Tag Recommendation in Social Networks with the Help of Optimized Text Summarization and Fuzzification

Mahsa Rahimi Resketi[1], HomayunMotameni[2*], Ebrahim Akbari [3], Hossein Nematzadeh[4]

Abstract–Hashtags,i.e., terms that are prefixed by a # symbol, are vastly used in social media like Twitter, Instagram, etc. Hashtags present rich sentiment information about people's favorite topics an d would make a text more accessible and popular. This paper proposed a model of the hashtag recommendation problem using an automatic summarizer using deep neural and Fuzzy log ic system,as also some semantic text mining models. The final summarized text is based o n Restricted Boltzmann Machine (RBM),and with the help of Extreme learning machines (ELM), improves the training data, then a fuzzy rule-based method on the sentences is don e to build the final result.

The experiments on two public data sets improved that the proposed model outperforms th e related approaches and is more efficient improvement than previous methods.

Keywords: Hashtag recommendation, text summarization, fuzzy, neural network

1. Introduction

In previous years, social networks, microblogging become so popular and important [1]. While variant social networks such as Twitter, Facebook, Instagram, etc. Becoming so popular, finding ways to use them more efficiently becomes urgent. A lot of information is produced daily in these networks, and using them would be an outstanding achievement.

For fast and accurate transmission of information, social media allow users to use hashtags (e.g., #News) in their posts to declare the main context of the post. Hashtags are chosen by users and added to the post to help other people find the post, therefore it was created in the first place to help people organize their messages.

Hashtags simply help organize content and improve information diffusion in social media [1].

Hashtags have a great part in providing crucial information almost about everything on the internet.

1 Department of Computer Engineering, Islamic Azad University, Sari, Mazandaran, Iran, mr2.mco@gmail.com

2* Corresponding Author: Department of Computer Engineering, Islamic Azad University, Sari, Mazandaran, Iran, * h_motameni@yahoo.com

3 Department of Computer Engineering, Islamic Azad University, Sari, Mazandaran, Iran, akbari@iausari.ac.ir

4 Department of Computer Engineering, Islamic Azad University, Sari, Mazandaran, Iran, hn_61@yahoo.com

Received: 2023.01.16; Accepted: 2023.03.09

The hashtag helps users to send messages with specific content, without the need to use any specific background work [2]. Hashtags could be applied in lots of different applications, such as the expansion of query [3], analysis of semantics[4], or mining of tweets[5].

However, while all hashtags should be manually annotated since now only 14.6% of tweets contain hashtags [4]. Therefore, the best way to improve the present situation would be a reliable hashtag recommendation method. For that reason, various approaches have been presented to recommend hashtags automatically, such as [6]-[10]. These methods usually use two different attempts which either utilize collaborative filtering or machine learning classifiers. All of these methods share a similar weakness which is, ignoring the semantic information in the document. Tomar et al. approach to recommend the tags was a feed-forward neural network and also utilizing word embeddings [11]. Their proposed method recommends hashtags based on the semantics of tweets. Recurrent neural networks (RNNs) proved a great performance in the representation learning field[12].

In this paper, by combining the neural network model and fuzzy system and with the help of semantic text summarization methods recommend hashtags. This method tries to reach the best hashtags with the use of semantic text summarization methods.

Sufficient experiments redone in this paper to evaluate the proposed method and compare it with other approaches.

The rest of the paper is ordered as follows. Related work is briefly reviewed in Section 2. In Section 3, the detail of the proposed approach is shown.

Section 4 presents the results and comparisons. And the last parts are conclusions and thoughts for future work which are presented in Section 5.

2. Related work

Hashtags become so popular in the previous years and have been vastly used in different applications, such as prediction of popularity [13], search [14], and detection of events [15].

It has been proved that manually labeled hashtags for marking topics and putting emphasis on a content are applicable in various fields, including sentiment analysis [16], and content recommendation [17]-[18], and for training vision models have used as manual supervision [19].

Regarding the application of recommendation [20]-[23], previous research on tag recommendation is classified into three different groups, namely probabilistic neural network-based, graph-based, and a mixture of these two.

Lu and Lee proposed a hashtag recommendation model that get the temporal clustering effect of latent topics in tweets [24]. Kowald et al. tried to use the time effect of the hashtag to create a tag recommendation algorithm named Base-Level Learning (BLL)[25]. At the same time, the neural networks proved to be superior in hashtag recommendation methods. Denton *et al* applied a convolutional neural network to define various features from images and use the meta data to achieve the best hashtag recommendation result [26]. Wang *et al.* proposed annotating hashtags with a new sequence generation framework, to do so it had viewed hashtags as short sequences of words [27]. And grouped the GCN and LSTMs to reach the best performance for hashtag recommendation [28].

