

Self-Attentive Neural Network for Hashtag Recommendation

Delu Yang, Rui Zhu and Yang Li*

College of Information and Computer Engineering, Northeast Forestry University, Harbin 150040, China

Received 7 December 2018; Accepted 19 May 2019

Abstract

A hashtag is a type of metadata tag used on social networks and can help people search for specific topics or content. To capture the interactive information between words and understand the content of microblog posts deeply, this study proposed a neural network model based on a word-level self-attention mechanism. Given a microblog post, the weight of each word was calculated through a self-attention mechanism, and then the representation of a microblog post was obtained through the weighted summation of words. Finally, a multi-layer perceptron was adopted on the representation of a microblog post to perform the classification. The effectiveness of the proposed model was verified through experiments of large-scale datasets. Results show that: (1) introducing word-level self-attention mechanism into hashtag recommendation is effective. (2) In comparison with the baseline methods used in previous studies, such as convolutional neural network or long short-term memory network, the proposed self-attentive neural networks can provide a more accurate representation of a microblog post and significantly improve the F-score of hashtag recommendation on the same dataset. This study provides references for the methods and evaluation of short-text hashtag recommendations, such as microblogs.

Keywords: Hashtag Recommendation, Self-Attention, Neural Network

1. Introduction

Social networking services (SNSs), such as Twitter, Facebook, and Google+, have attracted millions of users to publish and share the most up-to-date information, emergent social events, and personal opinions [1]. As a typical representative of SNSs, microblog services have become increasingly popular. For example, Twitter [2] has over 240 million active users, who generate approximately 500 million tweets every day. The massive amount of information generated daily makes discovering the hidden information in that data difficult. To organize this information accurately and effectively, many microblogging services allow users to create and use hashtags by placing the pound sign (#), typically in front of a word or unspaced phrase in a post. A hashtag archive is consequently collected into a single stream under the same hashtag. For example, the hashtag #HarryPotter allows users to find all posts that have been tagged with this hashtag. Hashtags are confirmed to be important for many applications in microblogs, including microblog retrieval [3], query expansion [4], and sentiment analysis [5-7]. However, only and approximately 11% of tweets are annotated with one or more hashtags [8].

Therefore, scholars have proposed the problem of automatic hashtag recommendations for microblogs. The key to the hashtag recommendation task is to understand the content of the microblog post. Most methods depend on sparse lexical features, including bag-of-word (BoW) models and exquisitely designed patterns. However, feature engineering is labor-intensive, and the sparse and discrete features cannot effectively encode semantic and syntactic

word information. Considering the capacity of neural network models to learn feature representation effectively, their advantages in hashtag recommendation research have gradually emerged. Therefore, using deep neural networks to represent the features of microblog post accurately, understand the content deeply, and then improve the accuracy of a hashtag recommendation is an urgent problem to be solved.

Therefore, this study proposes to use self-attention mechanism between words [9] to capture the interactive information of words in a microblog post. The proposed self-attentive neural network is adopted to learn the feature weights of each word and combine the representations of words to obtain the representation of the entire microblog post. This study aims to learn the coupling relationship between words and hashtags in a microblog post and improve the accuracy of hashtag recommendations.

2. State of the art

Considerable research has focused on the field of hashtag recommendation. The practical approaches can be divided into two categories, namely, content-based and collaborative filtering methods. Content-based methods use different techniques to measure the content of tweets to identify relevant hashtags from historical tweets [10,11]. Zangerle et al. [10] presented an approach based on similar microblogs and the hashtags in these microblogs. Mazzia and Juett [12] used the naive Bayes (NB) model to calculate the maximum posterior probability of a hashtag if all words in the microblog are known. Krestel et al. [13] used a topic model based on the latent Dirichlet allocation (LDA) [14] to train the tags and words in the text together to obtain the topic

*E-mail address: calliyang@126.com

ISSN: 1791-2377 © 2019 Eastern Macedonia and Thrace Institute of Technology. All rights reserved.

doi:10.25103/jestr.122.15

distribution of each document and the words. Ding et al. [15] considered the text and its labels as different descriptions of the same resource; these authors used the topic model and single-word alignment techniques to learn the probability of each word being aligned with the label under a particular topic. When performing the prediction task, these authors calculated the ranking score of a certain hashtag relative to each microblog in accordance with the probability distribution of the topic and word alignment.

