Knowledge and Information Systems (2021) 63:175–198
https://doi.org/10.1007/s10115-020-01515-7

REGULAR PAPER

Hashtag recommendation for short social media texts using word-embeddings and external knowledge

Nagendra Kumar[1] · Eshwanth Baskaran[2] · Anand Konjengbam[2] · Manish Singh[2]

Received: 7 December 2018 / Revised: 25 September 2020 / Accepted: 27 September 2020 /
Published online: 14 October 2020

Abstract
With the rapid growth of Twitter in recent years, there has been a tremendous increase in the number of tweets generated by users. Twitter allows users to make use of hashtags to facilitate effective categorization and retrieval of tweets. Despite the usefulness of hashtags, a major fraction of tweets do not contain hashtags. Several methods have been proposed to recommend hashtags based on lexical and topical features of tweets. However, semantic features and data sparsity in tweet representation have rarely been addressed by existing methods. In this paper, we propose a novel method for hashtag recommendation that resolves the data sparseness problem by exploiting the most relevant tweet information from external knowledge sources. In addition to lexical features and topical features, the proposed method incorporates the semantic features based on word-embeddings and user influence feature based on users' influential position. To gain the advantage of various hashtag recommendation methods based on different features, our proposed method aggregates these methods using learning-to-rank and generates top-ranked hashtags. Experimental results show that the proposed method significantly outperforms the current state-of-the-art methods.

Keywords Hashtag recommendation · Social media analysis · Information extraction and filtering · Semantic knowledge bases

✉ Nagendra Kumar
nagendra@iiti.ac.in

Eshwanth Baskaran
cs14btech11012@iith.ac.in

Anand Konjengbam
cs14resch11004@iith.ac.in

Manish Singh
msingh@iith.ac.in

[1] Indian Institute of Technology Indore, Indore 453552, India

[2] Indian Institute of Technology Hyderabad, Hyderabad 502285, India

1 Introduction

Over the past few years, hashtags have been widely used in social media to provide the topical information of user-generated content. Hashtags are shown to be useful in many applications including event detection [1], information diffusion [5], sentiment analysis [6], information retrieval [9], text classification [42], and so on. However, hashtags are manually created, and many social media texts do not contain hashtags due to users' uncertainty and unwillingness to use hashtags. We therefore take up the task of automatically recommending the hashtags to social media texts.

In this paper, we use publicly accessible tweets from Twitter to create our dataset. Twitter is one of the biggest social networking platforms with millions of active users. Users share information with their friends and followers in the form of tweets. Tweets are short texts with a maximum length of 280 characters. Due to the length constraint, tweets are usually broadcasted with limited context. Hashtags provide a better representation of tweets and facilitate improved user participation in discussions [4]. However, a large fraction of tweets do not contain any hashtag due to insufficient knowledge about the relevant hashtags [18,28].

Several methods [22,49] have been proposed to recommend hashtags for a tweet. Existing works on hashtag recommendation depend on the intrinsic information present in the tweet such as similar tweets [48], similar topics [39], and link information [38]. Unlike typical texts with many words or sentences, tweets often consist of one or two sentences, which present several challenges in existing methods. Short tweets do not provide sufficient term co-occurrence information. Employing traditional text-matching techniques [48] to find similar tweets have several limitations due to the sparse representation of tweets. Recommending hashtags based on similar topics using topic modeling techniques [14,39] may fail as these techniques are developed for long documents that contain sufficient term co-occurrences but short tweets contain a very limited term co-occurrence information. A large fraction of tweets do not contain links to external sources, which prevent to get co-occurrence information from external sources. As a result, existing link-based techniques [33,38] also do not perform well in understanding the context of a general tweet, thereby recommending better hashtags for a tweet.

The focus of this paper is to develop an effective method for recommending hashtags to short tweets using intrinsic information present in tweets and extrinsic information from external knowledge sources. The extrinsic information has been proven to be useful for many applications [19,46]. This information can also be used to fill the semantic gap in short tweet representation by getting sufficient term co-occurrences. The semantic gap problem can be realized from the following example tweet: 'Breaking News: Hope Hicks, told Maggie Haberman she plans to resign from White House #HopeHicks #Maggie #Haberman #Trump #Hope #Hicks #News #America #Pennsylvania #WhiteHouse #US.' The tweet is not only annotated with some of the keywords present in it but also with the terms that are not present in the tweet such as #Trump, #America, #Pennsylvania, #US. Although these keywords are semantically related to the tweet, the information about these keywords is not readily available. This is referred to as the semantic gap problem. To reduce the semantic gap, we utilize the extrinsic information from external sources by incorporating Wikipedia, word-embeddings, and web-pages. Furthermore, we propose a word-embedding-based tweet similarity method to recommend hashtags for a tweet from semantically related tweets, which can also reduce the semantic gap problem.

A tweet often contains hashtags that were originally generated by a popular or influential user. If a popular user starts a campaign or event with a hashtag, many of his followers and other users start promoting the event by putting the same hashtag as generated by the popular user. A popular user usually has a large number of users connected to him. As a result, hashtags created by a popular user are most commonly used in social media compared to that generated by an ordinary user. Therefore, it is essential to look into the user's influential position in the Twitter network while recommending hashtags for a tweet. In this paper, we determine user influence by considering user connections and status to recommend better hashtags.

Our key contributions are as follows:

- We propose a novel word-embedding-based framework to recommend hashtags for tweets.
- We address the data sparsity problem of short tweet representation by incorporating the extrinsic information of tweets.
- We present a user influence metric to recommend the improved hashtags.
- To achieve better performance, we integrate different hashtag recommendation methods and recommend the top-ranked hashtags generated by these methods using learning-to-rank.
- Our experimental results demonstrate that the proposed method achieves a significant improvement as compared to the current state-of-the-art methods.

The remainder of this paper is organized as follows. In Sect. 2, we briefly survey the related work. Section 3 presents our methodologies. We proceed by describing the experimental evaluations in Sect. 4. Finally, in Sect. 5, we conclude our work.

2 Related work

In this section, we first give a brief summary of the works on hashtags in Twitter and then review the related works on hashtag recommendation.

2.1 Hashtags in Twitter

The widespread use of hashtags has attracted significant research attention. Many studies have been proposed to address different aspects of hashtags such as hashtag popularity prediction [26,27,40], hashtag diffusion [5,35,43], hashtag sentiment analysis [6,21,42], hashtag recommendation [14,22,47], etc. Due to the usefulness of hashtags, many works have been proposed recently to recommend hashtags that help in better retrieval and categorization of tweets. We can categorize these approaches into three main classes, namely content-based hashtag recommendation, topic-based hashtag recommendation, and personalized hashtag recommendation.

2.2 Content-based hashtag recommendation

A major fraction of techniques on hashtag recommendation recommends hashtags for a tweet based on the tweet content [3,32,45,47]. Content-based hashtag recommendation methods usually exploit the similarity between tweets by utilizing their textual features. Zangerle et al. [47] proposed three hashtag recommendation methods, namely (a) OverallPopularityRank, (b) RecommendationPopularityRank, and (c) SimilarityRank. For a given tweet, they first determined the most similar tweets using TF-IDF similarity, and then ranked the hashtags from similar tweets using the three approaches mentioned above. OverallPopularityRank and RecommendationPopularityRank rank the hashtags by considering the popularity of hashtags in the whole dataset and most similar tweets, respectively. SimilarityRank ranks the hashtags based on the similarity score of most similar tweets. They reported that SimilarityRank performs the best in recommending hashtags. Mishne et al. [31] also recommended tags to blog posts based on the textual content of posts. They recommended tags for a new post from the existing contents based on TF-IDF similarity. Otsuka et al. [32] proposed a variant of TF-IDF ranking method, Hashtag Frequency-Inverse Hashtag Ubiquity (HF-IHU). HF-IHU is a hashtag ranking scheme that considers the relevance of hashtags, unlike TF-IDF ranking method that determines the relevance of terms. Kalloubi et al. [20] used entities within a tweet to compute similarity between tweets and recommended hashtags from top-*k* similar tweets. All the methods mentioned above depend on the textual content present in short tweets. However, tweets do not contain sufficient textual terms, which reduces the probability of relevant tweets to be selected for recommendation. On the other hand, our approach focuses on solving this problem by providing adequate contextual information from external knowledge sources to represent a tweet.