Liet al. proposed a recurrent neural network model for tweet hashtag recommendation [29]. Wang et al. attempt to use deep item representations for the hashtag recommendation, and meanwhile perform deep representation and relational learning in a principled way [30]. It became clear that hashtags declare the fundamental subjects of social media posts, and the attention-based LSTM model in [31] include the topic modeling into the LSTM architecture.

There isa lot of other researchers working on hashtags and finding ways to enhance the recommendation system.

3. Proposed method

This paper proposes a novel method that is a combination of a deep neural network and fuzzy system, attempting to find the features of the document and improve correlation and significance, to define the essential keywords and compute the final summary. The main idea was inspired by what Cheopade and Narvekar proposed [32]. Fig. 1 represents the framework of the proposed model.

3.1 Features Extraction

The input text document is inserted into the automatic summarizing system. At first, it extracts the features of the document.

Three different features are applied, to extract important keywords of sentences.

These features are proposed as:

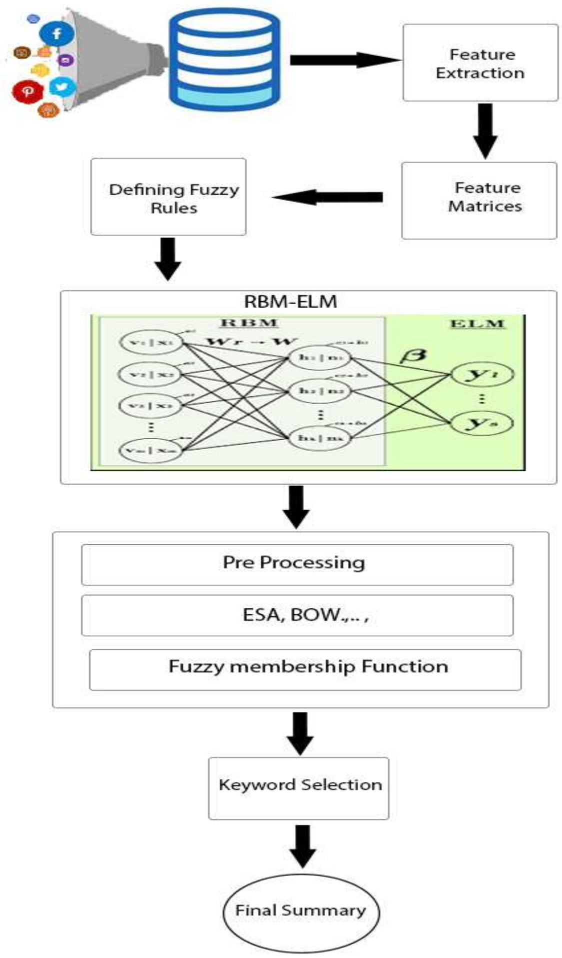

Fig. 1. Theframework of the proposed Auto Text Summarizer

1. **Emoticon** is the number of different emoticon

symbols that are used in the text document.

$F_1 = \sum E$ (1)

Where,

E= The count of Emoticon in each document.

2. **Term weight-Sentence**: this feature would help to define the most frequent words in each document

$F_2 = tf(t,d)\times\ idf(t,D)$ (2)

Where,

tf = frequency of the word in each sentence,

idf = inverse document frequency that is the word being frequentor rare in a document[33],

t= term,

d= each document,

D= a set of documents.

3. **Named entities- Sentence**, it is so obvious that named entities are usually the main factor of a sentence and they would declare the keywords or main idea of a document.

F_3 = it indicates the number of named entities in the document.

At last, with all of these features, a sentencing matrix is built.

3.2 Defining Fuzzy Rules

Once the mentioned matrix is built. Three fuzzy rules are formed and performed on the whole document. The feature matrix results for a sentence are named FR so:

Rule 1. (If FR is high) then (the sentence is important.)

Rule 2. (If FR is medium) then (the sentence is average.)

Rule 3. (If FR is low) then (the sentences is unimportant.)

3.3 Restricted Boltzman machine (RBM)

These sentences are then given to the RBM, then with a help of training data, they are. In the proposed method just one input and output layer, and two hidden layers. The input text is inserted initially into the first layer, then the score of the sentence with a bias value is fed to the first hidden layer. The whole step is done again for the next layer. Then result is inserted to the last layer (output), this would end in a better extraction of sentences.

