Data Analytics and Artificial Intelligence
Vol: 2(5), 2022
REST Publisher; ISBN: 978-81-948459-4-2
Website: http://restpublisher.com/book-series/data-analytics-and-artificial-intelligence/

Hashtag Recommendation for twitterusingLSTM-RNN

***M. Uma maheshwari,M. Suguna**
St. Joseph,s College of Arts And Science for Women, Hosur, Tamilnadu, India.
*Corresponding Author Email:umamahe996@gmail.com

Abstract. Abstract -The hashtag recommendation problem is addressed in this paper using high average-utility pattern mining. We present (Pattern Mining for Hashtag Recommendations). There are two main stages to it. In the beginning, offline processing converts the corpus of tweets into a transactional database taking into account the temporal information of the tagged tweets (tweets with hashtags). The technique identifies the top k high average utility temporal patterns. Offline construction is also done for irrelevant tags and the ontology of tags. Second, to extract the most pertinent hashtags for a specific orpheline tweet, an online processing inputs the utility patterns, the ontology, and the irrelevant tagged tweets (tweets without hashtags). A hashtag, also known as a hash symbol, is used to indicate a keyword or topic in a tweet. It was developed organically by users as a means of categorizing messages. Hashtags are also useful for many research applications such as sentiment classification and topic analysis. Only a small percentage of tweets are manually annotated. As a result, an automatic hashtag recommendation method is required to assist users in tagging their new tweets. This same reality that these initiatives all use the High-frequency sub-strategy to symbolize posts on Twitter and ignore semantic information in posts on Twitter is a bottleneck. We consider hashtag recommendation to be a classification task in this article, but we propose a novel, recurrent neural network model, recurrent neural network model to-factor-based tweet representations to recommend hashtags. The sentence vectors are then used to train a long short-term memory recurrent neural network (LSTM-RNN). With no feature extraction, designers use the derived tweet feature vector as characteristics to classify hashtags. Experiments on real-world Tweets to recommend hashtags show that our proposed LSTM-RNN model outperforms state-of-the-art methods, and the LSTM unit also outshines standard RNN and gated recurrent unit (GRU)
Keywords: Hashtag Recommendation, Ontology Construction, Hashtag Methods.

1. Introduction

In the past decade, Twitter has experienced tremendous success and has become one of the most important social network services. As the number of available tweets grows, the problem of managing tweets becomes extremely difficult. To avoid information overload, an efficient way called a hashtag, the '#' symbol used before a relevant keyword or phrase, was introduced by Twitter to categorize tweets. Hashtags also attracted much attention in various research areas such as sentiment classification [1] and topic analysis [2], [3].We note that recurrent neural networks (RNNs) have obtained outstanding performance in the representation learning field. Inspired by the recent improvement in document-level sentiment classification, in this paper, we propose an LSTMRNN model to learn semantic tweet representations for hashtag classification. Our method is based on the principle of compositionality [12], which states the meaning of a longer expression (e.g. a sentence) depends on the meanings of its units (e.g. a word). Specifically, our method models tweet representation in three steps: (a) it uses a skip-gram model with negative sampling (SGNS) to generate distributed wordrepresentations; (b) it utilizes a convolutional neural network (CNN) to compose sentence representations based on word embeddings; (c) it uses a long short-term memory (LSTM) model to encode the intrinsic (syntactic or semantic) relations between sentences into the tweet vectors. The tweet vectors are used as features to classify the hashtags of each tweet. The details of our metered are described in Section III. We conduct sufficient experiments to compare our proposed method with state-of-the-art approaches: (1) Firstly, we compare our LSTM-RNN method feed-forward neural networks with distributed word representations and approaches based on TFincludingCNN, SVM, and Random Forests; (2) Secondly, to explore the effectiveness of tweet representations,we compare to LSTM-RNNs which only consider word embeddings; (3) Finally, we compare different types of recurrent GRU). Our experiments reveal that: (1) tweet modeling with LSTM-RNNs outperforms state-of-the-art methods; (2) the LSTM unit obtains the best performance among three RNN units for capturing tweet semantics.