2.3 Topic-based hashtag recommendation

Topic models such as Probabilistic Latent Semantic Analysis (PLSA) and Latent Dirichlet Allocation (LDA) are found to be very useful to discover the latent topics from a text corpus. These methods have also been successfully employed to recommend hashtags for tweets [14,28,39,49]. Godin et al. [14] showed that there are many challenges in recommending hashtags based on the similarity of tweets due to their sparse representation. They employed LDA to recommend hashtags for a tweet based on latent topics of the tweet. LDA model was trained on existing tweets and the same model was used to generate the topic distribution of a new tweet. Top keywords from the dominant topic to this new tweet were recommended as hashtags. She et al. [39] proposed a supervised topic model to recommend hashtags for a tweet. They considered each tweet as a local topic and hashtags as labels of the local topic. They assumed that there is a global topic for the corpus. Using these assumptions, they identified the relationship among hashtags, words and topics of tweets. Ding et al. [7] proposed topic translation model to recommend the hashtags using topic modeling. They assumed that the content of a tweet and its hashtags are based on the same theme but written in different languages. The key idea behind their approach is to find the latent topics of tweets and recommend hashtags based on a particular topic. Ma et al. [28] introduced two PLSA-style topic models that capture relations between latent topics of tweets and respective hashtags. However, topic modeling algorithms were developed for long documents and may not work well in case of short tweets where sufficient term co-occurrence information is not available. In this paper, we incorporate semantically related information of a tweet from external sources, which provide sufficient term co-occurrences.

2.4 Personalized hashtag recommendation

Content-based and topic-based hashtag recommendation methods recommend hashtags based on only textual information present in the tweet. Besides these methods, several personalized hashtag recommendation methods are proposed, which also consider the users' interests while recommending hashtags. Zhao et al. [50] proposed personalized hashtag recommendation based on user-topic distribution. They first identified top-*k* similar users based on user-topic distribution and later determined hashtags counts associated to these users to recommend hashtags. Wang to et al. [44] proposed a personalized hashtag recommendation based on collaborative filtering and topical information (or content relevance). They combined global content information to capture topical semantic of posts and users' preferences from other similar users to obtain personalized information. Liang et al. [24] used various relations among tags, users, and items to determine the meaning of each user and tag. They utilize these relationships for a personalized recommendation. Kywe et al. [22] presented a method to incorporate the users' preferences in hashtag recommendation. They combined hashtags of both similar tweets and similar users to recommend personalized hashtags. Although they considered similar users, their method would recommend the hashtag only from similar tweets if target users have never used hashtags in the past. TF-IDF was again used to construct the feature vectors which is a sparse representation of short tweets and does not consider the semantics of tweets. However, we use word-embeddings to capture the semantics of the tweet. We also prepare a dense representation of tweets by considering both word-embeddings as well as external features.

Apart from the above-mentioned hashtag recommendation methods, a few graph-based methods [10,11,16,33] have been successfully employed to recommend hashtags. These methods find the correlation between hashtags and tweet texts by constructing a graph and recommend hashtags based on correlation. Our method is complementary to the graph-based methods as we also find the correlation between hashtags and tweets to recommend hashtags for a new tweet. Furthermore, Surendra et al. [38] proposed a method to recommend hashtags using hyperlinks present in tweets. However, we observe that 85.3% of tweets do not contain hyperlinks to external documents and the method [38] may fail to recommend relevant hashtags for these tweets. In this paper, we incorporate semantically related information of a tweet from external knowledge sources.

All the above-mentioned hashtag recommendation methods are primarily based on original tweet-text. Due to data sparsity and semantic gap problems, existing methods do not perform well in understanding the context of a general tweet, thereby recommending better hashtags for the tweet. In this paper, we exploit external knowledge to provide a better representation of tweets. Unlike existing methods, we address the data sparsity and semantic gap problem of tweets using information from multiple external sources. We propose word-embedding hashtag recommendation method that also reduces the semantic gap problem. To build a better hashtag recommendation system that can outperform the individual recommendation system, we aggregate different hashtag recommendation methods using learning-to-rank.

3 Methodology

Figure 1 shows the architectural overview of the proposed hashtag recommendation system. We divide the task of hashtag recommendation into five primary steps: (a) extrinsic feature extraction; (b) feature selection and processing; (c) candidate hashtag generation; (d) user influence score computation; (e) candidate hashtag recommendation. The proposed system first extracts the tweets from Twitter and stores them in a tweet database. It then extracts extrinsic features from external sources to obtain more context for short tweets. These features are cleaned by removing noisy features. After pre-processing of these features, different candidate hashtag generation techniques are proposed to recommend hashtags for a tweet. All these hashtag generation methods and user influence score are combined using learning-to-rank to recommend better hashtags for a tweet.

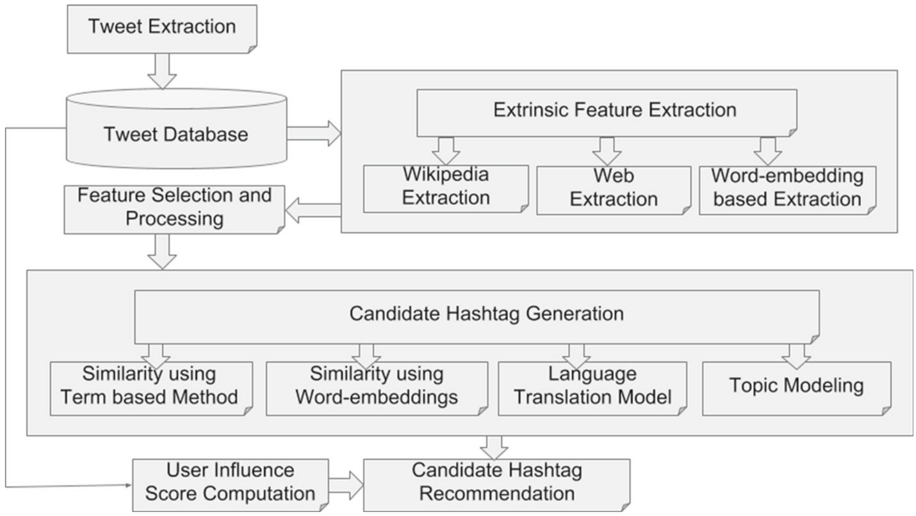

Tweet Extraction
Tweet Database
Extrinsic Feature Extraction
Wikipedia Extraction
Web Extraction
Word-embedding based Extraction
Feature Selection and Processing
Candidate Hashtag Generation
Similarity using Term based Method
Similarity using Word-embeddings
Language Translation Model
Topic Modeling
User Influence Score Computation
Candidate Hashtag Recommendation

Fig. 1 Systems architecture of hashtag recommendation

3.1 Extrinsic feature extraction

Unlike large documents or blog posts, tweets are short in length. Tweets do not contain sufficient terms or co-occurrence information of terms, which present a great challenge in hashtag recommendation. Employing tweet-tweet similarity methods on short texts may not achieve satisfactory results. Since there is very less context in tweets, it is essential to get more contextual information about tweets from external sources to recommend better hashtags. We extract extrinsic or external features for tweets from external knowledge sources using internal features of tweets. This extracted information from external sources provides an alternate and richer representation of a tweet. We collect external information from multiple sources as follows.

3.1.1 Wikipedia extraction

Wikipedia offers a vast amount of domain-specific knowledge. There is a common phenomenon that entities which occur in the same contexts tend to purport similar or related information [13,17]. Existing hashtag recommendation methods can be improved by adding the similar entities that are highly related to actual entities present in a tweet. Wikipedia can provide related entities and contextual information related to entities present in a tweet. We therefore take each entity[1] from the tweet and query Wikipedia[2] to get the Wikipedia page linked to that entity. In this paper, the term 'entity' refers to the 'named entity' obtained using a widely used named entity recognizer (NER), namely Stanford CoreNLP [29]. As most of the entity extraction methods are case-sensitive, we use TrueCase Annotator from CoreNLP to identify true cases of all the terms in ill-formed tweets. For instance, if a tweet 'Mr. Chris Cornell dies at sound garden concert in detroit' is processed by an entity recognizer, it may

[1] We also tried querying complete tweet to Wikipedia but noticed poor results.

