Twitter Hashtag Prediction Algorithm

Instructor: Dr Huan Liu

Hamsalekha Venkatesh: 1207607657
Nivedha Padmanabhan: 1207691273
Tanushree Chakravorty: 1207664935

ABSTRACT: *Hashtag is a metadata tag used for social networks like Twitter. Hashtag prediction is the task of mapping text to its accompanying hashtags which share relevant keywords or features about the text. This helps in increasing the search visibility in browsing through a specific set of topics. However, there are not any kind of official Twitter functions for hashtag recommendation. In our implementation of a recommendation system, we have designed a content based prediction algorithm. In this technique, a Hash-Map structure is used to store the keywords and its respective hashtags. Here the words form the keys and the hashtags form its values. This behaves like a dictionary (machine learning model) and is used to predict the relationship between the words and data and locate the best suited hash tag for the test TweetFile. The hash-map is a dynamic structure and can be generated and manipulated as needed.*

KEYWORDS: Keyword, TweetFile, TestData, TrainingData,

1. INTRODUCTION

Hashtag is a type of label or metadata tag used on social network (here Twitter). Users create and use hashtags by placing the hash character (or number sign) # in front of a word and a no spaced phrase, either in the main text of a message or at the end. It appears as #word.

A hashtag is simply a way for people to search for tweets that have a common topic. All messages tagged with a particular hashtag appear along with it, when searched. For example, if you type #Gravity (or #gravity or #GraVItY, because it’s not case-sensitive) into the Search Twitter box at the top of any Twitter page and hit Enter, you’ll get a list of tweets related to the movie. What you won’t get are tweets that say “Who discovered gravity?” because “gravity” isn’t preceded by the hashtag.

However, hashtags are NOT any kind of official Twitter function. The company has not created a list of topics that we can browse through to see if there’s one that interests us. Hence, hash tag prediction is different from normal texts classification. Hashtag prediction is the task of mapping text to its accompanying hashtags. Here we don’t know how many clusters we need to find. In addition, the tag set changes so frequently that it is almost impossible to effectively carry out classification or clustering, since a new tag would force us to establish a new class and a new classification rule.

In this work we propose a novel model for hashtag prediction. Our intuition is: if a hash table of the keywords words of the message is maintained along with its corresponding hashtags and probability of their occurrence, we can predict the possible hashtags with the occurrence of the word for the new tweets. The probability of occurrence of hashtags will be able to determine the possible hashtags as 3% of the hashtags used are repeated more than 5 times.

Our algorithm is evaluated on Twitter hashtags extracted from a dataset of more than 60k tweets. We analyze the contribution and the limitations of the various feature types to the spread of information, demonstrating that content aspects can be used as strong predictors thus should not be disregarded.

In experiment, we compare our model with other classification functions and show that our model maintains a false positive rate lower than 15%. Our work is relevant for researchers interested in navigating of emergent hypertext structures, and for engineers seeking to improve the navigability of social tagging systems.

2. METHOD AND IMPLEMENTATION:

2.1 ALGORITHM

```
STEP A: Processes the data
    Step A1: Latitude and Longitude fields of the data is removed.
    Step A2: User Ids and User names have been removed.
    Step A3: URLs and special characters have been removed.
    Step A4: To remove stop words we do the following:
        Step A4.1: Run a TF-IDF on the data.
        Step A4.2: Write the results of the stop words to a different file
        Step A4.3: Remove the stop words from the data file
    Step A5: The formatted data is written to a file

STEP B: Create Hash map Using Training Data
    Step B1: Separate the non-hashtag words and hashtags from the tweets
    Step B2: Non-hashtag words become the key of the hash-map and the set of hashtags as value.
    Step B3: Iterate through each tweet for words as key:
        Step B3.1: If (word present):
        Step B3.2:Append the hashtags to the already existing key in the hash-map
        Step B3.3: else: Create a new entry for the word (key, {hashtags})
        Step B3.4: append the hashtags corresponding to the key
    Step B4:  Hash map is generated for the training data

STEP C: Predict Hashtags for the Test Data
    Step C1: The Test Data file is opened in write format.
        Step C1.1: Perform Step A on the test data
    Step C2: Iterate through  words of each tweet to separate the hashtags from non-hashtag words
        Step C2.1:Maintain a counter and condition variable
        Step C3.1: for non-hashtag words in tweet and key in Hash-map
        Step C3.2: noOfWords=min(noOfWords,4)
        Step C3.3: pick Top4Words(iterating through counter)
        Step C3.4: return Top1Hashtag for each Top4Words
    Step C4: The predicted hashtags for the non-stop words are generated.
STEP D: Check Accuracy and other Performance metrics for the hashtag Prediction
    Step D1: Precision (P) = |Relevant and Retrieved|/Retrieved
    Step D2: Recall ( R)= |Relevant and Retrieved|/Relevant
```

