that we employ pre-trained non-contextual word embeddings, not fine-tuned on the changing context of microblogs.

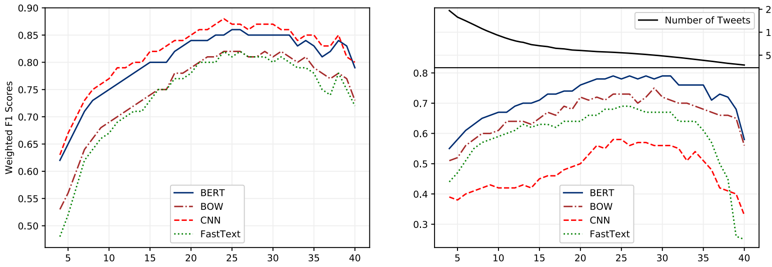
Table 2. The effectiveness results for topic detection in short text. The means of 10-fold cross-validation are reported. • indicates statistical significant difference at a 95% interval (with Bonferroni correction $p < 0.01$) in pairwise comparisons between the highest performing method and others (except the one with ∘).

Method	**Weighted F1 Score**
Boolean search on inverted index	0.202 ± 0.0002
Topic modeling (LDA) with SVM	0.456 ± 0.0001
Bag-of-words (TF-IDF) with SVM	0.672 ± 0.0003
Word embeddings (FastText)	0.649 ± 0.0002
Neural networks (CNN)	$0.754\bullet \pm 0.0019$
Transformer-based language models (BERT)	$0.739\circ \pm 0.0002$

Effect of Tweet Length. In this experiment, we employ the highest performing four models. Figure 2 shows the effectiveness of each model for varying tweet length (number of words). In Fig. 2a, we keep all train instances regardless of their length to show the effect of tweet length in evaluation. In Fig. 2b, we use the distinct subsets of training data to show the effect of tweet length in training. Each subset contains tweets with the same length.

In Fig. 2a, we observe that (i) the effectiveness of all methods decreases as tweet length in evaluation gets shorter. We thereby state that tweet length matters in evaluation. BERT and CNN have better performance in shorter tweets, compared to the others. (ii) The highest results for all methods are seen when tweet length is between 20 and 30 words. (iii) The highest performing method is CNN, while BERT challenges especially in extremely short and long tweets.

In Fig. 2b, we observe that (i) unlike the previous results, CNN performs poor when training data is limited to the same length. We thereby state that tweet length matters in training. CNN applies padding to input embedding matrix according to the longest tweet length [13]. Since this setup focuses on a specific length in training, CNN does not apply padding and model size gets smaller, which could be the reason of its poor performance. (ii) BERT outperforms the others in this setup, i.e. BERT is more robust to text length in training. (iii) Since the number of train instances gets too small as text length increases, effectiveness gets deteriorated after 30 words. However, BERT is more robust to train size, compared to other methods. Bag-of-words has also good performance in longer text, as expected due to the lower degree of sparsity.



(a) Training set includes tweets with all lengths. X-axis represents tweet length in test set. Number of tweets for each length is the same for training.

(b) Training set includes tweets with the same length. X-axis represents tweet length in both training and test sets. Number of tweets for each length is given at the top.

Fig. 2. The effect of tweet length (number of words) on topic-detection methods.

4 Conclusion and Future Work

We provide a comparative analysis of traditional and recent methods for topic detection in short text. We construct a tweet dataset with the recent events, including the COVID-19 pandemic and BlackLivesMatter movement. Our experimental results show that the sentence embeddings based on a neural model (CNN) and a Transformer-based language model (BERT) obtain the highest effectiveness scores. We also show that tweet length matters in both evaluation and training for the effectiveness of a topic-detection method. In future work, we plan to investigate other sentence embeddings, such as InferSent [8] or Universal Sentence Encoder [7]. The effect of tweet length can be further analyzed in different short-text datasets, such as news snippets.