[2] https://www.mediawiki.org/wiki/API:Main_page.

not detect the entities ‘sound garden’ and ‘detroit,’ which are in lowercase. Moreover, there are a few challenges while extracting knowledge from Wikipedia as follows:
Disambiguation: In the microblogging platform, users do not use a unique name for an entity and often use many ambiguous name. Due to ambiguity, querying an ambiguous term (or entity) to Wikipedia returns multiple pages. For example, querying a term ‘Trump’ results in several Wikipedia pages such as ‘Donald Trump,’ ‘Melania Trump,’ ‘Lara Trump,’ etc. To avoid the ambiguity, we process the complete tweet and use collective agreement between all possible ambiguous sense (or meaning) of that entity and other entities in the tweet [12].

Let us consider A_T is a set of all entities or anchors occurring in a tweet T, i is a anchor term present in the tweet, P_i is a page linked to anchor term i. $\mathrm{Pg}(i)$ ($P_i \in \mathrm{Pg}(i)$) is a set of all the pages linked to i. In order to disambiguate each anchor $i \in A_T$, relevance score $rel_i(P_i)$ is computed for all $P_i \in \mathrm{Pg}(i)$ and the page P_i with the highest value of $\mathrm{rel}_i(P_i)$ is picked for annotation $i \rightarrow P_i$. The relevance score, $\mathrm{rel}_i(P_i)$ is computed by considering the vote, $\mathrm{vote}_j(P_i)$ received from other anchor terms j present in A_T:

$$\mathrm{rel}_i(P_i) = \sum_{j \in A_T \setminus \{i\}} \mathrm{vote}_j(P_i) \tag{1}$$

The vote, $vote_j(P_i)$ received from other anchor terms j to a page P_i (linked to the anchor term i) is computed as follows:

$$\mathrm{vote}_j(P_i) = \frac{\sum_{P_j \in \mathrm{Pg}(j)} \mathrm{rel}(P_j, P_i).\mathrm{Pr}(P_j|j)}{|\mathrm{Pg}(j)|} \tag{2}$$

where prior probability $\mathrm{Pr}(p_j/j)$ is computed by dividing the number of times j occurs as an anchor in Wikipedia with the number of times j occurs in Wikipedia (with or without anchor). And the relatedness score, $\mathrm{rel}(P_i, P_j)$ is computed as follows:

$$\mathrm{rel}(P_i, P_j) = \frac{\log(\max(|I|, |J|)) - \log(|I \cap J|)}{\log(|W|) - \log(\min(|I|, |J|))} \tag{3}$$

where I and J are the set of all pages that link to P_i, P_j, respectively, and W is total number of pages in Wikipedia.

Type and amount of information Wikipedia pages contain lots of useful information, and these pages are often much cleaner than regular web-pages. However, Wikipedia pages also include some information which is not useful in our analysis. We observe that a Wikipedia page usually contains lots of noisy words such as stop words, unimportant verbs, and adjectives, which are not much useful in hashtag recommendation. We therefore extract entities from a page resulted by querying an entity. These entities are semantically related to the original entity in the tweet as most of the entities in a Wikipedia page describe its primary/original entity due to the inherent nature of Wikipedia. We next extract the anchor texts present in a Wikipedia page as anchor texts often represent important concepts or entities. We also extract the category information present at the bottom of the pages as they contain similar subjects as the primary entity. Moreover, Wikipedia pages contain lots of information, and it is more than the required for the tag recommendation. We therefore select only the top three paragraphs from Wikipedia pages.

3.1.2 Web extraction

To build an effective and accurate recommendation system, a significant amount of tweet information is required. Due to limited information and entities present in tweets, the method employed above (i.e., Wikipedia extraction) to extract related information is not sufficient. Wikipedia extraction may not be able to capture the context of the complete tweet. We therefore extract the contextual related information for the tweet from the Web. As we extract streaming tweets related to news events, our data comprise of the latest news events. Ample amount of information related to a news event is also available from different news media sites and web-pages.

To obtain context information for a tweet, we submit the tweet as a query to The Guardian news search engine.[3] The Guardian is one of the most relevant and widely used sources for daily news events. While querying the search engine using time filter, highly related news documents created recently appears at the top of search results. We extract the first ten documents as these first ten documents are highly related to the tweet. We further filter these extracted documents by measuring the cosine similarity between the headline of documents and the tweet. For the tweet, if a document headline shows the similarity less than a threshold (in our experiment, we empirically set threshold as 0.2), we discard that document. One of the reasons is that headline precisely captures the essence of the whole news article—the higher similarity between the document headline and the tweet, the more similar they are. Including more number of documents adds less relevant and noisy information. We extract the entities and keywords[4] from these un-structured pages.

3.1.3 Word-embedding-based extraction

We extract keywords that are semantically related to a tweet using word-embeddings. There are several keywords which are semantically related and co-occur in tweets but may not appear when we perform Wikipedia and Web extraction. We therefore use word-embeddings to find the semantically related words to a tweet using Word2vec most similar function [30]. We train our Word2vec model using 17.6 million news tweets collected from Twitter. Word2vec model creates word-embeddings by generating vector space from the tweet corpus where each word in the corpus is assigned to a vector in the space, and each of these vectors has a dimensionality of 300. Given a tweet, we extract the top-20 words for each term present in the tweet using Word2vec most similar function. We include these words in extrinsic features of a tweet. For a tweet, the information collected from all the above-mentioned external sources is termed as 'external document,' 'extrinsic document,' or 'document.' In the rest of the paper, we use these three terms interchangeably.

3.2 Feature selection and processing

Due to feature extraction from multiple external sources, many redundant and noisy features are generated. To clean the data, standard text processing techniques such as stop words removal, stemming, lemmatization are used. We remove punctuations, numerical characters, URLs, special characters and duplicate words from extrinsic documents. We next remove the words that are not in English and perform spelling correction step. Further, we notice that users often use pronouns to refer nouns or noun phrases in their tweets. While using different hashtag recommendation methods, these nouns are important to determine similar tweets. For example, a user mentions 'Trump' once in his tweet and refers to it using pronouns multiple times. Replacing pronouns with nouns will increase the similarity between tweets and thereby

[3] https://open-platform.theguardian.com/explore/.

[4] http://newspaper.readthedocs.io/en/latest/.

recommend better hashtags. We use Bart [41], a modular toolkit for coreference resolution. Next, we use part-of-speech (POS) tagging [29] to get important words by removing trivial words such as 'usually,' 'could,' 'where,' etc. The tagger assigns a POS tag to each word of the given text, such as adverb, modal, particle, etc. Adverbs are represented by 'RB,' modals are represented by 'MD,' etc. We remove adverbs, symbols, wh-pronouns, wh-adverbs, particles, and modals from the collected extrinsic document. Finally, we remove all features with sparsity more than 0.99 as it helps to prevent overfitting.

3.3 Candidate hashtag generation

In this section, we describe the different hashtag recommendation methods used to generate candidate hashtags.

3.3.1 Similarity using term-based method

A tweet has tweet content and extrinsic document (or document) extracted from external sources. We first discuss the hashtag recommendation based on tweet content similarity and then describe the hashtag recommendation based on document similarity.

It is studied that similar tweets most likely share similar hashtags [22,47]. Zangerle et al. [47] introduced a hashtag recommendation system that retrieves a set of existing tweets similar to a given tweet. It then recommends hashtags to a given tweet from similar existing tweets. Similarly, we also recommend suitable hashtags to a given tweet based on existing similar tweets. TF-IDF and cosine similarity methods [34] are employed to get similar tweets. We rank the tweets based on their similarities and recommend hashtags from the most similar tweets.

Extrinsic document for a tweet is a collection of features gathered from external sources using tweet content. As we create a larger and semantically related document for the tweet using its internal features (or tweet content), it can capture more contextual information of the tweet which was hindered due to its character limit. As extrinsic document contains most of the related keywords of the tweet, it can capture the similarity better than just tweet similarity. We therefore extend the tweet similarity function to document similarity. Similar to tweet similarity, TF-IDF and cosine similarity are applied to documents to get similar documents for a given document. A document belonging to a tweet share the same hashtags as the tweet. We rank documents based on their similarities and select hashtags from tweets that contain the most similar documents as recommended hashtags for the given tweet.