In an RBM that contains n visible and m hidden layers, the joint probability distribution of P(v,h) is obtained by:

$$P(v,h) = \frac{1}{z}\exp\left(-E(v,h)\right) \quad (3)$$

where v represents the visible layers that, $v\in \{0,1\}^n$, and h represents the hidden layers, $h\ \in \{0,1\}^m$, and at last E(v,h) is calculated by

$$E(v,h) = -\sum_{i=1}^{n} a_i v_i - \sum_{j=1}^{m} b_j h_j - \sum_{i=1}^{n}\sum_{j=1}^{m} v_i w_{ij} h_j \quad (4)$$

To normalize thefinal results Z given by

$$z = \sum_{v,h} \exp\left(-E(v,h)\right) \quad (5)$$

In equation (4), bi and a_i are values of the hidden and visible layers, respectively. The w_{ij} is weighted among hidden and visible layers.

In equation (6) the activation probabilities of the hidden layers are computed.

$$P\left(h_j = 1\middle|v\right) = \sigma(b_j + \sum_{i=1}^{n} v_i w_{ij}) \quad (6)$$

where σ is the sigmoid function and j starts from 1 to m. Then for the visible layers, the activation probabilities are given by

$$P(v_i = 1|h) = \sigma(a_i + \sum_{j=1}^{m} h_j w_{ij} \quad (7)$$

Algorithm 1 explains how the RBM is built.

(Algorithm-1): RBM

Procedure RBM
Initial bias vectors b and a,the matrix W, and momentum v.
Initiate the states of visible unit v_1
While i< Maximum-I
  For j=1:2
    Compute $P(h_{1j}=1|v_1)$ using Gibbs Sampling
  $h_{1j}\in\{0,1\}$ from $P(h_{1j}|v_1)$
  For j=1:2
    Compute $P(V_{2i}=1|h_1)$ using Gibbs Sampling
    $V_{2i}\in\{0,1\}$ from $P(V_{2i}|h_1)$
  For j=1:2
    Compute $P(h_{2j}=1|v_2)$ using Gibbs Sampling
  $W= W +\ \in (P(h_1=1|v_1)v_1^T - P(h_2=1|v_2)v_2^T)$
  $a=a+\in (v_1-v_2)$
  $b=b+\in (P(h_1=1|v_1)-P(h_2=1|v_2))$
x= Updating of momentum

3.4Extreme learning machines (ELM)

While RBM results could be more efficient using a neural network learning method, ELM is used in t a his step. ELM is neural network used in different contexts such as clustering classification, and feature learning with one layer or even multiple layers of hidden nodes. The hidden layers are random projections but with nonlinear transforms. Most of the time, the output weights of hidden layers are found and trained in just one step.

RBM can produce the result more accurately and the rate of its learning is so faster than any other network. It focused on an undirected network G(vect,δ), where vect presents a collection of N vertices numbering and δ defines edges with total number M. Whit out considering the starts and ends edges, connections among nodes are given by adjacency matrix which is

$$A \in \mathbb{R}^{N\times N}: A_{ij} = \begin{cases} 1, if\ vertex\ i\ and\ j\ are\ connected, \\ 0, \quad otherwise, \end{cases} \tag{8}$$

where I and j are from 1 to N.

In the next step, ELM is calculated in two different steps. In the first step, the input weights are executed based on a uniform distribution. Then in the next step, it is expressed as the inner product.

$$E = H\beta \tag{9}$$

Where

$$H = \begin{bmatrix} g(w_1^T a_1 + b_1) & \dots & g(w_{N_h}^T a_1 + b_{N_h}) \\ \vdots & \dots & \vdots \\ g(w_1^T a_N + b_1) & \dots & g(w_{N_h}^T a_N + b_{N_h}) \end{bmatrix}$$

$$\beta = \begin{bmatrix} \beta_1^T \\ \vdots \\ \beta_{N_h}^T \end{bmatrix} \epsilon \mathbb{R}^{N_h \times n} \tag{10}$$

The additional constraint inspired by Belkin and Niyogi is the degeneration of the solution [36], and I_n is an identity matrix (n-dimensional).

Let v and γrepresents the corresponding eigenvector and the eigenvalue, respectively. When $M \geq M_h$, H is calculated solving (10).

$$\left(I_{N_h} + \lambda H^T L H\right) v = \gamma H^T H v \tag{11}$$

By the first n + 1smallest eigenvalues $\gamma_1, \gamma_2, \dots and$ their eigenvectors $v_1, v_2, \dots, v_{n+1}$,andβmeans the output weights is defined:

$$\beta = [\tilde{v}_2, \tilde{v}_3, \dots, \tilde{v}_{n+1}] \tag{12}$$

In this formula,$\tilde{v}_i = v_i/||Hv_I||$, i=2,.….,n+1,defines the normalized eigenvectors.

If $M<M_h$, then the dimension of A is just less than the number of neurons in the hidden layer, and the other formula for N is

$$(I_n + \lambda L H H^T) v = \gamma H H^T v \tag{13}$$

So, for β

$$\beta = H^T[\tilde{v}_2, \tilde{v}_3, \dots, \tilde{v}_{n+1}] \tag{14}$$

where in this step the number of vectors are $\tilde{v}_i = v_i/||HH^T v_i||$, i = 2, ... , n + 1.