2. Related Works

This research work involves two main topics: pattern mining and hashtag recommendation. In the following, we present relevant related works to both topics with micro blogHashtag Recommendation: Hashtag recommendation on Twitter is a challenging problem because of the shortness and sparsity of tweets. Mazzia and Juett [5] apply a Naive Bayes model to classify tweets with hash tags and this method produces a ranked list of the top 20recommend 20ashtags. Zangerle et al. [6] compare three approaches based on the TF-IDF representations of tweets to recommend hashtags. Kywe et al. [7Kyleggest hash tags by combining hashtags of similar users and similar tweets. The TF-IDF scheme is used again in computing similar tweets. Otsuka et al. [9] propose the HF-IHU ranking scheme, which is a variation of TF-IDF, that considers hashtag relevancy. Thai and Sum [8] formulate recommending hashtags as a learning-to-rank problem and use Ranks to rank the candidate hashtags. Furthermore, this method only deals with tweets with URLs.RNNs are powerful models that have shown great promise in many natural language processing (NLP) tasks such as language modeling, question answering, sentiment classification, and machine translation. Here we briefly review two subareas:Language modeling, Machine translation. Pattern mining: Frequent pattern mining (FPM) [12]– [15] is a common and fundamental topic in data mining. FPM is a key phase of association-rule mining (ARM) but it has been generalized to many kinds of patterns, such as frequent sequential patterns [16], frequent episodes [17], and frequent sub graphs [18]. The goal of FPM is to discover all the desired patterns having support no lower than a given minimum support threshold. If a pattern has higher support than the threshold, it is called a frequent pattern; otherwise, it is called an infrequent pattern.Micro blogging: Micro blogging platforms are an amalgamation of blogging and instant messaging that allows bloggers to share their ideas, moods, and events with other people on the same platform in real-time. These platforms have become immensely popular; Twitter has example; has more than 313 million active users and monthly 1 billion unique visits to sites with embedded tweets [1]. Twitter, let users create freely hashtags and put them anywhere in their tweets, as long as they fit within the 140-character limit.

3. Methodology

Methodology includes the following stages Data collection: This stage creates the corpus of published tweets from the user tweets. Twitter Java API is integrated to retrieve the tweets on a JSON (JavaScript Object Notation) file. The JSON file is parsed to extract the hashtags for each tweet. The tweets are stored according to the timepublished. Natural Language Processing [50] may be incorporated to refine the extraction results by removing URLs (Uniform Resource Locator), special characters exceptfor the# character, unifying dates, and letter levels (upper or lower case,e,s), and so on. In addition, a filtering strategy is used to replace combined hash simple hashtags. For instance,the hashtag #EMABiggestF JustinBibber is replaced by #JustinBeiger#BLOGGER and #blogger represents the same hashtag but with different writing styles, these hashtags areunified to the same hashtag #blogger.Mining process: After transforming the user tweets to the corpus of the published tweets, the temporalhigh average utility patterns method is run to derive the relevant patterns and design the rules-based systemcalled KS represented by a set of the temporal top k high average utility hashtags. The published tweets are transformed into the temporal transactional database as described by Definitions 2 and 3, where each tweet is considered as a transaction and each hashtagis an item. A two-phase algorithm [48] is then adopted to discover the temporal top k high average utility hashtags including three steps: the average-utility upper bound value (See Definition 5) is used to prune the candidate itemsets, scanning the temporal transactional database only once to discover the high average utility hashtags, sorting the extracted patterns according to the average utility value and then selecting the top k high average utility patterns the set of the irrelevant tweets noted irreverent is deduced.Ontology construction: A given morphine tweet Oi is usually represented by a set of keywords different from Romeo set of hashtags in KS (i,e 8t 2 Oi; 8p 2 KS; t 62 p), but they represent the same meaning. For instance,consider the keywords of the orpheline tweet Oi = and the high average utility hashtags p = #Summer2018;#WorldCupg, Oi 6= p, but #WorldCup is an event.

4. Lstm-Rnn for Tweet Composition

Before introducing the tweet composition, we first reviewthe long short-term memorizecurrent neural network (LSTM-RNN).

(a) RNN (b) LSTM

(a) RNN (b) LSTM

FIGURE 1. The internal structure of the standard RNN unit (left) and LSTM (right)http://colah.github.io/posts/2015-08-Understanding-LSTMs/.