Collaborative filtering is a popular technique for recommendation systems. User preferences are considered by building a model from a user's past behaviors and similar decisions made by other users. Kywe et al. [16] proposed a collaborative filtering model based on a user group similar to the target user or based on a microblog similar to the target microblog to obtain the hashtags. Wang et al. [17] presented a joint model based on topic modeling and collaborative filtering to exploit local (current microblog content and the user) and global (hashtag-related content and usage preferences of like-minded users) information. Zhao et al. [18] used a hashtag-LDA recommendation approach that combined user profile- and LDA-based collaborative filtering. These authors jointly modeled the relations between users, hashtags, and words through latent topics.

In addition to these works, scholars have attempted to incorporate external information to improve hashtag recommendation. Li et al. [19] incorporated the topic-enhanced word embeddings, tweet entity data, hashtag frequency, temporal data, and tweet URL domain information to recommend hashtags for microblogs. Ma et al. [20] explored the latent relationship between tweet content, user interest, time, and hashtag. Motivated by the observation that most hashtags experience a life cycle of increasing and decreasing popularity, Lu et al. [21] developed a model that captures the temporal clustering effect of latent topics in tweets. Kowald et al. [22] proposed a cognitive-inspired hashtag recommendation algorithm that was based on temporal usage patterns of hashtags derived from empirical evidence (individual and social hashtag reuse patterns).

Recently, some attempts to use deep neural models for hashtag recommendation have been made [23-25]. Gong et al. [25] proposed a method incorporating textual and visual information. After observing that hashtags indicate the primary topics of microblog posts, Li et al. [26] proposed an attention-based long short-term memory (LSTM) model that incorporated topic modeling into the LSTM architecture through an attention mechanism. Most previous studies on hashtag recommendation have only considered the word content of microblogs. Li et al. [27] developed a topical co-attention mechanism that combines content and topic attention into a neural network. Zhu et al. [28] determined that syntactic information can effectively identify hashtags and applied the Tree-LSTM with syntactic information for hashtag recommendation. In the present study, we aim to integrate the self-attention mechanism between words into a deep neural network for hashtag recommendation.

The abovementioned works have used shallow textual features to perform hashtag recommendations. However, feature engineering is generally intensive, and the sparse and discrete features cannot effectively encode the semantic and syntactic information of words. Some methods have also aimed to learn deep textual features through deep neural networks. In comparison with the traditional methods, neural networks have achieved better results in this task. To capture the interaction between words, in the present study, we

propose using self-attention mechanism between words to obtain the text representation of microblog posts. Although the self-attention mechanism has been confirmed to improve the performance of numerous tasks, such as machine translation and natural language inference [29], the text in this task is spoken language, which is completely different from those in previous research. Whether the interactions between words in a short text can help improve text representation remains a question worthy of discussion. To the best of our knowledge, this study is the first to adopt the self-attention mechanism in hashtag recommendation.

The remainder of this study is organized as follows. Section 3 provides a brief introduction of the attention mechanism, self-attention mechanism, and the proposed model. Section 4 discusses our experimental dataset, experimental settings, analysis of experimental results, and parameter analysis. Section 5 summarizes the entire study and presents the conclusions drawn from this study.

3. Methodology

3.1 Attention Model

When people focus on an object, they focus not on the whole object from beginning to end but typically only on a certain part. Attention model simulates this process. In the attention model, each Key and its unique Value are the inputs, and the Query for each word is the output. When calculating the attention weights, the similarity between the Key and the Query is first computed and then normalized using a softmax function to obtain the weight of the Value that corresponds to each Key. Finally, the final representation is obtained through a weighted summation of the Value.

In general attention models, the Key and Value are typically the same due to word embedding. Inspired by Transformer [9], the scaled dot-product attention is used in the present study, thereby indicating that the similarity between Query and Key is calculated and then divided by a predefined hyperparameter $\sqrt{d_k}$ to avoid the inner product from being too large.

3.2 Self-Attention

Self-attention, also known as intra-attention, has been extensively used in recent years. For example, the self-attention mechanism is widely used in Google's recent machine translation model. In the encoder-decoder framework of machine translation tasks, the contents of source input and the target output differ. For example, in the English-Chinese machine translation, the source is an English sentence, and the target is the corresponding translated Chinese sentence. The attention mechanism ensues between each target and source pair. Self-attention refers not to the attention mechanism between a target and a source but to that between source elements or between target elements. Thus, self-attention can be regarded as the attention mechanism in the special case when a target equals a source.

