Content-based Hashtag Recommendation Methods for Twitter

Dipesh Poudel
Earlham College
Richmond, Indiana
dpoudel18@earlham.edu

ABSTRACT

For Twitter, a hashtag recommendation system is an important tool to organize similar content together for topic categorization. Much research has been carried out on figuring out a new technique for hashtag recommendation, and very little research has been done on evaluating the performance of different existing models using the same dataset and the same evaluation metrics. This paper evaluates the performance of different content-based methods (Tweet similarity using hashtag frequency, Naïve Bayes model, and KNN-based cosine similarity) for hashtag recommendation using different evaluation metrics including *Hit Ratio*, a metric recently created for evaluating a hashtag recommendation system. The result shows that Naive Bayes outperforms other methods with an average accuracy score of 0.83.

KEYWORDS

hashtag, tag recommendation, recommender systems, content-based, natural language processing

1 INTRODUCTION

A hashtag is a word or a phrase starting with a hash (#) sign. Twitter, a popular micro-blogging platform, provides a hashtag feature to its users to categorize topics and give easy search experience to users who want to explore feed related to a particular topic or theme. Hashtags are also used for mass broadcasts during disasters or elections, and for brand promotion [7]. A hashtag recommendation system aims to recommend hashtags to a user relevant to the user's tweet during the time of posting. For instance, for a tweet that says "I am excited for Manchester United vs Arsenal match", a good recommender system should be able to recommend popular hashtags that are relevant to this tweet. In this particular case, hashtags like #EPL, #Soccer, and #PremierLeague are popular hashtags trending in Twitter which are relevent to this tweet.

Most of the researchers in this field are interested in coming up with new techniques for suggesting hashtags or proposing optimized solutions for existing models. The dataset, pre-processing techniques and evaluation metrics vary in each research which makes it difficult to compare and evaluate the performance of various existing models. This study addresses this issue and builds a hashtag recommender system using three common content-based methods: Tweet similarity using hashtag frequency, Naïve Bayes model, KNN using cosine similarity. For model training, all of the models uses the same dataset, and pre-processing techniques. Moreover, the previous studies have used classic evalutaion metrics like *Precision*, *Recall*, *F1 Score*, and *Hit Rate* for performance evaluation. Alongside these evaluation metrics, the project evaluates the performance of the model with *HitRatio*, a new evaluation metric proposed by Alsini et al. [1].

This paper is divided into multiple sections: overview of the related work, the design of the projects including framework, model design, dataset, and pre-processing technique, and last few sections talks about evaluation, results and conclusion.

2 RELATED WORK

This section covers the relevant work related to the project. Section 2.1 discusses past research related to Tweet similarity using TF-IDF, section 2.2 discusses research related to Naïve Bayes model, and section 2.3 discusses research related to KNN based model.

2.1 Tweet similarity based on TF-IDF

One of the content-based algorithms for hashtag recommendation is a Tweet Similarity method that uses the TF-IDF scheme. The term frequency measures the number of occurrences of a term within a given document. Inverse document frequency is calculated by taking the number of all documents within the index divided by the number of documents which has the searched term.

Zangerle et. al. [12] brings forwards the idea of recommending hashtags using three steps. Based on the user's tweet, a set of similar tweets are extracted with the help of similarity score calculated using TF-IDF (Term frequency - Inverse document frequency). Second, the commonly used hashtags in those similar tweets are selected to be candidate hashtags . Third, the candidate hashtags are ranked, and the top-k hashtags are recommended where k denotes the number of hashtags to be displayed to the user. The paper proposes three ranking methods: OverallPopularityRank (based on which candidate hashtags are more popular), Recommendation-PopularityRank (based on which candidate hashtags occurred the most), SimilarityRank (based on which hashtags are contained in the most similar tweet to the user's tweet). With a recall score of 45-50% during the evaluation of the described model, the authors concludes that this method is feasible for suggesting hashtags.

Kywe et al. [6] proposed a method of scoring candidate hashtag by taking both tweet similarity and user similarity with TF-IDF as a scoring method. Using TF-IDF, the most similar tweets and the most similar users are chosen. Then, the hashtags are choosen from most similar tweet and user, and assigned a ranking score based on frequency. The authors use *Hit Rate* as a evaluation metric, and the result shows that incorporating both user preferences and tweet content will produce better recommendation than just taking tweet content into account.