All RNNs have the form of a chain of repeating componentsof a neural network. The repeating component in standard RNNshas a very simple structure, such as a single tan layer (Fig.1 (a)) LSTM also has this chain-like form, but its repeatingmodule also called the memory cell has a very different structure.Instead of having a single layer, there are four, interacting ina very special way, fig.1 (b).LSTM was introduced by[25] primarily to solve the problem of vanishing gradients instandard RNNs.The key to LSTM is the cell state, a horizontal line throughthe top of the memory cell. To clearly describe the intuition behind LSTMwe can imagine the cell state as a conveyor belt.It runs through the entire chain, with only minor linear interactions.Each memory cell could remove or add information tothe cell state via carefully designed structures called gates. Agate consists of a sigmoid layer and a pointwise multiplicationoperation. The output value of the sigmoid layer ranges from 0to 1 which can be signeetouch of information that should beet through. Here, we take memory cell c as an example todescribe its detailed structure: Forget gate: The forget gate provides a forgetting coefficientby looking at the input layer txand previously hidden layer ht−1 for cell state Ct−1. The coefficient ranges from0 to 1 and controls the information from Ct−1 to Ct.

$$ft = \sigma(Wf \bullet [ht-1, xt] + bf) \quad (1)$$

Input node: This unit also takes activation from the input layertaxed previously hidden layer ht−1. Typically, the tank layer is used to process the summed weighted input.

$$gt = \tanh(Wg \bullet [ht-1, xt] + bg) \quad (2)$$

Input gate: The input gate decides which values shouldbe updated in Ct−1 and its output multiplies the value of the input node to get a new candidate of Ct.

$$it = \sigma(Wi \bullet [ht-1, xt] + bi) \quad (3)$$

Internal state: The heart of memory cell c is the internal state:

$$Ct.Ct = ft * Ct-1 + it * gt \quad (4)$$

Output gate: The hidden layer is produced by theinternal state Ct and the value of the output gate

$$ot.ot = \sigma(Wo \bullet [ht-1, xt] + bo) \quad (5)$$

$$ht = ot * \tanh(Ct) \quad (6)$$

Since LSTM-RNNs are effective at capturing long-term memory dependencies without suffering from vanishing gradients, they have been used for many NLP tasks. In this paper, we novelty use LSTM-RNNs to learn semantic vector representations for tweets. The process of generating tweet representationsgiven a variable-length sequence of sentence vectors as input, LSTM-RNNs produce a fixed-length tweet vector. Unlike the recently proposed Encoder-Decoder models in neural machine translation, most of them just output the last hiddenstate as the fixed-size vector [26]. We not only utilize the sequence summarization property of LSTM-RNNs but also consider the global semantics of each tweet. Here, we average each hidden statehttp produce the tweet vector.More precisely, the process of generating a tweet vector works as follows. For tweet d, we first generate its sentence vectors s{x1, x2, xi, xn} via the CNN model. Then these sentence vectors are regarded as the input to the LSTM-RNNmodel. In each timestamp t, LSTM-RNN processes the inputtext and previous cell state Ct−1 to generate the currently hidden state hand output it. After getting all of the hidden states {h1, h2, hn}, we calculate their average and output it as a dissector representation.

5. Hashtag Recommendation Methods

- Text-based methods,
- Hybrid user-based methods; and
- Hybridmiscellaneous methods.

