Twitter Hashtag Recommendation using TF-IDF and Machine Learning Tools

Schokri Ben Mustapha
schokri@kth.se

Harsha Holenarasipura Narasanna
harshahn@kth.se

December 17, 2018

Abstract

Not long after Twitter introduced the hashtag as a labeling tool for tweets, the importance of which increased rapidly. For users the ability to comment on specific topics or retrieve information about them improved massively. With a high rise of micro-blogs the value of hashtags as a metadata for different kinds of data analysis became indispensable. In order to tap the full potential, hashtag recommendation became a highly demanded task in the recent years. Various machine learning methods have been implemented using different features and models. Most of them have in common that they try to retrieve trigger words from the tweets in order to relate them to the accountable hashtag. In this work we used the Term Frequency- Inverse Document Frequency (TF-IDF) to detect these words and use them as the feature input for a softmax-classifier. Although the approach is relatively simple compared to the former works the results show appreciable performance.

1 Introduction

With the rise of micro blogging platforms the ability to organize the vast amount of data became more important. On Twitter, users generate tweets to express an opinion or to comment on different topics. They can add pictures, videos and hyperlinks. In 2007, to coordinate the communication between users without limiting the freedom of how users generate tweets the hashtag was introduced. The hashtag consists of the hash symbol (#) appended with a word or a phrase without whitespace. It can be self-defined or reused from other users and appear anywhere in a blog post along with multiple hashtags. The labeling character of the hashtag can be used to add a comment to a specific topic or to get an overview about the topic using the hashtag as a search filter input. The enhancement of data analyzing applications using the hashtag as metadata lead to an increasing research attention in the NLP community. It was pointed out by [Wang et al., 2011] that only 14.6 % of all the Tweets were labeled with a hashtag. In order to benefit from the use of hashtags and cope with the short usage a recommendation systems became of big significance.

2 Related Works

The first tweet classifiers introduced methods to classify the sentiment of tweets using machine learning algorithms [Go et al., 2009]. Although, it is a different task it already pointed out major difficulties of tweet classification in general. Tweets are often written more casual leading to more slang, misspellings and often not well composed text. The limitation in character leads to relatively less information content while the number of tweets cover a vast amount of topics. All of this makes the tweet classification task more challenging then formal text classification. One of the first approaches to deal with the task of hashtag recommendation came up a few years later [Ding et al., 2012]. Their approach relates to keyphrase extraction but compared to keyphrases which are usually extracted from a given document hashtags don't have to appear in the tweet. They assume a vocabulary gap between the trigger words of tweets and hashtags and propose the combination of a topic model and a word alignment model to estimate the most probable topic-specific word alignment between the words and the hashtags. Most of the former works use textual information and uphold the assumption that the relation between tweet and hashtag can be understood by focusing on these trigger words. Some works incorporate personal and temporal information [Zhang et al., 2014] and others have analyzed the problem of hyperlinked tweets [Sedhai and Sun, 2014]. In re-

cent years neural networks have gained popularity to solve NLP tasks. Gong et al. used attention based neural network architectures and word-embeddings. They combine a local attention channel focusing on the trigger words and a global one considering all the words of the tweet.

3 Hypothesis

Scope of our model or approach is limited due to imposed time bound. However, our hypothesis comprises of both conventional statistical text classification and modern neural network approach. Although word-embeddings and NN seem to be the most promising solution we will focus on the trigger word assumption and a simpler model. Since one major problem compared to text classification was the little amount of information a tweet contains we will treat tweets labeled with the same hashtag as one document and the hashtags as class labels. By increasing the document size and adding more and more tweets we hope to get a good estimation of the subset of underlying important words related to each hashtag. We will then take the highest ranked words as an indicator for the class and use them as features to train a classifier.