Self-attention differs from the traditional attention mechanism. The traditional attention is calculated on the basis of the hidden states from the source and target. The result is the dependence between each word from the source and the target. However, self-attention differs from traditional attention, which is conducted at the source and target separately and captures the dependence between words on each side. Therefore, self-attention is more

effective than the traditional attention mechanism. This result is due to the traditional attention mechanism ignores the dependence between words in the source or target sentences. By contrast, self-attention can not only obtain the dependence between the source and target words but also effectively obtain the dependence between words in the source and target domains.

In self-attention, the condition is that Key = Value = Query. That is, the words in the sentence must be calculated once with other words in the same sentence to capture the internal structure of the sentence.

3.3 Multi-head Attention

In multi-head attention, the Query, Key, and Value are linearly transformed and fed into the scaled dot-product attention several times. Then, the attentions obtained from each time are concatenated together. Finally, after a linear transformation, multi-head attention is obtained. The linear transformation parameters of Query, Key, and Value notably differ every time. The advantage of multi-head attention is that information can be learned from different dimensions. However, multi-head attention cannot learn the position of each word in a sentence. To solve this problem, we must add an additional positional embedding, that is, the relative position of each word. Combining positional and word embedding, the embedding with location information can be encoded in the input.

3.4 Proposed Model

The task of hashtag recommendation is formulated as a multi-class classification problem. Furthermore, we will introduce the basic settings of our model, namely, Self-Attentive Neural Network (SANN), which is based on the fact that each word is represented as a low-dimensional,

continuous, and real-valued vector. This result is also known as word embedding [30,31]; all word vectors are stacked in a word embedding matrix $L_w \in R^{d_{emb} \times |V|}$, where d_{emb} is the word vector dimension, and $|V|$ is the vocabulary size. To use the semantic and grammatical associations of words well, the values of the word vectors are pre-trained from the text corpus with embedding learning algorithms [31]; given an input microblog, the embeddings in $x_i \in R^{d_{emb}}$ for each word in the text are used to obtain the first layer. Thus, a microblog post of length N is represented by a sequence of word vectors $X = (x_1, x_2, \dots, x_n)$.

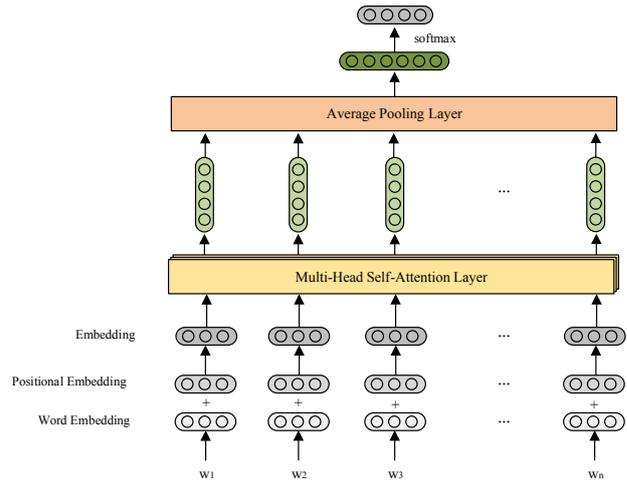


Fig. 1. Overall model of SANN.