2.2 Naïve Bayes Model

Naïve Bayes is a machine learning algorithm that utilizes Bayes' Theorem together with a assumption that the attributes are conditionally independent.

Mazzia and Juett [5] discusses Naïve Bayes model of recommending hashtags using probabilistic approach based on Bayes's Theorem. Given the list of words in a tweet, the probability of using a particular hashtag is calculated using a model based on Bayes's Theorem. The hashtags with the highest probabilities are recommended to the user. For data pre-processing, they removed stopwords, punctuations, links, retweets, mentions, and non-english tweets. They also removed 'micro memes' tweets. Micro memes, as defined by Huang et al. [5] are those tweets that uses the same hashtags but are very dissimilar in content. To reduce the impact of spam account that could potentially skew the model towards words present in the spam accounts, the authors capped the number of tweets that a particular user could contribute to 10 tweets in their model. This makes the model learn from different users without getting a large influence from a particular user. The model is evaluated using hold-out cross validation with the limit of 1 tweet per user in a test set. With minimum of 100 tweets required for consideration, the model achieves a score of 72% (the percentage of tweets where the original hashtag was included in the top-20 suggestion list).

2.3 KNN model

KNN (K-Nearest Neighbors)is a supervised machine learning algorithm used for the task of solving both classification and regression problems.

Otsuka et al. [10] covers KNN as a method for the hashtag recommendation with cosine similarity as a distance metric. Cosine similarity is a method which is used to measure how similar two documents are irrespective of size. First, a test tweet is iterated with all tweets in the training set and the cosine similarity between them is calculated in each iteration. Then, the k-Nearest Neighbors of the test tweet is calculated, with $k = 200$. These neighbors are then used to rank hashtags. Finally, the top-k recommended hashtags are returned.

Dovgopal and Nohelty [3] proposes Term-Corpus Relevance (TCoR) as a similarity measure for KNN and describe it as the best method for Tweet dataset."TCoR is a weighting measure that measures how strong of a class predictor the word is across the entire dataset."[3].

$$TCoR(w) = \frac{\frac{1}{A(w)} + \frac{1}{c_w}}{2}$$

$A(w)$ is the average number of words in Tweets containing the word w, and c_w is the number of hashtags the word co-occurs with. The results based on test dataset shows that this approach of KNN recommender successfully recommended 31.4% of the tweet and perform better than Naïve Bayes model.

2.4 Other Related Work

Sedhai and Sun suggest a recommendation model for tweets involving hyperlinks [11]. The paper proposes a method of recommending hashtags using schemes like considering similar tweets, the similar documents, the named entities of the document, and the domain of the hyperlink present in the tweet. Alvari [2] presents a collaborative approach for recommending hashtags using matrix factorization. Li et al. [8] uses deep learning techniques like recurrent neutral networks to solve the problem of hashtag recommendation. Similarly, Gong and Zhang [4] brings forward the idea of using Convolutional Neural Networks (CNNs) for the task of tag recommendation. The authors uses a trigger word mechanism, and proposes a novel attention-based CNN architecture for the recommendation model.

3 DESIGN

3.1 Software Architecture Diagram

Figure 1 describes the software architecture of the project.