3.1 Model

Model is implemented in four stages. Firstly, the large dataset with diverse topic tweets are collected. Dataset then undergoes data cleansing resulting in the relevant set of tweets. Then, the tweets are classified based on hashtags and arrive with data where set of tweets are labelled under their respective hashtags. Data is segregated for the purpose of training and testing. Secondly, conventional statistical text classification method called TF-IDF is applied with tweets set labelled under hashtag as a class for the training set. Thus, TF-IDF rank words for each class is obtained. Thirdly, full set of TF-IDF rank words form the "bag of words" for the feature set. With presence of rank words determining the feature value, bag of words is evaluated for each tweets and along with its corresponding hashtag, vectors set is formed. Vector set of the training set forge the multi-class logistic regression model. Finally, the model is evaluated against the test data and the results are studied. Each step is discussed in detail as follow.

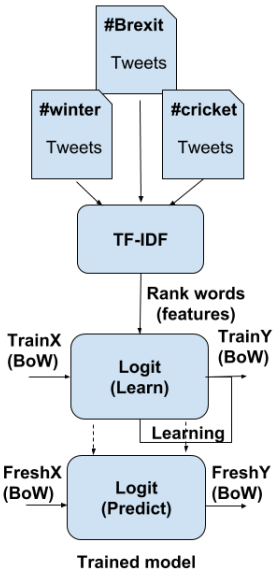

Figure 1: Proposed model

4 Implementation

4.1 Data gathering and Preprocessing

Twitter-hashtag dataset is not readily available and Twitter official API imposes restriction for the data access. Infact, Twitter allows the dataset to be commercially available for a price, signifying the importance of data. However, Get Old Tweets library [Henrique, 2018] provides the dataset by crawling the twitter search page. In essence, it exploits the principle that more and more data can be retrieved as we scroll the webpage. Moreover, the portion of tweets without hashtags are significantly higher. Thus, the library was edited to curate the results.

For a search query with defined criteria, the library provides the relevant dataset. Diverse topics were used for the search query such as wallstreet, brexit, christmas, british, cricket, scubadive. Then, dataset undergoes data cleansing which is the process of detecting and correcting (or removing) corrupt or inaccurate records from a record set and refers to identifying incomplete, incorrect, inaccurate or irrelevant parts of the data and then replacing, modifying, or deleting the dirty or coarse data. Tweets are refined from special characters, URLs, picture information, white space and stop words. In each catagory, the most frequent used hashtag is queried, the class is created and the tweets are refined by removal of

tweets without the hashtag.

Dataset is further segregated for the purpose of training and testing. Tweet-hashtag pair is randomly chosen and 75% of the data is allocated for train set while 25% for the test set.

4.2 Term Frequency - Inverse Document Frequency

To detect the subset of important words associated with the given hashtags we treat all tweets in the training data set as one document and calculate the Term Frequency – Inverse Document Frequency for all the words. The TF-IDF is the product of the Term Frequency (TF) and the Inverse Document Frequency (IDF). The TF is the frequency of the word in each document. It is calculated as the ratio between the number of occurrences $n_{w,D}$ of a word in a document and the total number of words in the document $\sum n_{w,D}$.

$$TF = \frac{n_{w,D}}{\sum n_{w,D}}$$

The IDF is a way of calculating the rareness of a word across all documents in the data set. It is calculated as the logarithm of the ratio of the total number of documents N and the number of documents the word appears in DF also referred to as the document frequency.

$$IDF = \frac{N}{DF}$$

To illustrate the performance of the TF-IDF we generate two hashtags, #LabourUnion and #Christmas, appearing in respectively two tweets. The example tweets are simplified for demonstrational reasons:

- We need a fair deal #LabourUnion
- We need a new deal #LabourUnion
- We need a tree to celebrate #Christmas
- We need a tree #Christmas

Word	**TF-IDF**	
	#LabourUnion	**#Christmas**
deal	0.06	0
tree	0	0.06
fair	0.03	0
new	0.03	0
to	0	0.03
celebrate	0	0.03

Table 4.2 shows the TF-IDF scores for the example hashtags. Both get a non-zero TF score for the words 'We', 'need' and 'a'. But since they are used in both documents the IDF is zero and therefore the TF-IDF as well. The other words have a non-zero IDF score but they only appear in one of the documents. This results only in a non-zero TF-IDF score for the documents in which they appear in. Since we are using only two documents for this example the non-zero IDF score is the same for all the words. The TF accounts for the different TF-IDF scores. The ranked scores indicate high importance of the word 'deal' for #LabourUnion and high importance of the word 'tree' for #Christmas.