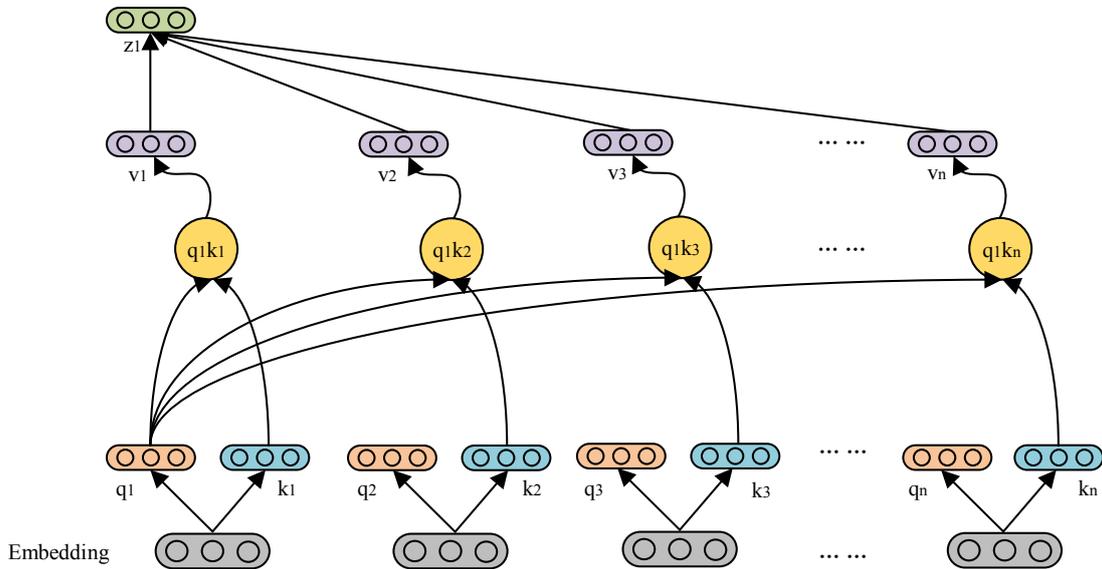


Fig. 2. Self-attention mechanism in SANN.

Our model is mainly based on self-attention and multilayer perceptron. This model can be divided into the following four steps:

(1) We use the position embedding information of each word in the post as the input, then Q, K, and V values with different parameters W are obtained to obtain the self-

attention weights of each word in the input sentence with the same dimension z_i .

(2) To represent a microblog post further, we use the attention value z_i of each word as the input, and an average pooling layer is performed to obtain Z .

(3) Then, we feed the text representation into a neural network to reduce the dimension, prevent over-fitting, and achieve the further representation of microblogs.

(4) The final output vector vec is then fed to a linear layer with an output length of the number of hashtags. A softmax layer is finally added to output the probability distribution of all candidate hashtags. The softmax function is calculated as:

$$\text{softmax}(c_i) = \frac{\exp(c_i)}{\sum_{r=1}^C \exp(c_r)} \quad (1)$$

where C is the number of hashtag categories, and c_i is the current category.

3.5 Model Training

The proposed model is trained in a supervised manner by minimizing the cross-entropy error of the hashtag classification. For tweets with more than one hashtag, each hashtag, as well as the post, is used as a training instance. The loss function is expressed as:

$$J = - \sum_{s \in S} \sum_{t \in \text{tags}(s)} \log p(t|s) \quad (2)$$

where S represents all training instances, and $\text{tags}(s)$ is the hashtag collection for microblogs.

4. Result Analysis and Discussion

The performance of the proposed method for hashtag recommendation was evaluated on a real-world dataset. Experiments were designed to answer the research question of whether the self-attention mechanism between words is useful for this task.

4.1 Dataset

The data used in this study was constructed from a large Twitter dataset from June to December, 2009, with 476,553,560 tweets [32]. The original dataset was larger than 1 TB, and a sub-dataset with 185,391,742 tweets from October to December was collected. Among them, 16,744,189 tweets included hashtags annotated by users. Considering the computational performance, we randomly select 100,000 tweets as a training set, along with 10,000 tweets as the validation and test sets, similar to previous works [24-28].

The statistics of our dataset is summarized in Table. 1.

Table 1. Statistics of the dataset. Nt(avg) is the average number of hashtags in the dataset.

| Tweets | Hashtags | Vocabulary Size | Nt(avg) |
|---------|----------|-----------------|---------|
| 120,000 | 17,987 | 238,200 | 1.357 |

4.2 Experimental Settings

4.2.1 Baseline Methods

For comparison, the following baseline methods were considered:

- NB: Hashtag recommendation was formulated as a classification task, and NB was applied to model the

posterior probability of each hashtag in a given microblog.

- LDA: We use the LDA method proposed by Krestel et al. [13] to recommend hashtags.
- TF-IDF: The BoW model with term frequency-inverse document frequency (TF-IDF) weighting was used to extract the textual feature; then, a multi-class SVM classification model [33] was used as the classifier.
- FastText: FastText is a simple and efficient text classification method [34], which considers the average of word/n-gram embeddings as document embeddings that are then fed into a linear classifier.
- CNN: A convolutional neural network (CNN) [35] was used for sentence classification. The architecture is a direct application of CNNs as used in computer vision but with 1D convolutions.
- LSTM: The hidden vector of the last position in a post from the LSTM model was used as the representation of the microblog text. On this basis, the text was classified.
- Bi-LSTM: Bidirectional LSTM (Bi-LSTM) model text semantics are from forward and backward. The forward and backward hidden states of the last position in a post were concatenated. On this basis, the text was classified.