3.3.2 Similarity using word-embeddings

The method described in the last section aims to recommend hashtags based on text content present in tweets. While finding similar tweets using TF-IDF and cosine similarity, we may miss the semantic information that is not readily available in tweets. We therefore use word-embeddings to compute the semantic similarity between tweets. We use word-embeddings-based Doc2vec model [23] to get the distributed representation of tweets. Each tweet in the semantic space is represented as a document vector or paragraph vector. The vector representation is learned using a neural network, which is trained using stochastic gradient descent where the gradient is acquired via backpropagation [37]. Semantic similarity between two vectors generated using Doc2vec is computed using cosine similarity. We first train Doc2vec model using our tweet corpus and then for a given tweet, we find the most similar tweets. We rank these tweets based on their similarities and select hashtags from the most similar tweets. Similarly, we apply the above Doc2vec-based similarity method for the external document of a tweet to select hashtags from the most similar external documents.

3.3.3 Language translation model

One can suggest hashtags for a tweet by merely looking into the content of the tweet such as annotating a tweet with entities present in it. However, this is not sufficient because tweet content and hashtag have diverse vocabularies, which is referred to as vocabulary gap [7,25]. Vocabulary gap is mainly created due to the following two reasons: (1) some entities that appear in the tweet can annotate a tweet but these entities are not statistically significant; and (2) some hashtags do not appear in the tweet. We can reduce the vocabulary gap by considering both the tweet and hashtags as parallel summaries of a resource in two different languages. We say that entities and hashtags present in a news tweet are two different representation of the tweet. It means that tweet and hashtag both want to convey the same meaning but both of them have different representations. For example, a tweet usually consists of a few sentences with a maximum length of 280 characters but hashtags are a few words with hash symbols. Language translation model (or LT model) forms a strong relationship between entities and hashtags with the support of other existing tweets. This entity and hashtag information is used to recommend the hashtags for a tweet.

In order to perform this task, LT model estimates the translation probabilities between entities and hashtags in tweets. Consider a tweet has N_e entities. The recommendation score (s) for a hashtag (h_t) can be computed as follows:

$$s(h_t) = \sum_{e_i \in N_e} P(h_t|e_i) \tag{4}$$

where $P(h_t|e_i)$ is the conditional probability that is computed by the number of times a hashtag h_t annotate a tweet t containing an entity e_i with respect to the frequency of e_i. We rank the hashtags based on their recommendation scores and select the top-ranked hashtags.

3.3.4 Topic modeling

Topic models are statistical models that are used to discover the important topics or hidden semantic structures from a large text corpus [2]. There are many topic modeling techniques such as PLSA (Probabilistic Latent Semantic Analysis), LDA (Latent Dirichlet Allocation), and TNG (Topical N-grams). Among these techniques, LDA is one of the most widely used technique and it is also used to recommend hashtags for tweets [14]. LDA is a generative topic model, which represents each document (or tweet) as a random mixture of latent topics with definite probabilities. Each topic is characterized by a distribution over words. The terms that usually appear together are placed under the same topic with high probabilities. Document-topic distribution (θ_{dj}) is computed using Gibbs sampling as follows:

$$\theta_{dj} = \frac{N_{dj}^{DT} + \alpha}{\sum_{k=1}^{T} N_{dk}^{DT} + T\alpha} \tag{5}$$

where D, T are the number of documents, topics, respectively. N_{dj}^{DT} is number of times topic j assigned to document d. α is a smoothing constant.

Term-topic distribution (ϕ_{ij}) is also estimated using Gibbs sampling as follows:

$$\phi_{ij} = \frac{N_{ij}^{WT} + \beta}{\sum_{k=1}^{W} N_{kj}^{WT} + W\beta} \tag{6}$$

where W and T are the number of terms, topics, respectively. Nij^{WT} is the number of times word i assigned to document j, and β is a smoothing constant. We train the model with parameters $\alpha = 0.1$, $\beta = 0.1$, and $T = 200$. To select the ideal number of topics T, we use perplexity measure. Perplexity is used to determine the performance of topic models [36]. It is a decreasing function of the log likelihood of the test tweets. So, a better model has a lower perplexity. We compute the rate of perplexity change on 10% test tweets. The point where the rate of perplexity no longer falls significantly with an increase in the number of topics is used as the ideal number of topics. Gibbs sampling method integrates these two assignments and updates the topic assignment until convergence.

We train the LDA model using 90% of the collected tweets and use the same model for the remaining test tweets. We again use Gibbs sampling to determine the document-topic distribution of a test tweet. Term-topic distribution is the same as computed during the training of the LDA model. After determining the document-topic distribution for a test tweet, we recommend top keywords from it as recommended hashtags based on the topic-term count.

3.4 User influence score computation

Twitter is one of the social network platforms where a few influential or popular users such as celebrities have a higher influence on the usage of hashtags by other users in Twitter community. Influential users are connected to a large number of other users. Their tweets have a higher exposure compare to an ordinary user [27,35] and can affect the hashtag adoption of connected users. According to a popular social media journalist,[5] influential social media users or leaders include hashtags in their tweets and they attract their connected users to promote their tweets. So, tweets created by popular users with a higher number of followers have better reach in the network. Followers often use the same hashtags that are created by the influential user whom they follow. Let us consider the following example:

There are two users u1 and u2, where user u1 is a celebrity having a verified status account and a higher number of followers than user u2. If user u1 tweets about an event with a hashtag (e.g., #FilmfareAwards) and user u2 tweets about the same event with a different hashtags (e.g., #FFAwards), then it is highly likely that the tweet by user u1 would have a higher reach. One of the reasons is that user u1 has more number of followers and there is a high probability that these followers would start tweeting or re-tweeting about the event with the same hashtag used by u1.

As it can be observed from the above example that hashtags used by influential users are more likely to be used by Twitter community. Verified users are one type of influential users. According to Twitter,[6] a user with public interest is given a verified status. When such users create a tweet, they get high visibility as their tweet is spread to a large audience. These users' tweets have higher reach in Twitter community and their hashtags are used by a large number of other Twitter users. Recommending hashtags used by verified users can help to avoid hashtag duplication/redundancy, which can lead to effective tweet categorization

[5] https://www.hashtags.org/how-to/marketing-how-to/twitter-marketing-101-what-does-it-take-for-a-hashtag-to-trend/.

[6] https://help.twitter.com/en/managing-your-account/about-twitter-verified-accounts.

and retrieval. This can also help users to join a larger conversation. We therefore give more weights to the hashtags used by such users while recommending hashtags for a tweet. For each hashtag, we compute user influence score (UInfh) as follows:

$$\text{UInf}_h = \alpha * U_v + \beta * \log(U_f) \tag{7}$$

where U_v is the average verified status of all the users who have used the hashtag h. Verified status for a user is set to 1 if the user is verified and 0 otherwise. U_f is the average of followers count of all the users who have used the hashtag h. α and β are two multiplicative constants. We tried using different values of α and β. In our experiment, we found that our system gives better results with parameters $\alpha = 1$ and $\beta = 0.25$. One of the reasons is that verified status is given to user accounts which are of public interest. These users have a diverse reach and their tweets influence people all over the world even if they do not have a large number of followers. So, it is important to give priority to the hashtags used by verified users by assigning a higher weight α. The above equation shows that if a hashtag is used by a verified user with a higher number of follower count, user influence score for the hashtag will be higher compared to other hashtags which are not used by such users.

3.5 Candidate hashtag recommendation

In Sect. 3.3, we presented the methods to generate the candidate hashtags. Each of these methods is capable enough to recommend hashtags for a given tweet. While using standalone recommendation method, we may not achieve better performance as compared to the combination of all the methods. We therefore aggregate the hashtags generated by all the methods and rank them to get the advantage of these methods. To this end, we frame our problem of hashtag recommendation as a learning-to-rank problem.

3.5.1 Pairwise learning-to-rank model

We use a pairwise learning-to-rank method to recommend the top-ranked hashtags from the candidate hashtags. For a given tweet t, we generate the candidate hashtags using hashtag generation techniques. Each candidate hashtag (h_t) is represented using a vector of features. We use two sets of features to represent a hashtag vector. The first set contains six features where each feature corresponds to a candidate hashtag selection scheme. For a given candidate hashtag, each of its features in the first set of feature vector is set to 1 if the corresponding hashtag selection method recommends that hashtag (h_t) for the given tweet. The second set contains a single feature named user influence score. For the candidate hashtag set of a tweet, we create a matrix of feature vectors where each row represents a hashtag. After generating the matrix, we train the RankSVM model to rank these hashtags.