ELM is shown in Algorithm 2.

3.5 Pre-Processing

In this step document is preprocessed to be ready for extracting sentences based on a particular subject. It is obvious the accurate summarizing is the first goal and so preprocessing would help remove frequent useless steps. This step includes filtering of the stop word, stemming, and Parts of speech tagging.

1. Filtering of stop word, in this step the words like “a”, “an”, “and”, “the”, semicolon, etc., are omitted,

(Algorithm-2): ELM

```
Procedure ELM (A)
Initial consts   Mh, m,λ
BuiltN from A
Built input biases and weights of neurons in
hidden layers, compute H
Normalize A
If M ≥ Mh
Find the 1stm+1 smallest eigenvalues like
Equation 11 and defineβbyEquation 12
Else
Detect the 1stm+1 smallest eigenvalues as in
Equation 13 and create β by Equation 14
Define all the embedding matrix by E=Hβ
Select centroids by defining k rows randomly
from E (c1,c2,…,ck)
While tis not changing
      For i=1: M
       For j=1: k
      Disj=||ei-cj||
      Define the smallest amount in dis and s
      ti=s
      For I =1: k
         Update ci as the mean value of cluster i
```

while they are not as important as other words.

2. Stemming, which defines the root of words by omitting suffixes such as “ed”, “s”,etc. from a word.

3. Parts of speech tagging that classify a word to the exact category it belongs to such as verb, noun, adjective, or adverb.

The Stop Word Remover is shown in Algorithm 3.

(Algorithm-3):Stop Word Remover

Procedure Stop Word Remover
n=size(raw_text);
Fori=1:n
If ismember (raw_text[i] ,StopWords)
remove raw_text[i]

3.6 BOW

The BOW is a model that counts the number of repetitions of words in each document. In this theory, the input data is presented as a vector $\vec{d}$ of the weight of words, that is, in each document $d_i \in D$, $d_i$is a vector with high dimensions of weight and each dimension represents a specific word. [34].

T[i,j] is in fact the tf-idf value of the word t_i in the document d_j.

$$T[i,j] = tf\,(t_i, d_j) \times \log \frac{n}{df_i} \tag{15}$$

In which the amount of word repetition is equal to:

$$tf\,(t_i, d_j) = \begin{cases} 1 + \log count\,(t_i, d_j), & count\,(t_i, d_j) > 0 \\ 0 & otherwise \end{cases} \tag{16}$$

$df_i=\{d_k : t_i \in d_k\}$ is the number of various documents that contain the word t_i (document repetition rate).

According to Algorithm 4, BoW is calculated for all texts. This algorithm determines the repetition of each word (verbs and words) and selects words with much higher frequency to create a new label.

(Algorithm-4): BoW

Procedure BoW
For each role[i]
[n m]=size(role[i])
For j=1:n
For t=1:m
Calculate the term frequency
Create a vector of frequency for this word

3.7 Explicit Semantic Analysis (ESA)

In this step initially, the semantic connection between the words is found. With the help of this semantic connection, one can easily extract important keywords and understand the content of the text. As a result of this operation, a series of related words are obtained, all placed in one group. Groups are formed to continue the search based on them.

- Weight of expressions

Although there are different weighting methods for indexing, they all have the following two things in common:

•The more times a phrase occurs in a document that belongs to a category, the more related that phrase is to the mentioned category.

•The more the phrase is repeated in different documents that represent different categories, the less suitable this phrase is for distinguishing between the various existing categories.

All the documents must have a similar set of features. The features that were not available in some documents have been considered zero weight. In the test phase, all the documents were assumed to have the same set of features, and the distance between the features was measured for the features in the mentioned set.

- **Constructing a semantic interpreter**

If the notion N_1,…,Nm and a group of documents d_1,..., d_n are presented as input, and a table T is built from them, in which each of the m columns will be related to a notion, and rows are related to a word that happens in the document, T[i,j] in the table is a tf-idf value of the word t_i from the d_j document. At last, cosine value (normalized) is calculated for rows to eliminate the difference between the length of the documents:

$$T[i,j] \longleftarrow \frac{T[i,j]}{\sqrt{\sum_{l=1}^{r} T[i,j]^2}} \quad (17)$$

Which r represents the number of words. The semantic defines of the word t_i is in line i of table T. This means that the semantic of a word is associated with its tf-idf value through the concept vector, which indicates the degree of relevance of each notion to a word. In this step, key tags are selected.This would lead to a scoring table which is used in the fuzzy step.