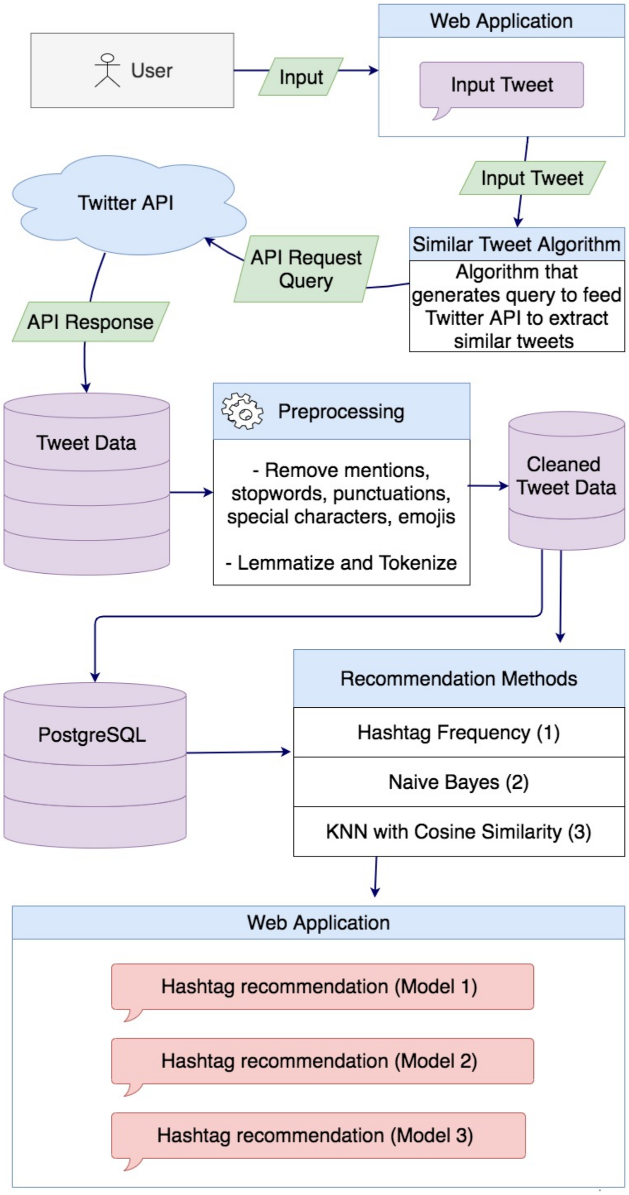

User
Input
Web Application
Input Tweet
Input Tweet
Twitter API
Similar Tweet Algorithm
Algorithm that generates query to feed Twitter API to extract similar tweets
API Request Query
API Response
Tweet Data
Preprocessing
- Remove mentions, stopwords, punctuations, special characters, emojis
- Lemmatize and Tokenize
Cleaned Tweet Data
PostgreSQL
Recommendation Methods
Hashtag Frequency (1)
Naive Bayes (2)
KNN with Cosine Similarity (3)
Web Application
Hashtag recommendation (Model 1)
Hashtag recommendation (Model 2)
Hashtag recommendation (Model 3)

Figure 1: Software architecture diagram of the project

First, the tweets are extracted from Twitter API, and they are cleaned using Python's $tweet-preprocessor$ library which removes URLs, Hashtags, Mentions, Reserved words (RT, FAV), Emojis, and Smileys from each tweet extracted. These cleaned tweets are then stored in PostgreSQL database. Using these data, the 3 hashtag recommendation models are trained namely Hashtag Frequency, Naive Bayes, and Cosine Similarity which are described in detail in next few sections. These model are then evaluated using 5 evaluation metrics namely Recall, Precision, F1 score, Hit rate, and Hit Ratio. More details on these evaluation metrics are described in the latter sections.

3.2 Similar Tweet Algorithm

To extract similar tweets from the twitter API, a Similar Tweet Algorithm is developed.

- First, the input tweet is broken down into keywords with tokenization.
- These keywords are then used to form bunch of two-word combinations.
- These grouping of two words is used with 'OR' to form a query which is used to retrieve similar related from Twitter.

3.3 Tweet Similarity using hashtag frequency

This project implements Tweet similarity using hashtag frequency method using the similar approach taken by Zangerle et al. [12]

This model is fairly straightforward. The commonly used hashtags in the retrieved similar tweets are selected to be candidate hashtags. The candidate hashtags are ranked using the SimilarityRank method proposed in Zangerle et al. [12] which ranks the hashtags based on which hashtags are present more considering all similar tweets. Finally, the top-k hashtags are recommended.