3.5.2 Training learning-to-rank model

We divide all the candidate hashtags of a tweet t into two sets: (a) positive set (h_t^+), and (b) negative set (h_t^-). (h_t^+) is the collection of those hashtags that are actually used to annotate the tweet and (h_t^-) are those hashtags that are not used to annotate the tweet. We formulate our ranking problem as a classification problem where $< h^+, h^- >$ is a positive instance for the classification and $< h^-, h^+ >$ is a negative instance for the classification. We represent each pair by the difference between feature vectors of two hashtags. Support Vector Machine classifier is trained using these settings for all the tweets in training set. For a new tweet, every hashtag in the candidate hashtag set is paired with every other hashtag in the same set. Each pair of hashtags is represented by the vector difference of the hashtags in the same pair. Each pair is then passed to the classifier that predicts it as a positive or negative instance. To better understand this, let us consider H_T is a set of candidate hashtags for a new tweet T. We can represent H_T as follows: $H_T = h_1, h_2, \ldots, h_t$. Now, we compute the recommendation score $\mathrm{RS}(h_t)$ for a hashtag h_t.

$$\mathrm{RS}(h_t) = \sum_{h_s \in H_T, h_t \neq h_s} I(h_t, h_s) \tag{8}$$

where $I(h_t, h_s) = 1$ if $< h_t, h_s >$ is classified as positive instance else 0. We rank the candidate hashtags based on the recommendation score.

4 Experimental evaluations

In this section, we first give the details about our dataset, dataset preprocessing steps, and then evaluate our proposed methods.

4.1 Dataset collection

In order to validate the effectiveness of our proposed method, we collect data from Twitter using Twitter Streaming API.[7] To get the news tweets, we use top trending news hashtags guided by hashtags.org.[8] Some of the common keywords used to query streaming API are NEWS, CNN, FoxNews, BBC, etc. In total, we extract 30 million news tweets in the month of March 2018. Among 30 million total tweets, there are 12.4 million retweets and 17.6 million tweets. Further, there are 14.7% tweets, which contain URLs of external sources. We also extract the data from external sources using these URLs. Table 1 presents the dataset statistics of collected 17.6 million tweets. In Table 1, term ‘entity’ indicates ‘named entity’ that is obtained using Stanford CoreNLP [29]. Figs. 2 and 3 show the entity distribution and hashtag distribution of tweets, respectively.

To create our dataset, we consider tweets with at least two hashtags so that we can validate our proposed method. There are two million tweets with at least two hashtags. We further remove tweets that are noisy and do not provide sufficient information about an event. We say that a tweet is informative if it contains at least two entities. Tweets without any entity or less than two entities do not convey sufficient information about a news event. It is difficult to get much context from non-informative tweets while querying Wikipedia and Web. Though our proposed method works for all kinds of tweets, we remove the tweets that are not informative for a better recommendation.

[7] https://developer.twitter.com/en/docs/tutorials/consuming-streaming-data.html.

[8] https://www.hashtags.org/.

Table 1 Dataset statistics

Type of tweets	Percentage of tweets (%)	Quantity of tweets (in millions)
Number of tweets with no hashtags	84.1	14.8
Number of tweets with at least one hashtag	15.9	2.8
Number of tweets with no entity	20.2	3.5
Number of tweets with at least one entity	79.8	14.1
Number of tweets with no hyperlinks	85.3	15.0
Number of tweets with at least one hyperlink	14.7	2.6

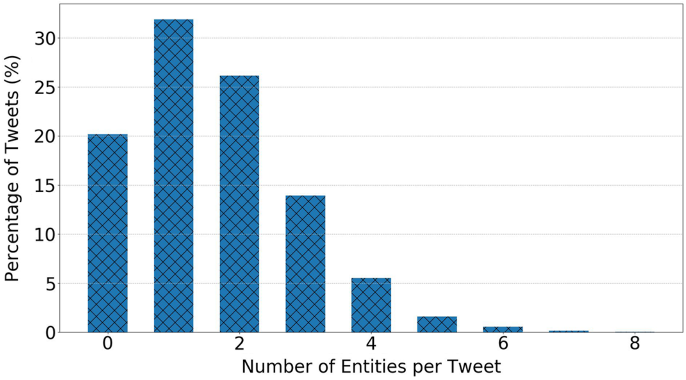


Fig. 2 Entity distribution

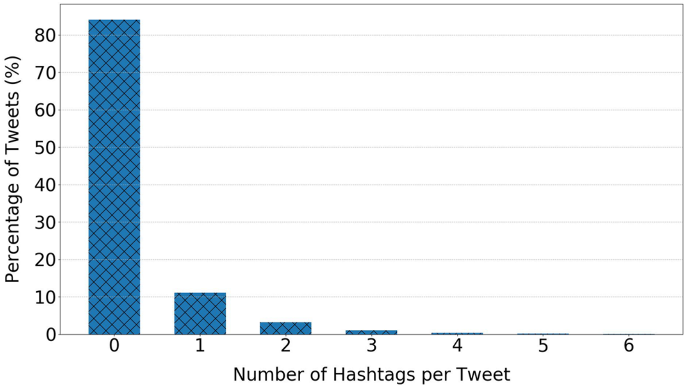


Fig. 3 Hashtag distribution

4.2 Dataset pre-processing

Due to the inherent nature of the microblogging platform, the collected data are very noisy. Users often use informal and specialized language, which create a great challenge to build a good quality recommender system. We therefore perform pre-processing to clean our dataset. Tweets often contain emojis, URLs, emails, pictographs, etc. These components are not much useful for hashtag recommendation and are removed from tweets. We further clean the dataset using the following steps:

4.2.1 Removing trailing hashtags

Hashtags are usually placed at the end of a tweet. However, there is no restriction on placing the hashtag in a tweet. Many users use hash symbols for a word, which is part of the tweet. We therefore remove all the trailing hashtags but remove only the '#' symbol from a hashtag if it is part of a tweet. For example, 'Trump clashes with #CNN reporter, suspends WhiteHouse pass #TrumpPressConference #Trump #WhiteHousePressConference.' In this example, #CNN is a part of the tweet sentence. We remove only the '#' symbol from #CNN but remove all the trailing hashtags such as #TrumpPressConference #Trump #WhiteHouse-PressConference. The final processed tweet is as follows: Trump clashes with CNN reporter, suspends WhiteHouse pass.

To remove trailing hashtags, we use part-of-speech (POS) tagging. We say that if a preceding word of a hashtag is conjunction, verb or preposition, then the hashtag is a part of the sentence. If a hashtag word is either first word of the sentence or its preceding word is one of following POS tags: 'SCONJ,' 'PART,' 'DET,' 'CCONJ,' 'CONJ,' 'AUX,' 'ADP,' 'ADJ,' 'VERB,' 'INTJ,' 'PRON,' 'ADV,' then the hashtag word is part of the sentence and only # symbol from the hashtag word is removed. Otherwise, the hashtag is not the part of the sentence and complete hashtag word is removed.

4.2.2 Replacing user mentions with user actual names

Users often write user mentions in tweets instead of writing their actual name. For example, consider the following example tweet: @AbeShinzo is meeting @narendramodi tomorrow in the city of Yamanashi Japan. In this tweet, instead of using the name of prime ministers Shinzo Abe and Narendra Modi, a user decides to use user mentions. Replacing these user mentions with their actual name may yield better similarity while recommending hashtags. We use the screen name attribute provided by Twitter and replace user mentions with the corresponding actual names.

4.2.3 Word splitting and contractions

Users often use a combination of multiple words as a hashtag such as #realifequotes, #independenceday, #healthylife, etc. Although 'independenceday' and 'independence day' convey the same meaning, they would be treated as different tokens by a hashtag recommendation system. If these words are the part of a tweet then they create a challenge for a hashtag recommendation system while finding the similarity between tweets. We therefore split these kinds of big phrases (or hashtags) into smaller meaningful words using word segment module.[9] We also resolve the contraction and slang problem appearing in a tweet such as 'yall,' 'gotta,' 'ima,' and 'youre.' Using contraction module,[10] these contractions can be fixed. For example, 'yall,' 'gotta,' 'ima,' and 'youre' are converted to 'you all,' 'got to,' 'i am going to,' and 'you are,' respectively.