4.2.2 Experimental Setup

In the experiments, hashtags were recommended in the following ways: the training data was first used to train our model, and the model that had the optimal performance on the validation dataset was saved. For the microblog with unlabeled data, the post was encoded through the proposed model, and then the softmax classification was performed. All neural models were trained with sentences of lengths reaching 50 words.

For each of the abovementioned models, the dimension of all the hidden states in the neural network was set to 500, and the dimension of word embeddings was 300 unless otherwise noted. A minibatch stochastic gradient descent algorithm was used together with the Adam method to train each model [36]. The hyperparameter β_1 was set to 0.9, and β_2 was set to 0.999 for optimization. The learning rate was set to 0.001. The batch size was set to 50. The network was trained for 20 epochs with early stopping.

For both our models and the baseline methods, we use the validation data to tune the hyperparameters. The results of the test data in the same hyperparameter setting were reported. Furthermore, the word embeddings used in all the methods were pre-trained from the original Twitter data released by [32] with the word2vec toolkit [31].

The hashtags annotated by users were used as the golden set. To evaluate the performance, the precision (P), recall (R), and F-score (F) were used as the evaluation metrics. Precision denotes the percentage of “tags truly assigned” among “tags assigned by system.” Recall denotes the “tags truly assigned” among “tags manually assigned.” F-score is the average of precision and recall. The same settings have been adopted by previous works [24-28].

4.3 Comparison with Other Methods

Table 2 lists the comparison results between the proposed method and the state-of-the-art discriminative and generative methods on the Twitter dataset. Notably, the hashtags with the highest probability were regarded as the result predicted using the model. In particular, only the top 1 recommendation result was evaluated.

Table 2. Evaluation results of the different methods for top 1 hashtag recommendation.

| Methods | Precision | Recall | F-score |
|----------|-----------|--------|---------|
| NB | 0.136 | 0.118 | 0.126 |
| LDA | 0.190 | 0.164 | 0.176 |
| TF-IDF | 0.248 | 0.218 | 0.232 |
| FastText | 0.269 | 0.228 | 0.247 |
| CNN | 0.319 | 0.269 | 0.292 |
| LSTM | 0.501 | 0.437 | 0.467 |
| Bi-LSTM | 0.503 | 0.440 | 0.469 |
| SANN | 0.742 | 0.642 | 0.688 |

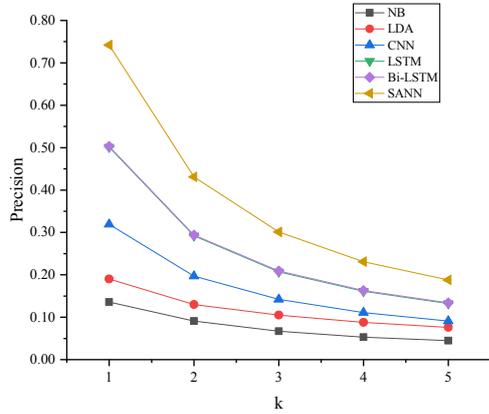


Fig. 3. Precision with recommended hashtags ranges from 1 to 5.

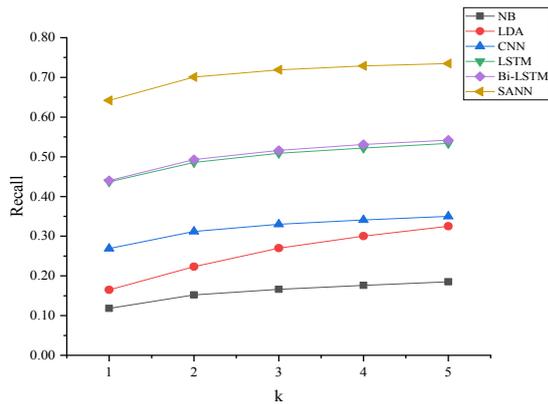


Fig. 4. Recall values with recommended hashtags ranges from 1 to 5.