3.4 Naïve Bayes model

This project implements Naïve Bayes model using the model proposed by Mazzia et. al. [9]. Hashtags are recommended using Bayes' Theorem as illustrated in the given formula.

$$P(Hashtag\,/\,Tweets) = \frac{P(Hashtag)\cdot P(Tweets\,/\,Hashtags)}{P(Tweets)}$$

Let H_i denote the hashtag at index i. The words, $x_1, x_2, \cdots, x_n$, in a tweet are assumed to be independent. Given the list of words presented in a tweet, the probability of using a particular hashtag is calculated using the above formula.

$$P(H_i/x_1, x_2, \cdots, x_n) = \frac{P(H_i)\cdot P(x_1/H_i)\cdots P(x_n/H_i)}{P(x_1, x_2, \cdots, x_n)}$$

where,

$P(H_i)$ = *the probability of getting a particular hashtag in a tweet among all tweets*

$P(x_1/H_i)$ = *the probability of getting a word (a word that is present in the test tweet) given a particular hashtag.*

$P(x_1)$ = *the probability of getting a word (a word that is present in the test tweet)*

$P(x_1, x_2, \cdots, x_n) = P(x_1)\cdot P(x_2)\cdot P(x_3)\cdots P(x_n)$

The top-k hashtags with the higher probabilities are recommended to the user.

3.5 KNN using cosine similarity

This project implements KNN using cosine similarity based on the method discussed by Otsuka et al. [10]. Let T_{train} denote a tweet from the training set, R_{test} denote a tweet from the test set. The formula to compute cosine similarity between them is as follows:

$$CosineSimilarity(T_{train}, T_{test}) = \frac{T_{train}\cdot T_{test}}{\|T_{train}\|\cdot\|T_{test}\|}$$

.

For a particular tweet, the cosine similarity between the given tweet and the tweet in the training set is calculated. This is done for all tweets in the training set. The tweets with the bigger value are selected as similar tweets. Then, the tweets are ranked by first determining the K-nearest neighbor of the test tweet, and using these neighbors to rank hashtags. Finally, the top-K recommended hashtags are recommended to the user.

3.6 Dataset

For hashtag recommender systems, a good dataset is a must to train and test the model. This project will be using the Twitter API to collect the data.

3.7 Pre-processing

The dataset goes through a pre-processing phase for cleaning data. The regular expression module (Regex) and NLTK library is extensively used for data pre-processing. First of all, only those tweets with at least one hashtags are selected. Then, all stopwords, URLs, mentions, emojis and punctuations are removed. Then, it is lemmatized and tokenized before using it for models. The data is also stored in PostgreSQL database for reuse purpose.

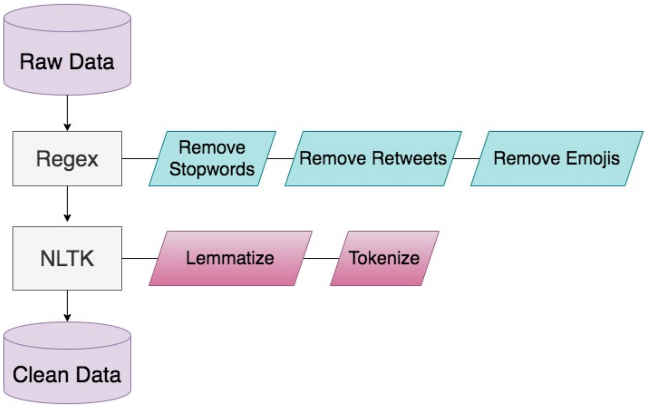
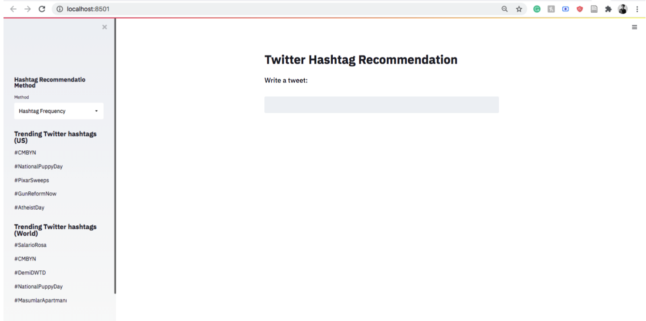