4.2.4 Expanding abbreviated terms

In Twitter, users often use abbreviations to convey the meaning in short form due to the limitation of tweet length and ease of writing abbreviations. However, these informal abbreviations are not uniformly used by all the users, which leads to low similarity between tweets while recommending the hashtags. For example, 'gratz,' 'maga,' and 'ianap' are commonly used abbreviations, which indicate 'Congratulations,' 'Make America Great Again,' and 'I am not a photographer,' respectively. To this end, we create a custom dictionary using Webopedia[11] and GitHub.[12] We replace all the key abbreviations using their values or expanded forms from the dictionary.

4.2.5 Entity resolution

Due to the inherent nature of social media, users do not use named entities or entities uniformly. Entities are very important for hashtag recommendation as many entities are also used as hashtags to annotate a tweet. For example, let us consider the following tweet: 'Donald Trump is visiting Singapore tomorrow to meet Kim Jong-un.' The same news tweet is tweeted differently from different users. Instead of using uniform entities many users use Kim, Kim Jong, Trump, or Donald J. Trump. We can resolve this by providing a uniform name for an entity. To this end, we use TAGME [12] which can annotate entity with its correct name. It uses Wikipedia data to perform this task and link each entity present in a tweet to its correct Wikipedia page. We use the title of the page to rename an entity to its correct form. In addition to all the above pre-processing steps, we also perform stop-word removal, stemming and lemmatization. Table 2 presents the summary of pre-processed informative tweets.

[9] https://github.com/jchook/wordseg.

[10] https://github.com/kootenpv/contractions.

[11] https://www.webopedia.com/quick_ref/Twitter_Dictionary_Guide.asp

[12] https://gist.github.com/Zenexer/af4dd767338d6c6ba662.

Table 2 Pre-processed Tweets

Dataset attributes	Values
Number of tweets	329,369
Number of hashtags	785,400
Number of unique hashtags	85,216
Average number of hashtags per tweet	2.38

4.3 Method

In this section, we give details of the methods that are compared with our proposed method (i.e., RankSVM with User Influence). We evaluate our proposed method against four baseline methods, two word-embedding-based methods, and one RankSVM-based method.

4.3.1 Term-based tweet similarity method

The method recommends hashtags to a given tweet from similar existing tweets. We use TF-IDF and cosine similarity to retrieve similar tweets. We have given the details of the method in Sect. 3.3.1.

4.3.2 Term-based document similarity method

This method recommends hashtags to a given tweet based on extrinsic documents that are similar to the extrinsic document of a given tweet. The detailed explanation is given in Sect. 3.3.1.

4.3.3 LT model

We implement our LT model, which is inspired by the word trigger method for suggesting tags [25]. We say that entities and hashtags present in a news tweet are two different representation of the tweet. For a given tweet, the model recommends hashtags that are highly associated with entities present in the tweet. We have described the method in Sect. 3.3.3.

4.3.4 LDA

LDA is one of the most popular techniques that have been used to recommend hashtags. LDA uses topic distribution to recommend hashtags for a given tweet. We use Collapsed Gibbs Sampling method to implement LDA [15]. We have provided the details of the method in Sect. 3.3.4.

4.3.5 Word-embeddings-based tweet similarity method

Word-embeddings-based method is used to compute the semantic similarity between tweets. We use word-embeddings-based Doc2vec model [23] to get the distributed representation of tweets and compute semantic similarity between tweets using cosine similarity. For a given tweet, hashtags are recommended from most similar tweets. The details of the method are provided in Sect. 3.3.2.

4.3.6 Word-embeddings-based document similarity method

We extend the word-embeddings-based tweet similarity to word-embeddings-based document similarity to recommend hashtags from most similar documents as described in Sect. 3.3.2.

4.3.7 RankSVM

RankSVM is a widely used learning-to-rank method [8], which aggregates the above-mentioned six methods to the get advantage of all these methods. RankSVM employs pairwise learning-to-rank strategy to perform this task as mention in Sect. 3.5.

4.4 Evaluation metrics

To evaluate the performance of our proposed method, we use two evaluation metrics, namely precision and recall. These are the most common evaluation metrics used to evaluate the performance of recommender systems. We say that recommended hashtags are retrieved hashtags (r_t), and relevant hashtags (r_l) are the hashtags that are originally present in the tweet (i.e., ground-truth hashtags). So according to the definition of precision, it is the fraction of retrieved hashtags that are relevant to the tweet. Recall is the fraction of the relevant hashtags that are successfully retrieved. We formally define precision and recall at k as follows:

$$\text{Precision@}k = \frac{r_t \cap r_l}{r_t} \tag{9}$$

$$\text{Recall@}k = \frac{r_t \cap r_l}{r_l} \tag{10}$$

where k is the number of top-ranked recommended hashtags. From the above definition, we can also say that precision is the number of common hashtags between actual hashtags present in the tweet and recommended hashtags divided by the number of recommended hashtags. Recall is the number of common hashtags between actual hashtags and recommended hashtags divided by the actual number of hashtags present in the tweet.

4.5 Effectiveness of the methods

Figures 4 and 5 show the performance of all the hashtag recommendation methods including proposed RankSVM-based method. These figures present the results for $k = 2, 3, 5$, where k is the number of top-ranked recommended hashtags. We use different values of k to investigate the performance of hashtag recommendation methods with a varying number of recommended hashtags. As we increase the value of k, precision decreases and recall increases for all the methods. All the methods show the best precision at $k = 2$. One of the reasons for this is that the average number of hashtags for a tweet is two. While increasing the recommended hashtags, the number of common hashtags between actual hashtags present in the tweet and recommended hashtags does not increase much, which leads to a lower precision. On the other hand, it can be inferred from Equation 10 that recall increases as the number of recommended hashtag increases. In Figs. 4 and 5, 'Term based TweetSim,' 'Term based DocSim,' 'Word-embeddings based TweetSim,' 'Word-embeddings based DocSim,' and 'RankSVM w/ User Influence' indicate 'Term based Tweet Similarity Method,' 'Term based Document Similarity Method,' 'Word-embeddings based Tweet Similarity Method,'

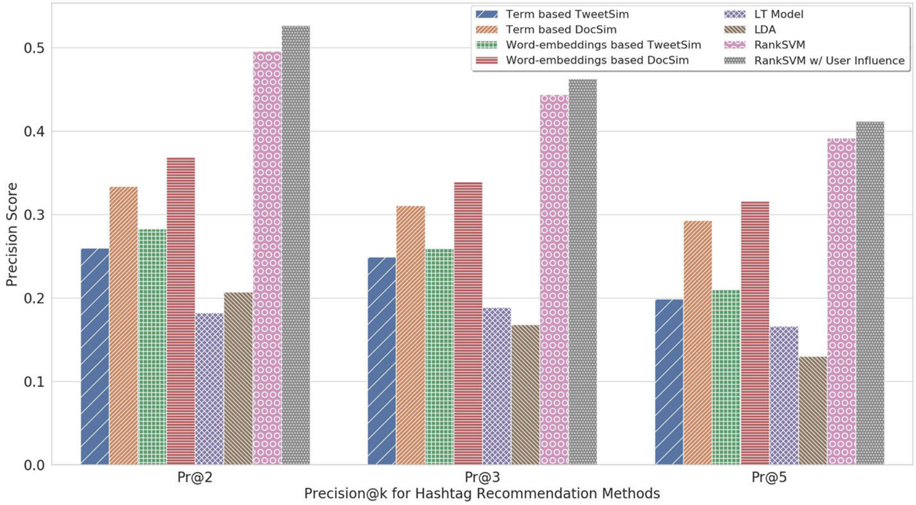


Fig. 4 Precision

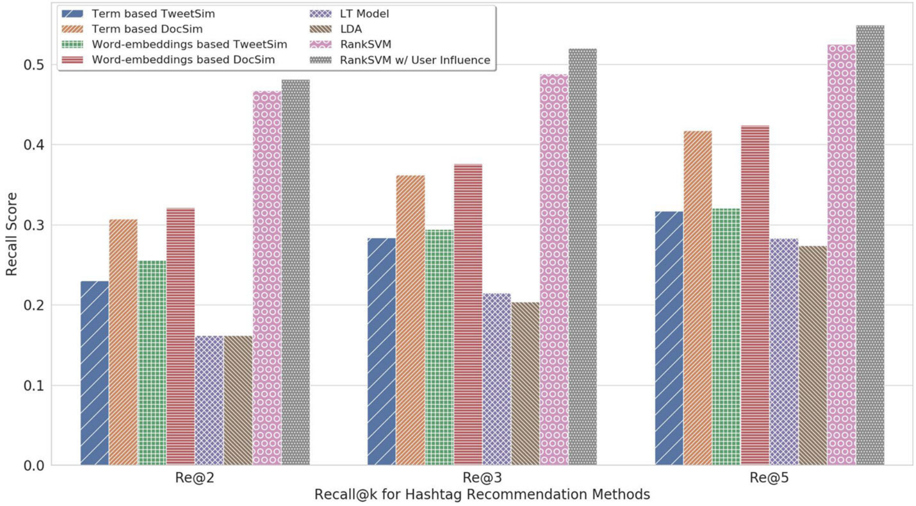


Fig. 5 Recall

‘Word-embeddings based Document Similarity Method,’ and ‘RankSVM with User Influence,’ respectively.