4.3 Multi-class logistic regression

Multi-class logistic regression: Set of rank words obtained from the previous stage form the bag of words and represents the feature set for the logistic regression model. Presence of rank words determine the binary value of the feature. For each tweet, bag of words is evaluated and along with the corresponding hashtag, input and output vector pairs are formed.

Stage involves two steps, firstly, the model is trained with given set of training data with relevant optimization along the the process. Thereafter, performance evaluation of the model is carried out with the test data. Choice of neural network is of critical choice, in this case 'Multi-class logistic regression with softmax activation' model is chosen for its fairly robust performance and unsophisticated architecture.

$$f(v_i) = \frac{e^{v_i}}{\sum_j e^{v_j}} \tag{1}$$

Above equation represents the softmax activation calculation. v_i corresponds to ith class magnitude. Multi-class logistic regression model is implemented using the library [**?**] [G. Varoquaux;F. Pedregosa;A. Gramfort; M. Kumar; L. Buitinck; S. Wu ; A. Mensch, 2018]

sklearn.linearmodel.LogisticRegression

configured for 0.1 regularization, lbfgs solver for 3000 iterations. LBFGS stands for Limited Memory Broyden–Fletcher–Goldfarb–Shanno algorithm, suited to problems with very large numbers of variables and proven good performance for unconstrained nonlinear optimization problems [Wikipedia, 2018]. In the first step, patterns in the vector pairs are captured by the classifier. Training involves the training vector pairs and the hyper-parameters optimization for each epoch in the process. Resulting in the network model suited to function according to the trained pattern.

5 Evaluation

Evaluation of the neural model involves the performance measurement for the fresh set of test data. Test data was allocated randomly from the initial dataset and in our case about 25% and are accommodated into vector pairs form. In the process, the predicted output vectors are evaluated against the actual output for each cases and results are tabulated for later analysis. Accuracy is emphasised to conclude on the performance of the model.

6 Results

Data is captured at each stage of the system. After the first stage, we obtain the tweets set under hashtag in the textual format on an average of 7500 tweets. In the second stage, we compute the TF-IDF and top rank words are obtained. Figure 2 shows the top five rank words. Moreover, the size of rank words determine the feature size for the linear model.

	0	1	2	3	4
#wallstreet	wallstreet	newyork	joinupdots	nyc	dreamlife
#winter	transfers	minibuses	coaches	shuttle	cabs
#christmas	thecakeshop	pastrychef	pastry	pembs	cakeshop
#cricket	cricket	f7fy2	string	constitution	sport
#scubadive	scubadive	skubadiving	scuba	diving	dive
#brexit	brexit	peoplesvote	referendum	labour	blair

Figure 2: TF-IDF top 5 rank words

Further the rankwords form the bag of words and vectors corresponding to the tweets from test dataset is fed to the classifier. Obtained predicted output is evaluated against the actual value for each hashtag(class). Figure 3 depicts the confusion matrix for 3000 epochs, 600 feature size and 0.1 regularization factor.

	wallstreet	winter	christmas	cricket	scubadive	brexit
wallstreet	401	150	41	25	2	16
winter	7	113	13	6	1	0
christmas	0	18	62	2	2	0
cricket	8	114	12	554	14	4
scubadive	1	15	8	5	105	0
brexit	27	143	60	3	3	392

Figure 3: Confusion matrix for 6*100 features

Accuracy of the model can be derived from the confusion matrix and found to be 69.91%. More insights about the model could be drawn by computing performance scores such as precision, recall, fscore and support metrics. Figure 4 depicts the scores for individual classes.

	wallstreet	winter	christmas	cricket	scubadive	brexit
precision	0.903153	0.204340	0.316327	0.931092	0.826772	0.951456
recall	0.631496	0.807143	0.738095	0.784703	0.783582	0.624204
fscore	0.743281	0.326118	0.442857	0.851653	0.804598	0.753846
support	635.000000	140.000000	84.000000	706.000000	134.000000	628.000000

Figure 4: Performance metrics for 6*100 features

Further insights about the model could be drawn by varying the feature size of the neural network model and examining the corresponding accuracy of the model. Figure 5 shows the relation between feature size and accuracy. Here feature size is normalized with respect to the class size for the convenience.