3.8 Table instruction

Once the last step is done, the root word should insert in the next step. A root table input is the root word and the random priority value of it. This value creates a fuzzy membership computation of the root query.

A sentencing table is created for whole the document, this table contains all the words of the document, the number of words in the sentence, etc. At last ,the fuzzy membership score is computed.

A frequency table is built for the entire document.

All information which is provided from previous steps would form a table that would help rank the sentences

3.9 Fuzzy membership

After creating the tables, the fuzzy membership value is computed by :

$$Rank(S_i) = \sum_{i=1}^{n} frequency(w_{ij}) + \sum_{i=1}^{n} rootmembership(w_{ij}) \quad (18)$$

Where,

n= all of sentences in the sentencing table.

S_i=ith sentence.

W_{ij}=jth word of ith sentence.

Frequency() the output is the repetition of arguments

root membership() the output is the amount of the membership with the root which is given by the user.

Finally, sentences are ranked to find the final summary of the document.

With the help of previous steps (especially Bow+ESA), the main keywords of each sentence are chosen to represent the final tags. Fig. 1 presents the framework of auto summarizer.

4. Evaluation

Experimental results obtained from comparative research of several recommendation methods are presented in this step. A full study is performed to compare the performance of different models with the proposed method.

4.1 Dataset

To analyze and evaluate the proposed solution, it was examined on two public datasets, datasets TPA [35] and AG [36].

The TPA dataset contains 1846 scientific papers and has five different types of tags. The AG dataset contains 127,600 news studies, and four kinds of different tags are used. 75% of the data in each dataset were analyzed as training data and 25% as test data.

4.2 Quantitative assessment

The F-measure is widely used in various articles to evaluate efficiency [37]-[40]. To calculate the F-measure, first, the accuracy and coverage values are computed. In this case, recall and precision are calculated with the following formulas:

$$\text{Pr}ecision = \frac{TP}{TP+FP} \quad (19)$$

$$\text{Re}call = \frac{TP}{TP+FN} \quad (20)$$

Where TP stands for True Positive, FP is False Positive, and FN is False Negative. Based on these values, the macro value of both is calculated according to the following formulas, and the required macro-F-measure is obtained from that.

$$macro-P = \frac{١}{n}\sum_{i=١}^{n} P_i \quad (21)$$

$$macro-R = \frac{١}{n}\sum_{i=١}^{n} R_i \quad (22)$$

Therefore, the macro-F-measure is obtained from the following formula:

$$macro-F_{١} = \frac{٢ \times macro-P \times macro-R}{macro-P+macro-R} \times \%١٠٠ \quad (23)$$

In these formulas, n declares the total number of groups.

4.3 Implementation details

The proposed method was built with MATLAB R2021a on a computer with Intel(R) Core (TM) i7-10750H CPU @ 2.60GHz 2.59 GHz, 16.0 Gigabyte of RAM, and a display adapter from NVIDIA GeForce GTX 1650 Ti.

4.4 Results and data analysis

The proposed method compared with 12 similar studies. The presented methods are generally in 4 groups of memory machines, convolutional neural network (CNN), long-short-term memory (LSTM), and capsule model. By comparing the proposed method, it has been determined that this method showed highest level of accuracy, coverage, and macro-f1. Common statistical machine methods such as Adaptive Boosting (AdaBoost) and Random Forest perform poorly because they perform so weak. The performance of this method is widely dependent on features that are time-consuming to work with. The Gradient Boosting Decision Tree (GBDT) method has much better performance than the AdaBoost and Random Forest because GBDT builds trees one after the other, where each new tree helps omit the errors. Deep learning approaches performed much better than conventional machine learning approaches because, in these methods, there is no need to manage and engineer features. Moreover, attention-based BiLSTM and attention-based Convolutional Neural Networks(ABCNN) consistently outperform CNN and LSTM methods because they use an attention-based mechanism.

These studies extract major information from the input text by monitoring the label information. This confirms the effectiveness of a mixed method that proves the gaps in the previous tag recommendation method. By using the text summarization method, the proposed solution solved the problems in the previous methods and thus achieve a better result and a higher Macro-f1.