It can be observed from Figs. 4 and 5 that LDA and LT models have lower precision and recall as compared to other methods. For all the three values of k, these methods do not perform well. One of the reasons for the poor performance of these methods is that these methods do not take the tweet context into account. LDA recommends very general topical words as hashtags. It first creates a fixed number of topics and keeps recommending hashtags from these topics that may not be an accurate recommendation for a given specific tweet. Similarly, LT model also does not consider tweet context into account and recommends hashtags that are highly associated with entities present in the tweet. For example, consider an entity *e* is associated with a hashtag *h* with high probability. For a new tweet *t* containing the same entity *e*, LT model would recommend the high probable hashtag *h* without considering the context of the new tweet.

Similarity using term-based methods perform better than LDA and LT model as similarity-based methods take tweet context into account. We further notice that the term similarity method based on extrinsic document performs better than the term similarity method based on tweet content. One of the reasons for this is that short tweets have very limited text content and it is difficult to find similar tweets based on their contents. On the other hand, an extrinsic document has enough content to find the most similar tweets based on the extrinsic document similarity.

Tweets usually suffer from semantic gap problem due to their limited length. Using word-embedding-based method, we can reduce the semantic gap problem, and thereby increase the performance of the hashtag recommendation system. As can be observed from Figs. 4 and 5, word-embedding method using tweet content achieves better precision and recall compared to term-based method using tweet content. Similarly, word-embedding method based on extrinsic document also performs better than term-based method using extrinsic document. Further, word-embedding method using extrinsic document performs better than word-embedding method using tweet content. One of the reasons is that extrinsic document has enough features from external sources. These extrinsic features from external sources add more contextual information of tweets and include sufficient non-trivial semantically related features from external sources.

Further, RankSVM performs better than all other existing individual methods. RankSVM is trained in such a way that it ranks the set of candidate hashtags from all the methods according to their relevance. It combines all the similarity-based methods, LDA and LT models. Term-based similarity method and word-embedding-based similarity method capture the context and semantics of tweets. LDA model captures the general topics of tweets and recommends the relevant hashtags. LT model captures the entity-hashtag relationship information for every entity in a tweet and recommends the relevant hashtags. Since all these methods are combined using RankSVM, it recommends more accurate and relevant hashtags for a tweet. Moreover, the largest improvement comes when RankSVM is combined with user influence. Since user influence feature gives weights to popular hashtags, the RankSVM learns to give preference to the popular hashtags which leads to a better performance.

5 Conclusion

In this paper, we proposed a novel method to recommend hashtags for tweets using lexical, topical, semantic and user influence features. To eliminate the problem of limited word co-occurrence and data sparsity in short tweets, the proposed method exploited the knowledge from extrinsic sources. External knowledge from extrinsic sources bridges the semantic gap between tweets and related hashtags. The paper proposed different candidate hashtag generation techniques based on lexical, topical and semantic features of tweets. Unlike existing methods that rely on the similarity of tweets, the paper proposed word-embedding-based hashtag recommendation method that reduces the semantic gap between tweets and hashtags. To recommend more relevant hashtags, the paper aggregated multiple hashtag generation methods using learning-to-rank. Experimental results showed that the proposed method achieves a significant improvement in precision and recall compared to existing hashtag recommendation methods.

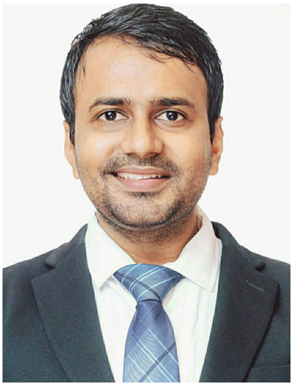
References

1. Atefeh F, Khreich W (2015) A survey of techniques for event detection in twitter. Comput Intell 31(1):132–164
2. Blei DM, Ng AY, Jordan MI (2003) Latent dirichlet allocation. J Mach Learn Res 3:993–1022
3. Brooks C.H, Montanez N (2006) Improved annotation of the blogosphere via autotagging and hierarchical clustering. In: Proceedings of the 15th international conference on World Wide Web. pp 625–632. ACM
4. Carpenter JP, Krutka DG (2014) How and why educators use twitter: a survey of the field. J Res Technol Educ 46(4):414–434
5. Chang HC (2010) A new perspective on twitter hashtag use: diffusion of innovation theory. Proc Assoc Inf Sci Technol 47(1):1–4
6. Davidov D, Tsur O, Rappoport A (2010) Enhanced sentiment learning using twitter hashtags and smileys. In: Proceedings of the 23rd international conference on computational linguistics: posters. pp 241–249. Association for Computational Linguistics
7. Ding Z, Qiu X, Zhang Q, Huang X (2013) Learning topical translation model for microblog hashtag suggestion. In: Twenty-third international joint conference on artificial intelligence. pp 2078–2084
8. Duan Y, Jiang L, Qin T, Zhou M, Shum H.Y (2010) An empirical study on learning to rank of tweets. In: Proceedings of the 23rd international conference on computational linguistics. pp 295–303. Association for Computational Linguistics
9. Efron M (2010) Hashtag retrieval in a microblogging environment. In: Proceedings of the 33rd international ACM SIGIR conference on research and development in information retrieval. pp 787–788. ACM
10. Feng W, Wang J (2012) Incorporating heterogeneous information for personalized tag recommendation in social tagging systems. In: Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining. pp 1276–1284. ACM
11. Ferragina P, Piccinno F, Santoro R (2015) On analyzing hashtags in twitter. In: International conference on Web and Social Media (ICWSM). AAAI Press, pp 110–119
12. Ferragina P, Scaiella U (2010) Tagme: on-the-fly annotation of short text fragments (by wikipedia entities). In: Proceedings of the 19th ACM international conference on information and knowledge management. pp 1625–1628. ACM
13. Firth JR (1957) A synopsis of linguistic theory, 1930–1955. Studies in linguistic analysis
14. Godin F, Slavkovikj V, De Neve W, Schrauwen B, Van de Walle R (2013) Using topic models for twitter hashtag recommendation. In: Proceedings of the 22nd international conference on World Wide Web. pp 593–596. ACM
15. Griffiths TL, Steyvers M (2004) Finding scientific topics. Proc Nat Acad Sci 101(suppl 1):5228–5235
16. Guan Z, Bu J, Mei Q, Chen C, Wang C (2009) Personalized tag recommendation using graph-based ranking on multi-type interrelated objects. In: Proceedings of the 32nd international ACM SIGIR conference on research and development in information retrieval. pp 540–547. ACM
17. Harris ZS (1954) Distributional structure. Word 10(2–3):146–162
18. Hong L, Convertino G, Chi EH (2011) Language matters in twitter: a large scale study. In: ICWSM
19. Hu X, Sun N, Zhang C, Chua T.S (2009) Exploiting internal and external semantics for the clustering of short texts using world knowledge. In: Proceedings of the 18th ACM conference on information and knowledge management. pp 919–928. ACM
20. Kalloubi F, Nfaoui EH, El Beqqali O (2017) Harnessing semantic features for large-scale content-based hashtag recommendations on microblogging platforms. Int J Seman Web Inf Syst (IJSWIS) 13(1):63–81
21. Kiritchenko S, Zhu X, Mohammad SM (2014) Sentiment analysis of short informal texts. J Artif Intell Res 50:723–762
22. Kywe SM, Hoang TA, Lim EP, Zhu F (2012) On recommending hashtags in twitter networks. In: International conference on social informatics. pp 337–350. Springer
23. Le Q, Mikolov T (2014) Distributed representations of sentences and documents. In: International conference on machine learning. pp 1188–1196
24. Liang H, Xu Y, Li Y, Nayak R, Tao X (2010) Connecting users and items with weighted tags for personalized item recommendations. In: Proceedings of the 21st ACM conference on Hypertext and hypermedia. pp 51–60. ACM
25. Liu Z, Chen X, Sun M (2011) A simple word trigger method for social tag suggestion. In: Proceedings of the conference on empirical methods in natural language processing. pp 1577–1588. Association for Computational Linguistics
26. Ma Z, Sun A, Cong G (2012) Will this# hashtag be popular tomorrow? In: Proceedings of the 35th international ACM SIGIR conference on research and development in information retrieval. pp 1173–1174. ACM