Figure 2: Pre-processing Framework

4 WEB APPLICATION

A streamlit-based web application is created which takes user's tweet as an input and recommends hashtags using those 3 developed methods. Also, the real-time trending hashtags are shown as a bonus. The figure below shows the basic design of the website



localhost:8501

Hashtag Recommendatio Method

Method

Hashtag Frequency

Trending Twitter hashtags (US)

#CMBYN

#NationalPuppyDay

#PixarSweeps

#GunReformNow

#AtheistDay

Trending Twitter hashtags (World)

#SalarioRosa

#CMBYN

#DemiDWTD

#NationalPuppyDay

#MasumlarApartmanı

Twitter Hashtag Recommendation

Write a tweet:

Figure 3: Basic design of the web application

5 EVALUATION

The proposed method of recommender systems was designed to maximize the ability to correctly match the hashtags that human use on their tweets. To check the performance of each of the methods, their performance evaluation is a must. Alsini et al. [1] mentions that the common evaluation metrics for evaluating the performance of hashtag recommendation system have been *Precision*, *Recall*, *F1 Score*, and *Hit Rate*. This project used *Precision*, *Recall*, *F1 Score*, *Hit Rate*, and *Hit Ratio* as evaluation metrics.

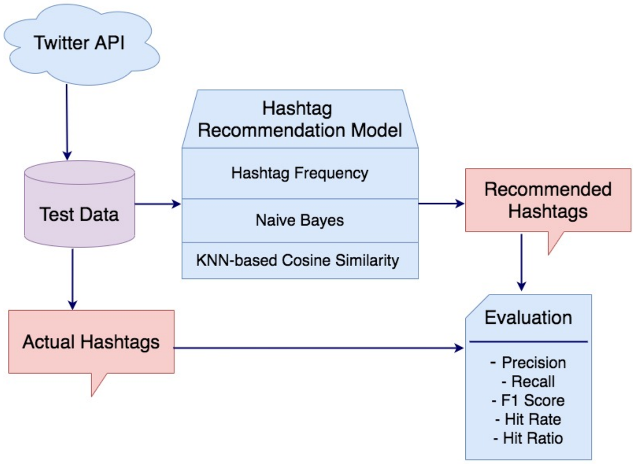


Figure 4: Evaluation Model

Let C denote the number of common hashtags between the top-k recommended hashtags and actual hashtags, N_R denote the total number of hashtags recommended, and N_A denote the total number of actual hashtags. For the case of hashtag recommendation system, *Precision* is determined by dividing the number of common hashtags between Top-k recommended hashtags and actual hashtags by number of hashtags recommended.

$$Precision = \frac{C}{N_R}$$

Recall is determined by dividing the number of common hashtags between Top-k recommended hashtags and actual hashtags by number of actual hashtags.

$$Recall = \frac{C}{N_A}$$

We can directly compute *F1 Score* with the *Precision* and *Recall* score.

$$F1\ Score = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall}$$

A *Hit rate* is calculated by determining if there's at least one common hashtag between Top-k recommended hashtags and actual hashtags. The score of 1 denotes that there is a common hashtag whereas the score of 0 denotes that there is not a common hashtag. Let H denotes the total number of tweets in the test test with an score of 1, N denotes the total number of test tweets.

$$Hit\ Rate = \frac{H}{N}$$

Alsini et al. points out the drawbacks of using these metrics for evaluating the recommendation model. Even if all actual hashtags are recommended by the model, *Precision* score will decrease with an increase in number of recommended hashtags. Similarly, even if all recommended hashtags are part of actual hashtags, *Recall* score will decrease with an increase in number of actual hashtags. Hit rate doesn't tell about the quality of recommendation since a score of 1 is achieved even with just one common hashtag. T

Alsini et al. propose a new method for evaluating hashtag recommendation systems called *Hit Ratio*. The authors describe *Hit Ratio* as the number of common hashtags between top-k recommended hashtags and actual hashtags divided by the minimum of N_R and N_A.

$$Hit\ Ratio = \frac{C}{min(N_R, N_A)}$$

Hit ratio is a useful metric for hashtag recommendation system since it provides consistent results for the case of partially correct recommendations [1]. *Hit Ratio* is also a better metric than *Hit Rate* for Hashtag recommendation as it considers the all matching hashtags in a tweet instead of just looking at one common hashtag.