(Table-1): Comparing efficiency of the proposed method against other popular studies

Method	TPA			AG		
	Macro-P	Macro-R	Macro-F1	Macro-P	Macro-R	Macro-F1
AdaBoost[41]	0.751	0.721	0.731	0.799	0.780	0.799
RF[42]	0.744	0.725	0.732	0.769	0.769	0.768
GBDT[43]	0.811	0.789	0.797	0.820	0.821	0.820
LSTM[44]	0.805	0.797	0.798	0.861	0.862	0.861
BiLSTM[1][45]	0.815	0.811	0.810	0.882	0.880	0.881
Att-BiLSTM[46]	0.819	0.811	0.812	0.891	0.890	0.890
CNN[47]	0.804	0.798	0.800	0.914	0.908	0.911
ABCNN[48]	0.817	0.813	0.811	0.917	0.913	0.914
VD-CNN[2][49]	0.813	0.813	0.809	0.913	0.910	0.912
CapsNet[3][50]	0.820	0.815	0.814	0.921	0.918	0.920
Capsule-B[51]	0.818	0.806	0.810	0.926	0.919	0.917
ACN[52]	0.829	0.825	0.824	0.926	0.922	0.923
Proposed method	**0.829**	**0.830**	**0.829**	**0.960**	**0.955**	**0.957**

[1]Bidirectional LSTM
[2]Very Deep Convolutional Network
[3]Capsule Network

5. Conclusion

Due to the great attention paid to social networks in today's world, more than ever to hashtags for organizing, raising the fact that in today's world, tags are widely used in organizing and searching. Among the huge amount of data available, creating tags automatically has been highly considered. Tag recommendation from a text source can be considered a text extraction problem. In this paper, an attempt was made to create a series of suggested keywords from the data with the help of text summarization methods. A mixture of fuzzy system and deep neural network is proposed. Here the training of sentences over a set of data and applying rules based on them make the result humanized. The proposed solution was examined on two public datasets, and the final results proved its superiority over other similar methods.

The results indicate 83% of macro-measure for the TPA dataset, and 96% for the AG dataset, which are superior to other methods and have shown higher efficiency.

Future work will explore clustering or classification text summarization methods to enhance the final results.

References

[1] Li, Yang & Liu, Ting & Hu, Jingwen., " Topical Co-Attention Networks for Hashtag Recommendation on Microblogs," *Neurocomputing*, Vol. 28, February 2019, pp 356-365.

[2] Belhadi, Asma & Djenouri, Youcef & Lin, Chun-Wei & Cano, Alberto, " A Data-Driven Approach for Twitter Hashtag Recommendation," *IEEE Access*, vol. 8, pp. 79182-79191, 2020.

[3] K. Massoudi, M. Tsagkias, M. De Rijke, and W. Weerkamp, ``Incorporating query expansion and quality indicators in searching microblog posts," *in Proceedings ofEur. Conf. Inf. Retr.,* 2011, pp. 362-367.

[4] X. Wang, F. Wei, X. Liu, M. Zhou, and M. Zhang, ``Topic sentiment analysis in Twitter: A graph-based hashtag sentiment classification approach," *in Proceedings of20th ACM Int. Conf. Inf. Knowl. Manage*., 2011, pp. 1031-1040.

[5] Y. Wang, J. Liu, J. Qu, Y. Huang, J. Chen, and X. Feng, ``Hashtag graph-based topic model for tweet mining," in Proc. *IEEE Int. Conf. Data Mining*, Dec. 2014, pp. 1025-1030.

[6] A. Mazzia and J. Juett, "Suggesting hashtags on Twitter," *EECS 545m, Machine Learning, Computer Science and Engineering*, University of Michigan, 2009.

[7] E. Zangerle, W. Gassler, and G. Specht, "Recommending#-tags in Twitter," *in Proceedings of the Workshop on Semantic Adaptive Social Web (SASWeb 2011). CEUR Workshop Proceedings*, Vol. 730, 2011, pp.67–78.

[8] S. M. Kywe, T.-A. Hoang, E.-P. Lim, and F. Zhu, "On recommending hashtags in Twitter networks," *in Social Informatics. Springer*, 2012, pp. 337–350.

[9] S. Sedhai and A. Sun, "Hashtag recommendation for hyperlinked tweets," *in Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval. ACM*, 2014, pp.831–834.

[10] E. Otsuka, S. A. Wallace, and D. Chiu, "Design and evaluation of a Twitter hashtag recommendation system," *in Proceedings of the 18th International Database Engineering & Applications Symposium. ACM*, 2014, pp. 330–333.

[11] A. Tomar, F. Godin, B. Vandersmissen, W. De Neve, and R. Van de Walle, "Towards Twitter hashtag recommendation using distributed word representations and a deep feed forward neural network," *in Proceedings ofAdvances in Computing, Communications and Informatics (ICACCI, 2014 International Conference* on. IEEE, 2014, pp. 362–368.

[12] J. Li, H. Xu, X. He, J. Deng and X. Sun, "Tweet modeling with LSTM recurrent neural networks for hashtag recommendation," *2016 International Joint Conference on Neural Networks (IJCNN)*, Vancouver, BC, Canada, 2016, pp. 1570-1577.

[13] B. Samanta, A. De and N. Ganguly, "STRM: A sister tweet reinforcement process for modeling hashtag popularity," *IEEE INFOCOM 2017 - IEEE Conference on Computer Communications*, Atlanta, GA, USA, 2017, pp. 1-9.