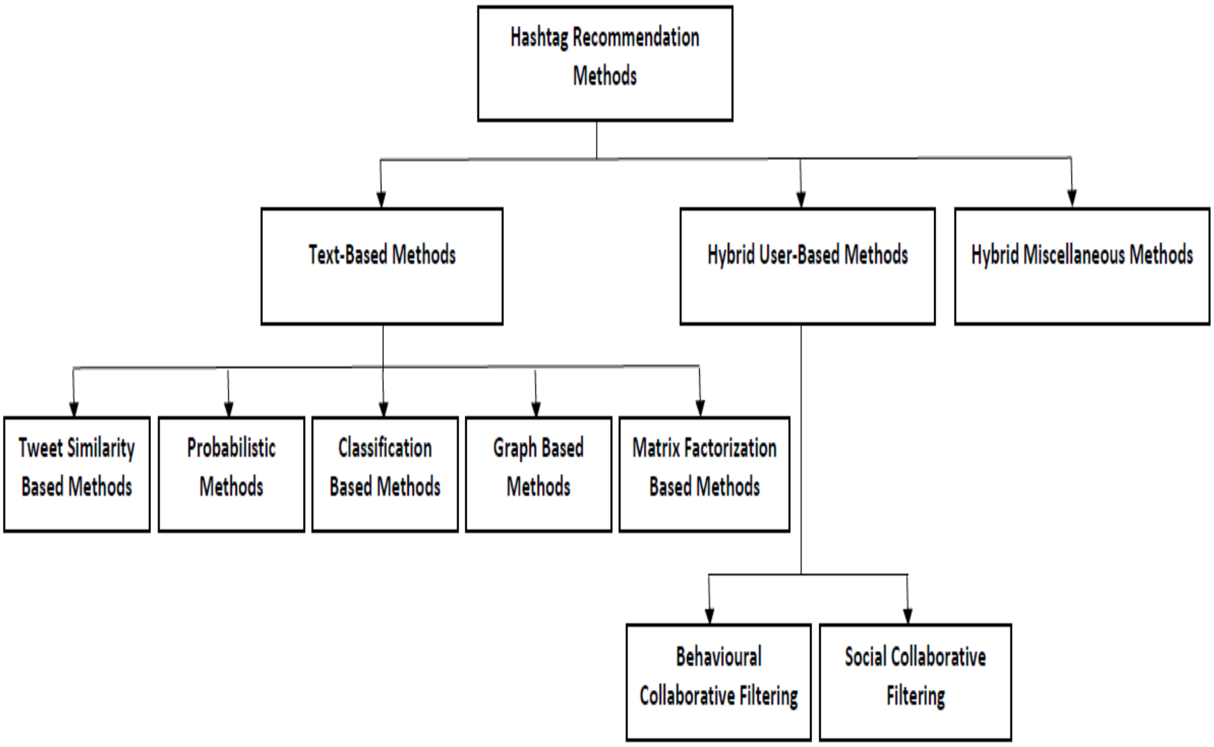

FIGURE 2. Our proposed taxonomy of the hashtag recommendation methods.

Text-based methods form the largest category, comprising five subcategories, asshownin Figure 1. These five subcategories are detailed in the subsections below.Tweet Similarity-Based MethodsTweet similarity methods used in hashtag recommendation find similar tweets to aquery tweet using similarity measures [8,32,33]. The tweet similarity method was firstdeveloped in hashtag recommendation by Zangerle et al. [8], who used the TF-IDF approachto represent tweets as one-hot vectors.Probabilistic MethodsNaive Bayes is a machine learning technique based on the Bayes Theorem. Mazzia et al. [42] proposed one of the earliest general hashtag recommendation models using Naïve Bayes to calculate the conditional probability of a hashtag given a set of terms in the tweet as P(hjt1, t2, . . . , TN = P(t1, t2,. . . , tnjh)P(h)/P(t1, t2, . . . , TN= P(hjt1)P(hjt2) . . . P(it The result is a value between 0 and 1. The hashtagswith the highest probabilities were recommended. However, some terms and hashtags never co-occur, leading to a joint the conditional probability of 0 even if some words have a high probability score.Classification Based MethodsClassification is a supervised learning method that trains a classifier to predict a class label given an instance. While binary classifiers tackle only two categories, multiclass classifiers multiple categories.Hashtag recommendation is commonly tackled as a multi-class classification problem of hashtags [49,52,53,56], where every hashtag is considered as a distinct class label. The intuition of theclassification-based hashtag recommendation is that the abundance of posts and hashtags equips classifiers with an immenseamount of labeled data to learn strong representations [80]. Classification-based hashtag recommendation requires less task-specific assumption and engineering in comparison with topic-based hashtag recommendation [80].Hybrid User-Based MethodsCollaborative filtering is a user-based method that recommends items based on other users' collaborative behaviors, the similarity of their interests, shared topics, or social relations. The set of similar users is also called like-minded users. Although collaborative filtering is a large category in the traditional recommendation systems, it was rarely used as the sole method for recommending hashtags due to data sparseness and the cold startproblem. Data sparseness is caused by the free style of writing hashtags and the increasing number of usersHybridMiscellaneous MethodsIn this category of research, multi-modalities and multi-factors are incorporated to recommend hashtags under various assumptions. Join et al. [12] assumed that users frequently tweet about topics they are interested in. Thus, their hashtag recommendation model determined the relationship between the significant terms and hashtags by their shared topics. With 16 predefined class labels (art and design, books, business, etc.), they classified significant terms in a tweet into one of these class labels using Naive Bayes. To personalizetheir model, they extracted the significant terms from every user and classified them into one of the predefined class categories. They ranked candidates' hashtags based on the similarity of the terms to the hashtags, user interests, and popularity. They found that the performance of the hashtag recommendation on sports tweets outperformed the performance of tweets in other categories, and the performance on news tweets was the poorest compared to all other categories.