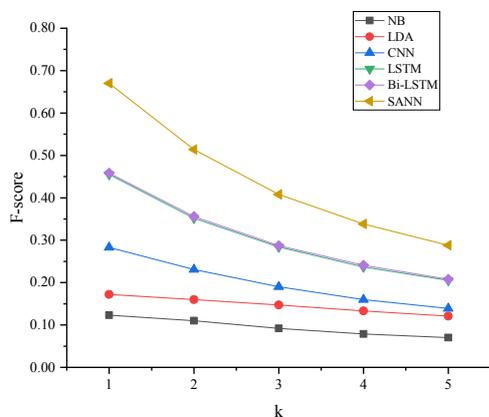


Fig. 5. F1 score with recommended hashtags ranges from 1 to 5.

To summarize, this study obtained the following conclusions:

First, for the general topic model LDA, the word-gap problem causes poor capture of the sentence topic at times

and only obtains sparse features of sentences. The LDA achieved a result that is slightly better than that of NB but worse than that of TF-IDF.

Second, the neural models achieved better experimental results than the traditional baseline methods, such as NB, TF-IDF, and LDA, thus indicating that neural networks are effective in learning the semantic information of microblog posts and could considerably improve the performance. In addition, the results of CNN and FastText were comparable, thereby demonstrating that FastText is suitable for classification tasks with numerous categories.

Third, the proposed neural network model that combines the self-attention mechanism SANN achieved the optimal results. In comparison with the CNN and LSTM, SANN improved significantly in all the evaluation metrics. The interaction between words was effective in this task.

Finally, although its precision and F-score decreased when the number of hashtags was large, SANN still outperformed the other methods. Moreover, the relative improvement was higher on extracting only one hashtag than on more than one hashtag, thus showing that recommending more than one hashtag for a microblog post is more difficult.

4.4 Parameter Analysis

The results of hashtag recommendation with different parameters in terms of the number of heads and dimensions of self-attention were compared as follows:

Table 3. Evaluation results of the different Parameters for top 1 hashtag recommendation.

| Number of Heads (h) | Dimension of Self-Attention (dim k) | F-score |
|---------------------|-------------------------------------|---------|
| 4 | 32 | 0.688 |
| 4 | 64 | 0.676 |
| 4 | 128 | 0.665 |
| 4 | 256 | 0.643 |
| 8 | 32 | 0.672 |
| 8 | 64 | 0.64 |
| 8 | 128 | 0.638 |
| 8 | 256 | 0.646 |
| 16 | 32 | 0.653 |
| 16 | 64 | 0.636 |
| 16 | 128 | 0.622 |
| 16 | 256 | 0.587 |

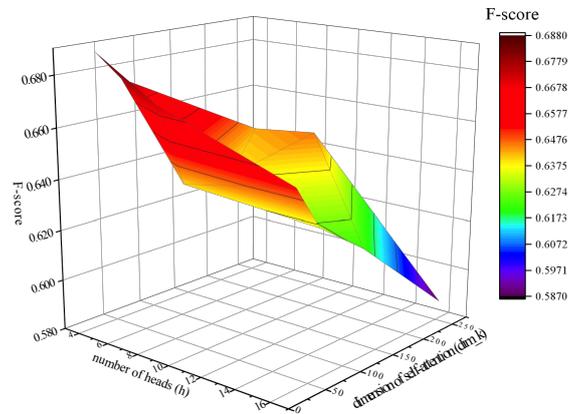


Fig. 6. Analysis of parameters: number of heads and dimension of self-attention

From the analysis of these results, increasing the dimension of self-attention would not increase the F-score when the number of heads is fixed. However, increasing the number of heads would not increase the F-score when the

dimension of self-attention is fixed. The results showed the complexity of parameters of self-attention when learning the representation of short texts, such as tweets.

5. Conclusions

To understand the content deeply and reveal the correlations of interactions between words and hashtags in microblog posts, this study proposed a deep neural network that combines self-attention mechanism and multi-layer perception. A detailed empirical analysis of hashtag recommendation on a Twitter dataset was conducted. The following conclusions were drawn:

(1) Introducing a word-level self-attention mechanism into short-text hashtag recommendation is effective.

(2) By integrating the self-attention mechanism between words, the model proposed in this study can capture the interactions between words and hashtags. Compared with the CNN and LSTM network, our model achieved a significantly improved performance in hashtag recommendation task.