References

1. Allan, J., Papka, R., Lavrenko, V.: On-line new event detection and tracking. In: Proceedings of SIGIR, pp. 37–45 (1998). https://doi.org/10.1145/290941.290954
2. Belkin, N.J., Croft, W.B.: Information filtering and information retrieval: two sides of the same coin? Commun. ACM **35**(12), 29–38 (1992). https://doi.org/10.1145/138859.138861
3. Bird, S., Klein, E., Loper, E.: Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit. O'Reilly Media, Inc., Newton (2009)
4. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent Dirichlet allocation. J. Mach. Learn. Res. **3**(1), 993–1022 (2003). https://doi.org/10.5555/944919.944937
5. Bojanowski, P., Grave, E., Joulin, A., Mikolov, T.: Enriching word vectors with subword information. Trans. Assoc. Comput. Linguist. **5**, 135–146 (2017). https://doi.org/10.1162/tacl_a_00051
6. Van Canneyt, S., Claeys, N., Dhoedt, B.: Topic-dependent sentiment classification on Twitter. In: Hanbury, A., Kazai, G., Rauber, A., Fuhr, N. (eds.) ECIR 2015. LNCS, vol. 9022, pp. 441–446. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-16354-3_48
7. Cer, D., et al.: Universal sentence encoder for English. In: Proceedings of EMNLP: System Demonstrations, pp. 169–174 (2018). https://doi.org/10.18653/v1/D18-2029
8. Conneau, A., Kiela, D., Schwenk, H., Barrault, L., Bordes, A.: Supervised learning of universal sentence representations from natural language inference data. In: Proceedings of EMNLP, pp. 670–680 (2017). https://doi.org/10.18653/v1/D17-1070
9. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. In: Proceedings of NAACL-HLT, pp. 4171–4186 (2019). https://doi.org/10.18653/v1/N19-1423
10. Fang, A., Ounis, I., Habel, P., Macdonald, C., Limsopatham, N.: Topic-centric classification of Twitter user's political orientation. In: Proceedings of SIGIR, pp. 791–794 (2015). https://doi.org/10.1145/2766462.2767833
11. Joulin, A., Grave, E., Bojanowski, P., Douze, M., Jégou, H., Mikolov, T.: Fasttext.zip: compressing text classification models. arXiv preprint arXiv:1612.03651 (2016)
12. Kalchbrenner, N., Grefenstette, E., Blunsom, P.: A convolutional neural network for modelling sentences. In: Proceedings of ACL, pp. 655–665 (2014). https://doi.org/10.3115/v1/P14-1062
13. Kim, Y.: Convolutional neural networks for sentence classification. In: Proceedings of EMNLP, pp. 1746–1751 (2014). https://doi.org/10.3115/v1/D14-1181
14. Li, Q., Shah, S., Liu, X., Nourbakhsh, A., Fang, R.: Tweetsift: tweet topic classification based on entity knowledge base and topic enhanced word embedding. In: Proceedings of CIKM, pp. 2429–2432 (2016). https://doi.org/10.1145/2983323.2983325

15. Manning, C.D., Schütze, H., Raghavan, P.: Introduction to Information Retrieval. Cambridge University Press, Cambridge (2008). https://doi.org/10.1017/CBO9780511809071
16. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Proceedings of NIPS, pp. 3111–3119 (2013)
17. Onal, K.D., et al.: Neural information retrieval: at the end of the early years. Inf. Retrieval **21**(2–3), 111–182 (2018). https://doi.org/10.1007/s10791-017-9321-y
18. Pedregosa, F., et al.: Scikit-learn: machine learning in python. J. Mach. Learn. Res. **12**, 2825–2830 (2011)
19. Pennington, J., Socher, R., Manning, C.: GloVe: global vectors for word representation. In: Proceedings of EMNLP, pp. 1532–1543 (2014). https://doi.org/10.3115/v1/D14-1162
20. Ray Chowdhury, J., Caragea, C., Caragea, D.: Cross-lingual disaster-related multi-label tweet classification with manifold mixup. In: Proceedings of ACL: Student Research Workshop, pp. 292–298 (2020). https://doi.org/10.18653/v1/2020.acl-srw.39
21. Sanh, V., Debut, L., Chaumond, J., Wolf, T.: DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. In: NeurIPS EMC^2 Workshop (2019)
22. Vaswani, A., et al.: Attention is all you need. In: Proceedings of NIPS, pp. 5998–6008 (2017)
23. Wolf, T., et al.: Huggingface's transformers: state-of-the-art natural language processing. arXiv preprint arXiv:1910.03771 (2019)
24. Yuan, S., Wu, X., Xiang, Y.: Incorporating pre-training in long short-term memory networks for tweets classification. In: Proceedings of IEEE ICDM, pp. 1329–1334 (2016). https://doi.org/10.1109/ICDM.2016.0181
25. Zeng, J., Li, J., Song, Y., Gao, C., Lyu, M.R., King, I.: Topic memory networks for short text classification. In: Proceedings of EMNLP, pp. 3120–3131 (2018). https://doi.org/10.18653/v1/D18-1351