[14] Yuqi Gao, Jitao Sang, Tongwei Ren, and Changsheng Xu., " Hashtag-centric Immersive Search on Social Media," *In Proceedings of the 25th ACM international conference on Multimedia (MM '17).* Association for Computing Machinery, New York, NY, USA, 2017, 1924–1932.

[15] C. Xing, Y. Wang, J. Liu, Y. Huang, W.-Y. Ma, "Hashtag-based sub-event discovery using mutually generative lda in twitter," *in: Proceedings of the AAAI Conference on Artificial Intelligence, AAAI Press*, 2016, pp. 2666–2672.

[16] Amin Javari, Zhankui He, Zijie Huang, Raj Jeetu, and

Kevin Chen-Chuan Chang, “Weakly Supervised Attention for Hashtag Recommendation using Graph Data,” *In Proceedings of The Web Conference 2020 (WWW '20). Association for Computing Machinery*, New York, NY, USA, 2020, pp 1038–1048.

[17] K. Chen, T. Chen, G. Zheng, O. Jin, E. Yao, and Y. Yu, ``Collaborative personalized tweet recommendation," *in Proceedings of 35th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr. (SIGIR)*, Portland, OR, USA, Aug. 2012, pp. 661_670.

[18] I. Ronen, I. Guy, E. Kravi, and M. Barnea, ``Recommending social media content to community owners," *in Proceedings of 37th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr. (SIGIR)*, Gold Coast, QLD, Australia, Jul. 2014, pp. 243_252.

[19] A. Veit, M. Nickel, S. Belongie, and L. van der Maaten, ``Separating self-expression and visual content in hashtag supervision," *in Proceedings of IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Salt Lake City, UT, USA, Jun. 2018, pp. 5919_5927.

[20] Mingsong Mao, Jie Lu, Guangquan Zhang, Jinlong Zhang, “Multirelational social recommendations via multigraph ranking,” *IEEE Trans. Cybern*. Vol. 47, No. 12, 2016, pp 4049–4061.

[21] Qian Zhang, Dianshuang Wu, Jie Lu, Feng Liu, Guangquan Zhang, “A cross-domain recommender system with consistent information transfer”, *Decis. Support Syst*. Vol. 104, 2017, pp 49–63.

[22] Jing Li, Pengjie Ren, Zhumin Chen, Zhaochun Ren, Tao Lian, Jun Ma, “Neural attentive session-based recommendation,” *in: Proceedings of the ACM Conference on Information and Knowledge Management*, *ACM*, 2017, pp. 1419–1428.

[23] Zhu Sun, Jie Yang, Jie Zhang, Alessandro Bozzon, Long-Kai Huang, and Chi Xu. “Recurrent knowledge graph embedding for effective recommendation,” *In Proceedings of the 12th ACM Conference on Recommender Systems (RecSys '18)*. Association for Computing Machinery, New York, NY, USA, 2018, pp 297–305.

[24] H.-M. Lu and C.-H. Lee, ``A Twitter hashtag recommendation model that accommodates for temporal clustering effects," *IEEE Intell. Syst*., Vol. 30, No. 3, May 2015, pp. 18_25.

[25] D. Kowald, S. C. Pujari, and E. Lex, ``Temporal effects on hashtag reuse in Twitter: A cognitive-inspired hashtag recommendation approach," *in Proceedings of 26th Int. Conf. World Wide Web*, Perth, WA, Australia, Apr. 2017, pp. 1401_1410.

[26] E. Denton, J. Weston, M. Paluri, L. Bourdev, and R. Fergus, ``User conditional hashtag prediction for images," *in Proceedings of 21th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD),* Sydney, NSW, Australia, Aug. 2015, pp. 1731_1740.

[27] Y. Wang, J. Li, I. King, M. R. Lyu, and S. Shi, ``Microblog hashtag generation via encoding conversation contexts," *in Proceedings of Conf. North, Minneapolis*, MN, USA, Jun. 2019, pp. 1624_1633.

[28] M. Li, T. Gan, M. Liu, Z. Cheng, J. Yin, and L. Nie, ``Long-tail hashtag recommendation for micro-videos with graph convolutional network," *in Proceedings of 28th ACM Int. Conf. Inf. Knowl. Manage. (CIKM*), Beijing, China, Nov. 2019, pp. 509_518.

[29] J. Li, H. Xu, X. He, J. Deng, X. Sun,” Tweet modeling with LSTM recurrent neural networks for hashtag recommendation,” *in: Proceedings of the International Joint Conference on Neural Networks*, *IEEE*, 2016, pp. 1570–1577.