The hashtags of a microblog are related not only to the text but also to the publication time and authors' occupation. Future work will aim to incorporate other external information, such as time, authors' occupation, and user relationships, into the task to improve the recommendation performance.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (NSFC) via Grant 61806049, the Natural Science Foundation of Heilongjiang Province of China via Grant F2018001 and the special funds of Central College Basic Scientific Research Bursary via Grant 2572018BH04.

This is an Open Access article distributed under the terms of the Creative Commons Attribution License



References

- Tran, V. C., Hwang, D., Nguyen, N. T., "Hashtag Recommendation Approach Based on Content and User Characteristics". *Cybernetics and Systems*, 49(5-6), 2018, pp.1-16.
- Twitter. Available online: <https://www.twitter.com> (accessed on 17 March 2019).
- Efron, M., "Hashtag retrieval in a microblogging environment". In: *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, Geneva, Switzerland: ACM, 2010, pp.787-788.
- Bandyopadhyay, A., Ghosh, K., Majumder, P., Mitra, M., "Query expansion for microblog retrieval". *International Journal of Web Science*, 1(4), 2012, pp.368-380.
- Davidov, D., Tsur, O., Rappoport, A., "Enhanced sentiment learning using twitter hashtags and smileys". In: *Proceedings of the 23rd international conference on computational linguistics: posters. Association for Computational Linguistics*, Beijing, China: China Association for Computational Linguistics, 2010, pp.241-249.
- Wang, X., Wei, F., Liu, X., Zhou, M., Zhang, M., "Topic sentiment analysis in twitter: a graph-based hashtag sentiment classification approach". In: *Proceedings of the 20th ACM international conference on Information and knowledge management*, New York, USA:ACM, 2011, pp.1031-1040.
- Li, Y., Jiang, J., Liu, T., Sun, X., "Personalized Microtopic Recommendation with Rich Information". In: *Social Media Processing: 4th National Conference, SMP 2015*, Guangzhou, China: Springer, 2015, pp.1-14.
- Hong, L., Ahmed, A., Gurumurthy, S., Smola, A.J., Tsioutsouklis, K., "Discovering Geographical Topics in the Twitter Stream". In: *Proceedings of the 21st International Conference on World Wide Web*, New York, USA: ACM, 2012, pp.769-778.
- Vaswani, A., Shazeer, N., Parmar, N., et al., "Attention Is All You Need". *Advances in neural information processing systems*, 2017, pp. 5998-6008.
- Zangerle, E., Gassler, W., Specht, G., "Recommending#-tags in twitter". In: *Proceedings of the 2nd International Workshop on Semantic Adaptive Social Web (SASWeb 2011)*, Girona, Spain: 2011, pp.67-78.
- Godin, F., Slavkovic, V., De Neve, W., Schrauwen, B., et al., "Using Topic Models for Twitter Hashtag Recommendation". In: *Proceedings of the 22nd International Conference on World Wide Web*, Rio de Janeiro, Brazil: ACM, 2013, pp.593-596.
- Mazzia, A., Juett, J., "Suggesting hashtags on twitter". Retrieved from <http://www-personal.umich.edu/~amazia/pubs/545-final.pdf>, 2011-08-15/2019-03-17.
- Krestel, R., Fankhauser, P., Nejdl, W., "Latent Dirichlet Allocation for Tag Recommendation". In: *Proceedings of the Third ACM Conference on Recommender Systems*, New York, USA: ACM, 2009, pp.61-68.
- Blei, D.M., Ng, A.Y., Jordan, M.I., "Latent Dirichlet Allocation". *Journal of machine Learning research*, 3, 2003, pp.993-1022.
- Ding, Z., Zhang, Q., Huang, X., "Automatic hashtag recommendation for microblogs using topic-specific translation model". In: *Proceedings of the 24th International Conference on Computational Linguistics (COLING)*, Mumbai, India, 2012, pp.265-274.
- Kywe, S.M., Hoang, T.A., Lim, E.P., Zhu, F., "On Recommending Hashtags in Twitter Networks". In: *International Conference on Social Informatics*, Berlin, Heidelberg: Springer-Verlag, 2012, pp.337-350.
- Wang, Y.; Qu, J.; Liu, J.; Chen, J.; Huang, Y. "What to Tag Your Microblog: Hashtag Recommendation Based on Topic Analysis and Collaborative Filtering". In: *Asia-pacific Web Conference*, Springer, Cham, 2014, pp.610-618.
- Zhao, F., Zhu, Y., Jin, H., Yang, L.T., "A personalized hashtag recommendation approach using LDA-based topic model in microblog environment". *Future Generation Computer Systems* 65(C), 2016, pp.196-206.
- Li, Q., Shah, S., Nourbakhsh, A., Liu, X., Fang, R. "Hashtag Recommendation Based on Topic Enhanced Embedding, Tweet Entity Data and Learning to Rank". In: *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, New York, USA: ACM, 2016, pp.2085-2088.
- Ma, Z., Sun, A., Yuan, Q., Cong, G., "Tagging Your Tweets: A Probabilistic Modeling of Hashtag Annotation in Twitter". In: *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, New York, USA: ACM, 2014, pp.999-1008.
- Lu, H., Lee, C. "A Twitter Hashtag Recommendation Model that Accommodates for Temporal Clustering Effects." *IEEE Intelligent Systems*, 30(3), 2015, pp.18-25.
- Kowald, D., Pujari, S.C.; Lex, E. "Temporal Effects on Hashtag Reuse in Twitter: A Cognitive-Inspired Hashtag Recommendation Approach". In: *Proceedings of the 26th International Conference on World Wide Web; International World Wide Web Conferences Steering Committee*, Republic and Canton of Geneva, Switzerland: WWW '17, 2017, pp.1401-1410.