2.2 METHODS INMPLEMENTED TO IMPROVE EFFICIENCY

2.2.1. STOP WORDS ARE REMOVED:

The stop words are removed as they do not provide any important information about the tweets they belong to. They are considered as noise which make the dataset more impure and fuzzy. They are found very commonly throughout the document corpus and this makes it difficult and no specific meaning can be derived from them. These words are singled out and removed. This greatly improves the efficiency of the algorithm as a major portion of the tweets contain these words.

The length of the hash map is significantly reduced which in turn improves the lookup speed when evaluating the testing data.

Though it can be used for phase search, the algorithm's efficiency decreases exponentially with the increase in key in Hash-map and efficiency factor will reduce eventually. Hence removal of stop words has improved the accuracy by **~2%.**

2.2.2. RE-TWEETS NOT REMOVED:

This step was performed in order to remove duplicates from the training data and the testing data to compute the accuracy of our algorithm even with very few repeats. This was implemented intentionally to test our algorithm against a huge hash map to monitor its performance in an extreme case.

But studying our data, we decided to keep the retweeted tweets as they give a higher
The re-tweets actually indicate how popular the tweets were and can help us predict the more obvious hashtag. For Example, November related tweets most generally #Thanksgiving and #BlackFriday related tag as most relevant but recently #ParisAttack discussions where popular. Due to Re-Tweets, the probability factor of #ParisAttack to be predicted gets high. It has improved the efficiency **~8.4%**

2.2.3. TF-IDF ON TESTDATA:

The efficiency of the algorithm increases with the less number of Words in a tweet to predict hashtags for. Hence TF-IDF will reduce the number of repeated (more frequently used) words thereby the similar tags will not be assigned to most of the tweets. This improved the efficiency by **~0.6%.**

2.2.4. COUNTER ITERATIONS BETWEEN HASH TAG SETS:

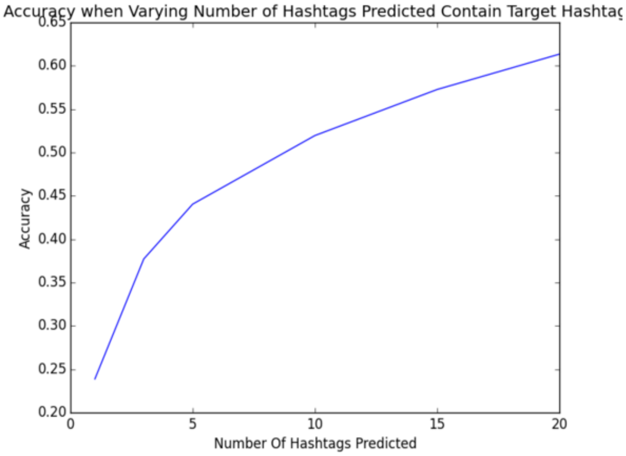
The Hashtags that are associated with the tweets are considered for prediction. The more the hashtags appear along with a word, the more is the probability of associating the same hash tag with that when it reappears in the tweets. Using this transitivity theory, a counter is iterated through the hashtags to return the maximum number appeared hashtag. This has improved the efficiency by **~6%.**

2.2.5. COUNTER ITERATIONS BETWEEN KEYS:

Though the number of words are reduced by removing Stop words and applying TF-IDF the number of words may still be more in the tweets. Hence the number of Key words are reduced through this technique. In this technique the counter is iterated through the Keywords to pick the maximum occurring Keyword in the tweet File. This has improved the efficiency by **~1.3%**.

3. RELATED WORK

Tianxi Li and Yu Wu proposed an intuitive way to solve this problem to use Euclidean distance points as a measurement of their similarity. The idea behind this approach is that in the Euclidean space, the distance is equivalent to the norm of a vector. Their compared the performances of various distance above and used 400 tweets as a test dataset which are not included in the sample dataset. Based on their conclusion, OBD (Ontology Based Distance) and cOBD (centralized Ontology Based Distance) outperform EuCD with OBD being the best in the long run. Also, the accuracy of their method went up to 86% due to the vagueness of tweets.