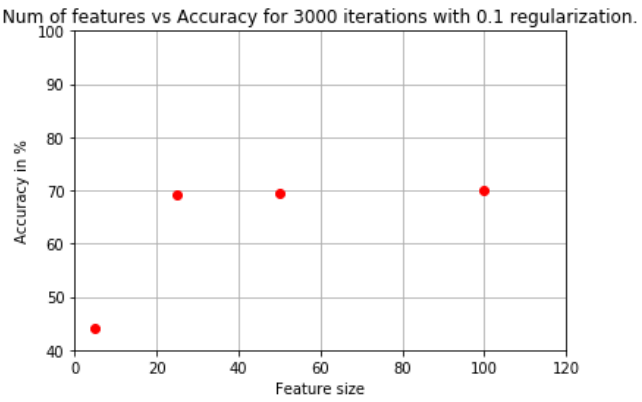

Figure 5: Feature size vs accuracy

It is worth noting that the accuracy of the model is enhanced with the increase in feature size, could be be understood as large size of bag of words increases the chances of class identification. Further, the accuracy stagnates for 50 and above likely because of less data size.

7 Conclusion

Proposed model harnesses the potential of both conventional statistical text classification TF-IDF and followed by modern neural network based multi-class logistic regression model. For our limited data set, the accuracy of 69.91% accuracy was achieved. Further, observation was made on the impact of feature size on the performance metric of the model which emphasized on the large data set for predicatability enhancement. Future consideration would be to carry out lemmatization of words, detailed experiments on

https://www

various algorithms underlying the logistic model and also in the consideration of large data set.

References

[Ding et al., 2012] Ding, Z., Zhang, Q., and Huang, X. (2012). Automatic hashtag recommendation for microblogs using topic-specific translation model. *Proceedings of COLING 2012: Posters*, pages 265–274.

[G. Varoquaux;F. Pedregosa;A. Gramfort; M. Kumar; L. Buitinck; G. Varoquaux;F. Pedregosa;A. Gramfort; M. Kumar; L. Buitinck; S. Wu ; A. Mensch (2018). scikit-learn: machine learning in Python. original-date: 2010-08-17T09:43:38Z.

[Go et al., 2009] Go, A., Bhayani, R., and Huang, L. (2009). Twitter sentiment classification using distant supervision. *CS224N Project Report, Stanford*, 1(12).

[Henrique, 2018] Henrique, J. (2018). A project written in Python to get old tweets, it bypass some limitations of Twitter Official API.: Jefferson-Henrique/GetOldTweets-python. original-date: 2016-01-29T05:34:49Z.

[Sedhai and Sun, 2014] Sedhai, S. and Sun, A. (2014). Hashtag Recommendation for Hyperlinked Tweets. In *Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval*, SIGIR '14, pages 831–834, New York, NY, USA. ACM.

[Wang et al., 2011] Wang, X., Wei, F., Liu, X., Zhou, M., and 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*, CIKM '11, pages 1031–1040, New York, NY, USA. ACM.

[Wikipedia, 2018] Wikipedia (2018). Broyden–Fletcher–Goldfarb–Shanno algorithm. Page Version ID: 873636455.

[Zhang et al., 2014] Zhang, Q., Gong, Y., Sun, X., and Huang, X. (2014). Time-aware Personalized Hashtag Recommendation on Social Media. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 203–212, Dublin, Ireland. Dublin City University and Association for Computational Linguistics.