6. Compared Methods and Implementation

We compare our LSTM-RNN model with several state-of-the-art hashtag classification approaches described briefly as follows:kNN: The kNN method classifies a new tweet d1 via a similarity measure. Here, we use the cosine distanceas the similarity measure. That is, kNN computes the distance between d1 and another tweet d2 from thetraining set: cos(d1, d2) = d1 • d2 ||d1|| • ||d2|| (12)The features were generated by computing the TF-IDF scheme of each tweet. We set k∈{50,100}. The CNNThe classifier was trained by using the kit kitikit-learn toolkit (sklearn) [29]. SVM: The SVM algorithm is well known for its very good practical results. It's also a powerful classifier forhashtag classification. We made use of the libsvm [30] to train SVM: (1) SVM-Linear with linear kernel; (2) SVMRBF with RBF kernel. Like CNN, we also used the TF-IDF representations as the tweet features.Random Forests: The Random Forests algorithm is an ensemble learning method for classification and othertasks that operates by constructing a multitude of decision trees and outputting the class that is the mode ofthe classes of the individual trees. The number of decision trees we set was 50. We also made use of sklearn to train Random Forests. The TF-IDF scheme was used again to generate Random Forests input.Feed Forward Neural Network (FFNN): A FFNN is an artificial neural network where connections between theunits do not form a cycle. FFNNs are widely used in many practical applications, and [11] first uses FFNNs toclassify hashtags with distributed word representations. We implemented three ReLU layers FFNN via Cafe[31] and used the word embeddings as input.LSTM-RNN: We made use of Cafe to implement a two layers LSTM-RNN model with 200 cells at each layer.Afterward, we got two LSTM-RNN models: LSTMword took word embeddings as input and LSTM-tweetconsidered tweet vectors as input.RNN: We implemented a two layers RNN with Cafe which considered the tweet vectors as input called RNNtweet. Gated Recurrent Unit (GRU): GRU is another sophisticated recurrent unit proposed in [32]. Recently [33]empirically evaluated GRU on some sequence modeling tasks and found GRU to be comparable to LSTM. We also used Cafe to implement a two layers GRU-tweet which considered the tweet vectors as input.It is important to note that our method is supervised. Therefore we do not compare it with any unsupervised methods such as LDA [13]. The training details about neural networks are given below: (1) we used the gensim3 [34] Python implementation of Word2Vec to train a word embedding model. Inspired by Lai et al. [35] that more training data would be beneficial to train a precise model, we combined the full English wikipedia4and the New York Times corpus5 to obtain the genesis training corpus. Each word has a 300-dimensional feature vector. (2) LSTM parameters were initialized from a uniform distribution between [-0.05,0.05]. For CNN, we initialized the weights in each layer from a zero-mean Gaussian distribution with a standard deviation of 0.01. The biases were initialized with the constant 0.1. (3) All of the neural networks were trained using momentum-accelerated mini-batch SGD and momentum set to 0.9. (4) The batch size of RNN was set to 32 (32 tweets).

7. Experimental Results

Accuracy Performance Comparison lists the accuracies of all the approaches on our test set. We havethe following experimental results LSTM-tweet achieves the highest accuracy based on distributed tweet representations and LSTMs. This demonstrates that our proposed method can suggest more accurate hashtags; GRU-tweet falls behind LSTM-tweet but performs better than RNN-tweet. This is because GRUs can capture more semantics and have a better capacity to summarize tweet information than standard RNNs;Neural Network methods perform better than CNN, SVM, and Random Forests. One reason may be the input for neural networks is word embeddings or tweet vectors rather than TF-IDF forms of tweets. The distributed word or tweet representations contain more semantic features about words. In addition, we think the great ability to generalization and respond to unexpected patterns of neural networks is anotherreason; (4) Neural Networks based on tweet vectors outperform those based on word vectors. A reasonable explanation is that the effectiveness of our tweet modeling with LSTMs and sentence composition with CNNs.Citrate Performance Comparison Table III shows the overall hit rate results. As shown by the highest hit rates in bold type, it is obvious that LSTM-tweet takes the lead. Compared to the RNN unit and GRU unit, the LSTM unit is a fine choice for hash tag recommendation. Fig. 6 also illustrates the results of standard RNN, GRU, and LSTM. In this study, we not only novelty use LSTM-RNNs to classify hash tags but also propose a framework for modeling tweets based on the principle of compositionality. Experiments show that tweet vectors are more suitable as input for neural networks than word embeddings. This is because tweet vectors contain word features, local relations of words, and global semantics of sentences. Without any feature engineering, our end-to-end
The method achieves the best performance and could recommend suitable hashtags for new tweets.