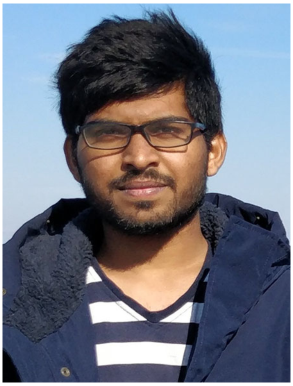
27. Ma Z, Sun A, Cong G (2013) On predicting the popularity of newly emerging hashtags in twitter. J Assoc Inf Sci Technol 64(7):1399–1410
28. Ma Z, Sun A, Yuan Q, Cong G (2014) Tagging your tweets: A probabilistic modeling of hashtag annotation in twitter. In: Proceedings of the 23rd ACM international conference on conference on information and knowledge management. pp 999–1008. ACM
29. Manning C, Surdeanu M, Bauer J, Finkel J, Bethard S, McClosky D (2014) The stanford corenlp natural language processing toolkit. In: Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations. pp 55–60
30. Mikolov T, Sutskever I, Chen K, Corrado G.S, Dean J (2013) Distributed representations of words and phrases and their compositionality. In: Advances in neural information processing systems. pp 3111–3119
31. Mishne G (2006) Autotag: a collaborative approach to automated tag assignment for weblog posts. In: Proceedings of the 15th international conference on World Wide Web. pp 953–954. ACM
32. Otsuka E, Wallace S.A, Chiu D (2014) Design and evaluation of a twitter hashtag recommendation system. In: Proceedings of the 18th international database engineering and applications symposium. pp 330–333. ACM
33. Pan J.Y, Yang H.J, Faloutsos C, Duygulu P (2004) Gcap: Graph-based automatic image captioning. In: Conference on computer vision and pattern recognition workshop, 2004. CVPRW'04. pp 146. IEEE
34. Ramos J et al (2003) Using tf-idf to determine word relevance in document queries. Proc First Inst Conf Mach Learn 242:133–142
35. Romero D.M, Meeder B, Kleinberg J (2011) Differences in the mechanics of information diffusion across topics: idioms, political hashtags, and complex contagion on twitter. In: Proceedings of the 20th international conference on World wide web. pp 695–704. ACM
36. Rosen-Zvi M, Griffiths T, Steyvers M, Smyth P (2004) The author-topic model for authors and documents. In: Proceedings of the 20th conference on uncertainty in artificial intelligence. pp 487–494. AUAI Press
37. Rumelhart DE, Hinton GE, Williams RJ (1986) Learning representations by back-propagating errors. nature 323(6088):533
38. Sedhai S, Sun A (2014) Hashtag recommendation for hyperlinked tweets. In: Proceedings of the 37th international ACM SIGIR conference on research and development in information retrieval. pp 831–834. ACM
39. She J, Chen L (2014) Tomoha: Topic model-based hashtag recommendation on twitter. In: Proceedings of the 23rd international conference on World Wide Web. pp 371–372. ACM
40. Tsur O, Rappoport A (2012) What's in a hashtag?: content based prediction of the spread of ideas in microblogging communities. In: Proceedings of the fifth ACM international conference on Web search and data mining. pp. 643–652. ACM
41. Versley Y, Ponzetto SP, Poesio M, Eidelman V, Jern A, Smith J, Yang X, Moschitti A (2008) Bart: A modular toolkit for coreference resolution. In: Proceedings of the 46th annual meeting of the association for computational linguistics on human language technologies: demo session. pp 9–12. Association for Computational Linguistics
42. Wang X, Wei F, Liu X, Zhou M, Zhang M (2011) 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. pp 1031–1040. ACM
43. Wang Y, Zheng B (2014) On macro and micro exploration of hashtag diffusion in twitter. In: 2014 IEEE/ACM international conference on advances in social networks analysis and mining (ASONAM). pp 285–288. IEEE
44. Wang Y, Qu J, Liu J, Chen J, Huang Y (2014) What to tag your microblog: hashtag recommendation based on topic analysis and collaborative filtering. In: Asia-Pacific web conference. pp 610–618. Springer
45. Xiao F, Noro T, Tokuda T (2012) News-topic oriented hashtag recommendation in twitter based on characteristic co-occurrence word detection. In: International conference on web engineering. pp 16–30. Springer
46. Yang H, Chua T.S, Wang S, Koh C.K (2003) Structured use of external knowledge for event-based open domain question answering. In: Proceedings of the 26th annual international ACM SIGIR conference on research and development in informaion retrieval. pp. 33–40. ACM
47. Zangerle E, Gassler W, Specht G (2011) Recommending#-tags in twitter. In: Proceedings of the workshop on semantic adaptive social web (SASWeb 2011). CEUR workshop proceedings. vol 730, pp 67–78
48. Zangerle E, Gassler W, Specht G (2013) On the impact of text similarity functions on hashtag recommendations in microblogging environments. Soc Netw Anal Min 3(4):889–898
49. Zhang Q, Gong Y, Sun X, Huang X (2014) Time-aware personalized hashtag recommendation on social media. In: Proceedings of the 25th international conference on computational linguistics: technical papers COLING 2014. pp 203–212

50. Zhao F, Zhu Y, Jin H, Yang LT (2016) A personalized hashtag recommendation approach using lda-based topic model in microblog environment. Future Gener Comput Syst 65:196–206

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Dr. Nagendra Kumar is an Assistant Professor in the Department of Computer Science and Engineering at IIT Indore, India. Prior to joining IIT Indore, he was a scientist at Institute for Infocomm Research (I2R), Agency for Science, Technology and Research (A*STAR), Singapore. He also worked at National University of Singapore in 2018 as a visiting research scholar, where he contributed in improving topic modeling algorithms and developed an interactive tweet analysis system. He obtained his PhD from IIT Hyderabad, in 2019. His research interests are in Natural Language Processing, Social Network Analysis, Deep Learning, Artificial Intelligence, Machine Learning, and Data Mining.



Eshwanth Baskaran works as a senior software engineer at Whatsbusy Inc. in the United States. He pursued his BTech in Computer Science and Engineering at Indian Institute of Technology (IIT) Hyderabad, India. His research work revolves around Social Networking Analysis, Natural Language Processing, and Information Retrieval. He is an active member in the open-source community.

Dr. Anand Konjengbam is a researcher at Shizuoka University, Hamamatsu, Japan. He received his PhD from the Indian Institute of Technology Hyderabad, India, and was an assistant project engineer at the Indian Institute of Technology Guwahati, India. His research interests include Data Mining, Natural Language Processing, Sentiment Analysis and Information Management.

Dr. Manish Singh is an Assistant Professor in the Computer Science and Engineering Department at Indian Institute of Technology (IIT) Hyderabad, India. He is a member of the Data and Informatics Group (DIg) in IIT Hyderabad. He obtained his PhD from the University of Michigan. He earned my Dual degree (BTech & MTech) in Computer Science from IIT Delhi. His main research areas are Data Mining, Information Retrieval, and Databases. He is currently working on Community Question-Answer Mining, Social Network Analysis, Opinion Mining, and Recommendation Systems.