R.Dovgopol et.al developed a method for recommending relevant hash tags to users in real-time. Any hash tag recommendation would face several difficulties. One, the tweets is very short and often includes abbreviations, misspellings and incorrect grammar. The limited number of words in a tweet makes traditional document technique because of the fact that a word may be rarely repeated in a tweet. They computed the variation of accuracy with respect to the number of hashtags predicted.

Su Mon Kywe et al. proposed a traditional recommendation system which predicts the preference of a user towards items or social elements. They also proposed a collaborative filtering approach which is based on the underlying assumption that if a person X has adopted several common items as adopted by another person Y previously, X is more likely to adopt other Y's items than the items of random person. They manifest in 2 ways- item –to-item and user-to-user collaborative filtering. The third approach proposed by them is the content based approach which measures the similarity between the items by comparing their features and characteristics. The recommendation of an item is made to a targeted user if the item is similar to other items adopted by user before.

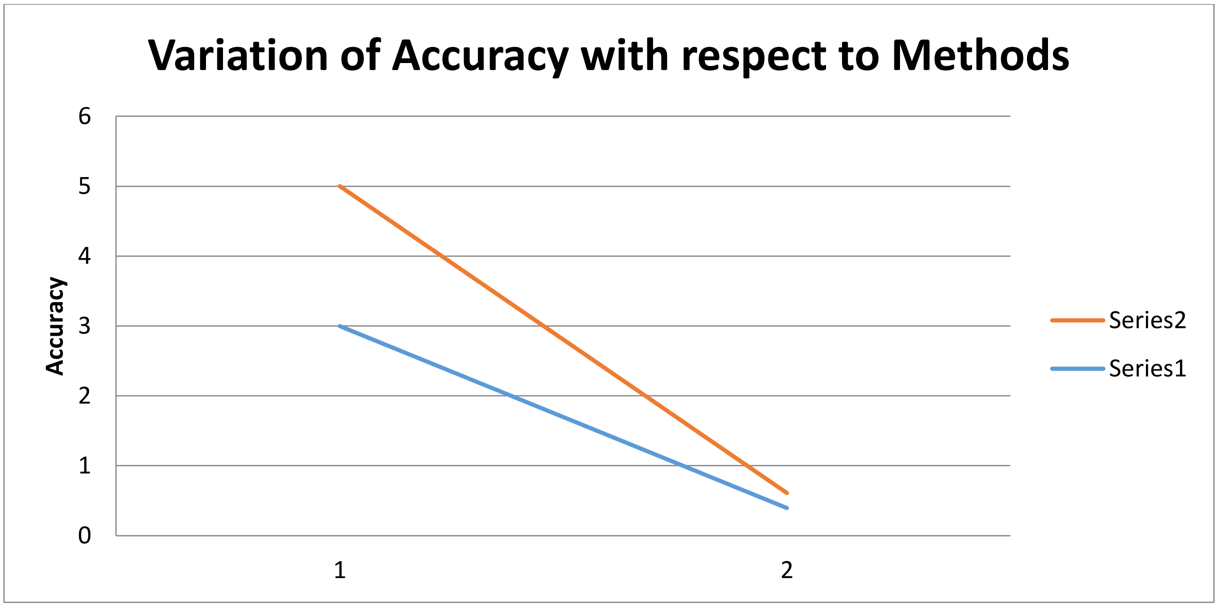
4. DATA SOURCE

The dataset consists of more than 0.26 million tweets, where 70% (42 k) of tweets is used as training data and 30% (18 k) of it is used as testing data. For the comparison between estimated output and actual output, a sample of both the data is used.

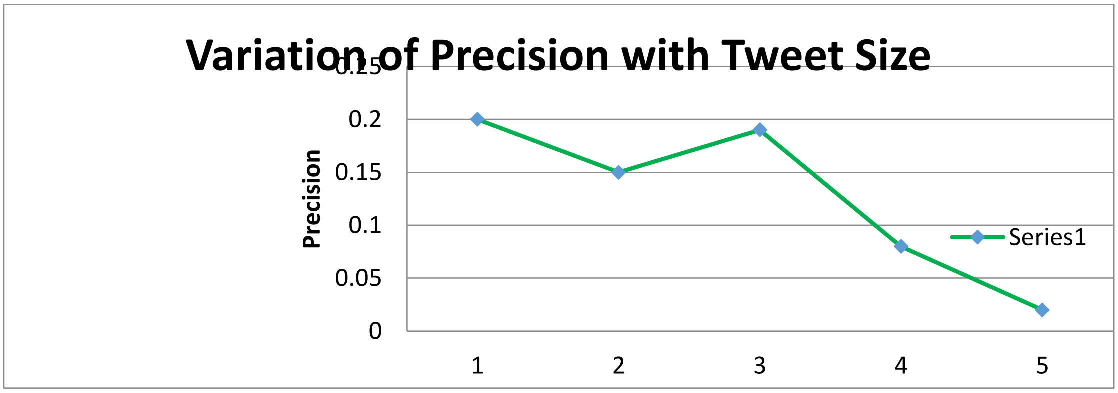
5. EXPERIMENT AND RESULTS:

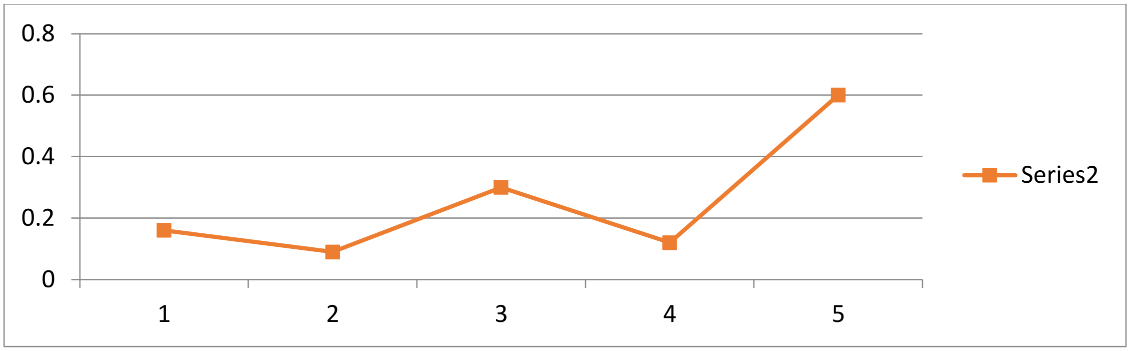
METHOD	PRESCISION	RECALL	ACCURACY
Hash Map Technique	0.15	0.12	46.4228% (best case)
Naïve Bayes Technique	---	---	~36.11% – 43.13%

- The above metrics have been generated using sample test sets for the HashMap technique.
- The Naïve Bayes implementation are the results provided with the implementation cited as [1]
- We studied the Naïve Bayes from [1] but decided to go with our method as we found it more suitable for the kind of data set we were dealing with (Note: due to high number of retweets, our method was simple, yet effective)



Series 1-is the curve obtained when all the 3 methods (Stop Words Removal, Re-Tweets Removal, TDF-IDF on Test Data); Series 2- is the curve obtained when all the 2 methods (Stop Words Removal, TDF-IDF on Test Data)





Variation of recall with Tweet Size

The above graphs show the varying values of precision and recall with respect to changing graphs.

- Some important insights from these graphs and metrics are as follows:
  - There is a constant tradeoff between precision and recall for a particular number of tweets. By keeping the tweet size at a somewhat average threshold we can achieve a good balance between precision and recall.
  - The above values in the table represents the changing accuracy against increasing number of tweets. As the testing data increases against constant training data, our algorithm suffers and the accuracy decreases as shown.

6. DISCUSSION AND CONCLUSIONS:

Due to the vagueness of many tweets, the correct rate of more than 46% is actually very high. Apart from the accuracy, our method has other advantages:

(1) The whole system is easy to implement and modify. Since it uses basic data structures like arrays and hash maps it can be easily implemented

(2) It is easy to update when dictionary changes (only needs to compute an extra column and add it back to original Hash-Map)

(3) It won't lose power when the topics trend changes with time, and it can work with personal elements and settings, which makes it more flexible.

References

[1] Damian Borth, Rongrong Ji, Tao Chen, Thomas Breuel, and Shih-Fu Chang. Large-scale visual sentiment ontology and detectors using adjective noun pairs. In Proceedings of the 21st ACM international conference on Multimedia, pages 223–232. ACM, 2013

[2] http://www.worldcomp-proceedings.com/proc/p2011/ICM3338.pdf "Twitter Hashtag PredictionAlgorithm"

[3] http://www.mysmu.edu/faculty/fdzhu/paper/SocInfo'12_43.pdf "On recommending Hashtags in Twitter Networks"

[4] https://twitter.com/hashtag/predict

[5] http://arxiv.org/pdf/1502.00094.pdf " Twitter Hashtag Recommendation " Roman Dovgopol et all.