TABLE 1. Accuracy results(higher is better) for hashtag classification. The best method is in **bold**

Method	Accuracy
KNN-50	19.4
KNN-100	22.1
SVM-Linear	22.7
SVM-RBF	21.3
Random Forests	24.7
FFNN	24.2
LSTM-word	25.9
RNN-tweet	26.3
GRU-tweet	27.7
LSTM-tweet	**28.6**

8. Conclusion

In this paper, we introduce neural networks (CNNs and LSTM-RNNs) for tweet modeling to recommend hashtagson Twitter. Our proposed distributed tweet representations not only encode word features but also contain semanticsof sentences and sentence relations. The method to suggest hashtags consists of four components: word embedding generation, sentence composition, tweet composition, and hashtagclassification. The experimental results show that our LSTMRNNs can outperform other state-of-the-art supervised methods such as SVM, Random Forests,and FFNNs for hashtag recommendation. We also evaluate three commonly used RNN units: standard RNN, GRU, and LSTM. LSTM unit achieves the best performance in our dataset.Pattern mining method to solve the hashtag recommendation problem. The proposed approach PM-HRec benefits from the high average-utility patterns to improve the hashtag recommendation of the orpheline tweets. Offline processing is first performed to transform the corpus into a transactional database considering the temporal information of tagged tweets. It discovers the top k high average utility hashtags by adopting the two-phase algorithm. Irrelevant tagged tweets and the ontology of tagged tweets are also determined in thisoffline step, performed only once regardless of the number of orpheline tweets processed.Hashtag recommendation systems for tweets have evolved within the field of online social networks. It also presents a new taxonomy for hashtag recommendation of tweets based on their methodologies. The taxonomy classifies hashtag recommendation methods for tweets into three main categories: text-based, hybrid user-based, and hybrid miscellaneous methods. Text-based methods find hashtags similar to what a user intends to adopt based on the textual information. This category is further classified into tweet-similarity-based methods, probabilistic methods, classification-based methods, graph-based methods, and matrix factorization-based methods. Since methods of collaborative filtering suffer from the cold-start problem, they are integrated with other methods. Hybrid user-based hashtag recommendation methods recommendhashtags based on the similarity of the users' behavior, interests, or relations. This category is further classified into behavioral and social collaborative filtering methods. Hybrid miscellaneous hashtag recommendations take advantage of multi-modalities and multi-factors to recommend the hashtags. Regardless of the specific techniques employed, it has become clear that the best outcome can be achieved using hybrid methods (user-based or miscellaneous) for their ability to overcome problems occurring with content-based and collaborative filtering methods. It was noticed that understanding various factors that affect the performance of hashtag recommendation and the underlying assumptions has a significant impact on the algorithmic approach that should be considered.The current paper does not consider the other tweet components (URL,) in the recommending hashtags task. Our future work will focus on expanding new fields of tweets such as temporal, geographical, and user information, in the other hand, we will try to train some DL architectures on different semantic knowledge bases such as Dbpedia or BabelNet to improve the accuracy of results. Acknowledgment I am delighted to express my heartfelt appreciation to our department's head and staff, as well as family and friends. This paper is made possible by their encouragement, assistance, and support.