This project used these five evaluation metrics described above to evaluate the performance of content-based hashtag recommendation models.

6 RESULT

Model	Precision	Recall	F1-Score	Hit-Rate	Hit-Ratio
HF	0.69	0.87	0.76	0.91	0.88
NB	0.72	0.86	0.78	0.93	0.9
CS	0.65	0.81	0.72	0.84	0.80

Table 1: Evaluation Results

Model was evaluated and verified, by the described evaluation model, using 100 high quality tweets from verified Twitter personality with at least 3 hashtags on each tweets ensuring the quality of the tweet. The result shows that Naive Bayes method is relatively better for recommending hashtags for Twitter with an average accuracy score of 0.83. It outperforms Hashtag frequency method and KNN-based Cosine Similarity Method which had accuracy score of

0.81 and 0.76 respectively.
High recall score is explained by the difference in average length of number of recommended hashtags and actual hashtags. The number of recommended hashtags was higher in most of the cases as compared of actual hashtags which also self-explain why precision come out to be so low.
Hit rate and Hit ratio, by definition, are supposed to get better accuracy score than precision and recall. They have high score here because of relatively less dependencies on number of recommended hashtags, and the presence of at least one common hashtags.

Overall, given the simplicity of the model, it can be argued that 0.82 is a pretty good score for a domain like hashtag recommendation.

7 FUTURE WORK

There is potential to improve the performance of the models with more data points if an robust date pipeline is created with PostgreSQL database to store and query real-time tweets. Similarly, the web application could be made more user-friendly.

8 ACKNOWLEDGEMENT

I would like to thank Dr. Charlie Peck and Dr. David Barbella for the detailed feedback to improve this project.

REFERENCES

[1] Areej Alsini, Du Q Huynh, and Amitava Datta. 2020. Hit ratio: An Evaluation Metric for Hashtag Recommendation. *arXiv preprint arXiv:2010.01258* (2020).

[2] Hamidreza Alvari. 2017. Twitter hashtag recommendation using matrix factorization. *arXiv preprint arXiv:1705.10453* (2017).

[3] Roman Dovgopol and Matt Nohelty. 2015. Twitter hash tag recommendation. *arXiv preprint arXiv:1502.00094* (2015).

[4] Yuyun Gong and Qi Zhang. 2016. Hashtag recommendation using attention-based convolutional neural network.. In *IJCAI*. 2782–2788.

[5] Jeff Huang, Katherine M Thornton, and Efthimis N Efthimiadis. 2010. Conversational tagging in twitter. In *Proceedings of the 21st ACM conference on Hypertext and hypermedia*. 173–178.

[6] Su Mon Kywe, Tuan-Anh Hoang, Ee-Peng Lim, and Feida Zhu. 2012. On recommending hashtags in twitter networks. In *International conference on social informatics*. Springer, 337–350.

[7] Su Mon Kywe, Ee-Peng Lim, and Feida Zhu. 2012. A survey of recommender systems in twitter. In *International Conference on Social Informatics*. Springer, 420–433.

[8] Jia Li, Hua Xu, Xingwei He, Junhui Deng, and Xiaomin Sun. 2016. Tweet modeling with LSTM recurrent neural networks for hashtag recommendation. In *2016 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 1570–1577.

[9] Allie Mazzia and James Juett. 2009. Suggesting hashtags on twitter. *EECS 545m, Machine Learning, Computer Science and Engineering, University of Michigan* (2009).

[10] Eriko Otsuka, Scott A Wallace, and David Chiu. 2016. A hashtag recommendation system for twitter data streams. *Computational social networks* 3, 1 (2016), 3.

[11] Surendra Sedhai and Aixin Sun. 2014. Hashtag recommendation for hyperlinked tweets. In *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*. 831–834.

[12] Eva Zangerle, Wolfgang Gassler, and Gunther Specht. 2011. Recommending#-tags in twitter. In *Proceedings of the Workshop on Semantic Adaptive Social Web (SASWeb 2011). CEUR Workshop Proceedings*, Vol. 730. 67–78.