[30] H. Wang, X. Shi, D.Y. Yeung, “Relational stacked denoising autoencoder for tag recommendation,” *in: Proceedings of the AAAI Conference on Artificial Intelligence, AAAI Press*, 2015, pp. 3052–3058.

[31] Y. Li, T. Liu, J. Jiang, L. Zhang, “Hashtag recommendation with topical attention-based LSTM,” *in: Proceedings of the International Conference on Computational Linguistics, ACM,* 2016, pp. 3019–3029.

[32] H. A. Chopade and M. Narvekar, "Hybrid auto text summarization using deep neural network and fuzzy logic system," *2017 International Conference on Inventive Computing and Informatics (ICICI)*, Coimbatore, India, 2017, pp. 52-56.

[33] S. Karwa and N. Chatterjee, "Discrete Differential Evolution for Text Summarization," *2014 International Conference on Information Technology*, Bhubaneswar, India, 2014, pp. 129-133,

[34] Mahdi, A.E., A. Alahmadi, and A. Joorabchi, "Combining Bag-of-Words and Bag-of-Concepts Representationsfor Arabic Text Classification," *in 25th IET Irish Signals & Systems Conference 2014 and 2014 China-Ireland International Conference*

on Information and Communities Technologies (ISSC 2014/CIICT 2014), pp. 343-348, 2014.

[35] Tang, J., et al., "Cross-domain collaboration recommendation," *in Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining, Association for Computing Machinery*: Beijing, China, pp. 1285–1293, 2012.

[36] Zhang, X., J. Zhao, and Y. "LeCun, Character-level convolutional networks for text classification," *in Proceedings of the 28th International Conference on Neural Information Processing Systems MIT Press*: Montreal, Canada., Vol. 1, pp. 649–657, 2015.

[37] Li, P., C. Tang, and X. Xu, "Video summarization with a graph convolutional attention network," Frontiers of Information Technology & Electronic Engineering, Vol. 22, pp. 902-913, 2021.

[38] Zhang, K., et al. "Video Summarization with Long Short-Term Memory," *in Computer Vision – ECCV* 2016,. Cham: Springer International Publishing, pp.766-782, 2016.

[39] ji, Z., et al., Video Summarization With Attention-Based Encoder–Decoder Networks. *IEEE Transactions on Circuits and Systems for Video Technology*, 2017, PP 1709-1717.

[40] Zhao, B., X. Li, and X. Lu " ,Property-Constrained Dual Learning for Video Summarization," *IEEE Transactions on Neural Networks and Learning Systems*, Vol. 31, pp. 3989-4000, 2020.

[41] Zhang, P. and Z. Yang, "A Novel AdaBoost Framework With Robust Threshold and Structural Optimization", *IEEE Transactions on Cybernetics*, Vol. **48,** pp. 64-76, 2018.

[42] Wang, Y., et al., "Bernoulli random forests: closing the gap between theoretical consistency and empirical soundness," *in Proceedings of the Twenty-Fifth International Joint Conference on ArtificialIntelligence, AAAI Press*: New York, New York, USA, pp. 2167–2173, 2016.

[43] Friedman, J.H., "Stochastic gradient boosting," *Computational Statistics & Data Analysis*, Vol. **38,** pp. 367-378, 2002.

[44] Cho, K., et al., "Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation," *in proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1724–1734, 2014.

[45] Zhang, S., et al. "Bidirectional Long Short-Term Memory Networks for Relation Classification,". *in PACLIC*, pp. 73-78, 2015.

[46] Zhou, P., et al., "Attention-Based Bidirectional Long Short-Term Memory Networks for Relation Classification,"*Journal of Biomedical Informatics*, pp. 207-212, 2016.

[47] Kim, Y., "Convolutional Neural Networks for Sentence Classification," *in proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pp. 1746–1751, 2014.

[48] Yin, W., et al., "ABCNN: Attention-Based Convolutional Neural Network for Modeling Sentence Pairs," *Transactions of the Association for Computational Linguistics*, Vol. 4, pp. 259-272, 2015.

[49] Schwenk, H., et al. "Very Deep Convolutional Networks for Text Classification," in *EACL*, pp. 1107–1116, 2017.

[50] Sabour, S., N. Frosst, andG.E. Hinton, "Dynamic routing between capsules," *in Proceedings of the 31st International Conference on Neural Information Processing Systems*, Curran Associates Inc.: Long Beach, California, USA, pp. 3859–3869, 2017.

[51] Yang, M., et al., "Investigating Capsule Networks with Dynamic Routing for Text Classification," pp. 3110-3119, 2018.

[52] Lei, K., et al., "Tag recommendation by text classification with attention-based capsule network," *Neurocomputing*, Vol. **391**, pp. 65-73, 2020.