References

[1]. D. Davidov, O. Tsur, and A. Rappoport, “Enhanced sentiment learning using Twitter hashtags and smileys,” in Proceedings of the 23rd International Conference on Computational Linguistics: Posters. Association for Computational Linguistics, 2010, pp. 241–249.
[2]. A. Cui, M. Zhang, Y. Liu, S. Ma, and K. Zhang, “Discover breaking events with popular hashtags on twitter,” in Proceedings of the 21st ACM International conference on Information and knowledge management. ACM, 2012, pp. 1794–1798.
[3]. X. Meng, F. Wei, X. Liu, M. Zhou, S. Li, and H. Wang, “Entity-centric topic-oriented opinion summarization in Twitter,” in Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2012, pp. 379–387.
[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 of the 20th ACM international conference on Information and knowledge management. ACM, 2011, pp. 1031–1040.
[5]. A. Mazzia and J. Juett, “Suggesting hashtags on Twitter,” EECS 545m, Machine Learning, Computer Science and Engineering, University of Michigan, 2009.
[6]. A. Belhadi, Y. Djenouri, J. C.-W. Lin, C. Zhang, and A. Cano, “Exploring pattern mining algorithms for hashtag retrieval problem,” IEEE Access, vol. 8, pp. 10 569–10 583, 2020.
[7]. R. Agrawal, T. Imieli´nski, and A. Swami, “Mining association rules between sets of items in large databases,” in ACM SIGMOD Record, vol. 22, no. 2, 1993, pp. 207–216.
[8]. R. Agrawal, R. Srikant, et al., “Fast algorithms for mining association rules,” in International Conference of Very Large Data Bases, vol. 1215, 1994, pp. 487–499.
[9]. J. Han, J. Pei, Y. Yin, and R. Mao, “Mining frequent patterns without candidate generation: a frequent-pattern tree approach,” Data Mining and Knowledge Discovery, vol. 8, no. 1, pp. 53–87, 2004.
[10]. C. C. Aggarwal, Y. Li, J. Wang, and J. Wang, “Frequent pattern mining with uncertain data,” in ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2009, pp. 29–38.
[11]. J. Han, J. Pei, B. Mortazavi-Asl, H. Pinto, Q. Chen, U. Dayal, and M. Hsu, “PrefixSpan: Mining sequential patterns efficiently by prefix-projected pattern growth,” in International Conference on Data Engineering, 2001, pp. 215–224.
[12]. H. Mannila, H. Toivonen, and A. I. Verkamo, “Discovery of frequent episodes in event sequences,” Data Mining and Knowledge Discovery, vol. 1, no. 3, pp. 259–289, 1997.
[13]. C. Jiang, F. Coenen, and M. Zito, “A survey of frequent subgraph mining algorithms,” The Knowledge Engineering Review, vol. 28, no. 1, pp. 75– 105, 2013.
[14]. Feng, W.; Wang, J. We can learn your #hashtags: Connecting tweets to explicit topics. In Proceedings of the 2014 IEEE 30th International Conference on Data Engineering, Chicago, IL, USA, 31 March–4 April 2014; pp. 856–867. [CrossRef]
[15]. Al-Dhelaan, M.; Alhawasi, H. Graph Summarization for Hashtag Recommendation. In Proceedings of the 2015 3rd International Conference on Future Internet of Things and Cloud, Rome, Italy, 24–26 August 2015; pp. 698–702. [CrossRef]
[16]. Zhang, Q.;Wang, J.; Huang, H.; Huang, X.; Gong, Y. Hashtag Recommendation for Multimodal Microblog Using Co-Attention Network. In Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17, Melbourne, Australia, 19–25 August 2017; pp. 3420–3426. [CrossRef]
[17]. Alsina, A.; Datta, A.; Li, J.; Huynh, D. Empirical Analysis of Factors Influencing Twitter Hashtag Recommendation on DetectedCommunities. In Proceedings of the Advanced Data Mining and Applications—13th International Conference, ADMA 2017, Singapore, 5–6 November 2017; pp. 119–131. [CrossRef]
[18]. Li, Y.; Jiang, J.; Liu, T.; Qiu, M.; Sun, X. Personalized Microtopic Recommendation on Microblogs. ACM Trans. Intell. Syst. Technol. 2017, 8. [CrossRef]
[19]. Kalloubi F, Nfaoui E H, El Beqqali O: Harnessing semantic features for large scale content based hashtag recommendations on microblogging platforms. International Journal on Semantic Web and Information Systems, 13 (1), pp. 6381. (2017).
[20]. Ben Lhachemi N, Nfaoui E H: An extended spreading activation technique for hashtag recommendation in microblogging platforms. The 7th International Conference on Web Intelligence, Mining and Semantics. (2017).