23. Zhang, Q., Wang, J., Huang, H., Huang, X., Gong, Y. "Hashtag Recommendation for Multimodal Microblog Using Co-Attention Network". In: *Twenty-Sixth International Joint Conference on Artificial Intelligence*, Melbourne, Australia:IJCAI, 2017, pp.3240-3426.
24. Gong, Y., Zhang, Q. "Hashtag Recommendation Using Attention-Based Convolutional Neural Network". In: *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, New York, USA:IJCAI, 2016, pp.2782-2788.
25. Gong, Y., Zhang, Q., Huang, X. "Hashtag recommendation for multimodal microblog posts". *Neurocomputing*, 272, 2018, pp.170-177.
26. Li, Y., Liu, T., Jiang, J., Zhang, L. "Hashtag Recommendation with Topical Attention-Based LSTM". In: *Proceedings of the 24th International Conference on Computational Linguistics(COLING)*, Osaka, JAPAN, 2016, pp.3019-3029.
27. Li, Y., Liu, T., Hu J., Jiang J. "Topical Co-Attention Networks for hashtag recommendation on microblogs". *Neurocomputing*, 331, 2019, pp.356-365.
28. Zhu, R.; Yang, D.; Li, Y. "Learning improved Semantic Representations with Tree-Structured LSTM for Hashtag Recommendation: An Experimental Study". *Information*, 10(4), 2019.
29. Guo, M., Zhang, Y., Liu T. "Gaussian Transformer: a Lightweight Approach for Natural Language Inference". In:*The Thirty-Third AAAI Conference on Artificial Intelligence* , Hawaii, USA:AAAI-19, 2019.
30. Bengio, Y., Ducharme, R., Vincent, P., Jauvin, C. "A neural probabilistic language model". *Journal of machine learning research*, 3, 2003, pp.1137-1155.
31. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J. "Distributed Representations of Words and Phrases and their Compositionality". *Advances in Neural Information Processing Systems*, 26, 2013, pp.3111-3119.
32. Yang, J., Leskovec, J. "Patterns of Temporal Variation in Online Media". In: *Proceedings of the Fourth ACM International Conference on Web Search and Data Mining*, Hong Kong, China, 2011, pp.177-186.
33. Hearst, M.A., Dumais, S.T., Osman, E., Platt, J., Scholkopf, B. "Support vector machines". *IEEE Intelligent Systems and their applications*,13(4), 1998, pp.18-28.
34. Joulin, A., Grave, E., Bojanowski, P., Mikolov, T. "Bag of Tricks for Efficient Text Classification". Retrieved from <https://arxiv.org/abs/1607.01759>, 2016-07-06/2019-03-17.
35. Kim, Y. "Convolutional Neural Networks for Sentence Classification". In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Doha, Qatar, 2014, pp.1746-1751.
36. Kingma, D.P., Ba, J. "Adam: A Method for Stochastic Optimization". Retrieved from <https://arxiv.org/abs/1412.6980>, 2014-12-22/2019-03-17.