Design and Evaluation of a Twitter Hashtag Recommendation System

Eriko Otsuka
School of Engineering and Computer Science
Washington State University
eotsuka@wsu.edu

Scott A. Wallace
School of Engineering and Computer Science
Washington State University
wallaces@vancouver.wsu.edu

David Chiu
Department of Mathematics and Computer Science
University of Puget Sound
dchiu@pugetsound.edu

ABSTRACT

Twitter has evolved into a powerful communication and information sharing tool used by millions of people around the world to post what is happening now. A hashtag, a keyword prefixed with a hash symbol (#), is a feature in Twitter to organize tweets and facilitate effective search among a massive volume of data. In this paper, we propose an automatic hashtag recommendation system that helps users find new hashtags related to their interests.

We propose the *Hashtag Frequency-Inverse Hashtag Ubiquity* (HF-IHU) ranking scheme, which is a variation of the well-known TF-IDF, that considers hashtag relevancy, as well as data sparseness. Experiments on a large Twitter data set demonstrate that our method successfully yields relevant hashtags for user's interest and that recommendations more stable and reliable than ranking tags based on tweet content similarity. Our results show that HF-IHU can achieve over 30% hashtag recall when asked to identify the top 10 relevant hashtags for a particular tweet.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Information filtering

Keywords

hashtag recommendation, twitter

1. INTRODUCTION

Twitter is one of the prevalent micro-blogging platforms today, with over 200 million active users posting *tweets*, a message limited to 140 characters [1]. The tweet's character limit promotes users to casually update posts, and with increasing ownership of mobile devices, many users are engaged to Twitter activities, resulting in over 400 million tweets sent to the Twitter network per day [2,3]. The downside to this popularity is that Twitter users may easily be overwhelmed by the massive volume of data. As a mechanism to combat the issue, Twitter users have organically incorporated the hashtag culture into their tweets. A *hashtag* is a word or a phrase without spaces prefixed with the hash symbol # inserted anywhere in the body of tweets. Trendy topics can be quickly propagated among millions of users through tagging, which creates an instant community with similar interests. With the implementation of the hashtag search feature in Twitter, many individual users and business marketers have started applying tagging to organize posts into related conversations and facilitate easier search by associated hashtags.

As tagging culture becomes widely adopted, the development of hashtag recommendation systems have gained researchers' attention. Some recent studies have proposed to recommend predefined hashtags [4,5] or general topics hidden in each tweet [6]. Though these systems are beneficial in encouraging and assisting users to get into the tagging habit, it may not be sufficient for information seekers who wish to find newly emerging hashtags. In contrast, recommending the most popular hashtags does reflect timely topics, but it often includes heavily used general hashtags and suggestions are not personalized. Other studies have proposed recommending hashtags based on similar tweets [7].

In this paper, we propose a new method to automatically recommend personalized trending hashtags based on users' tweets. Specifically, we make the following contributions:

- We build an effective hashtag recommendation system using a proposed hashtag ranking method, *Hashtag Frequency-Inverse Hashtag Ubiquity* (HF-IHU).
- We conduct a nuanced evaluation of HF-IHU over a large Twitter data set. We compare HF-IHU against several popular schemes, including: k nearest-neighbors using Cosine Similarity, k-popularity, and Naïve Bayes. Our results show that HF-IHU achieves substantially higher recall than the other schemes and is resistant to retweets.

The remainder of this paper is organized as follows. Our hashtag ranking algorithm is also presented in Section 2. In Section 3 we describe our experimental setup and the performance results of our algorithm. We discuss related works in Section 4. Finally, in Section 5 we conclude.

2. OUR APPROACH: HF-IHU

Our hashtag ranking algorithm is inspired by the well-known TF-IDF [8] approach used in information retrieval. Our algorithm relies on two central data structures that are compiled from a large number of Tweets. The first is a Term to Hashtag-Frequency-Map (THFM); the second is the converse—a Hashtag Frequency-Map (HFM). In the THFM, the primary keys are terms that have been observed in tweets. The value associated with each primary key is a map from hashtag to a frequency count indicating how often that hashtag (the secondary key) has occurred with the term specified by the primary key. The HFM is an analogous data structure using hashtags as the primary key and term frequencies as the final

IDEAS'14 July 07 - 09 2014, Porto, Portugal

http://dx.doi.org/10.1145/2628194.2628238

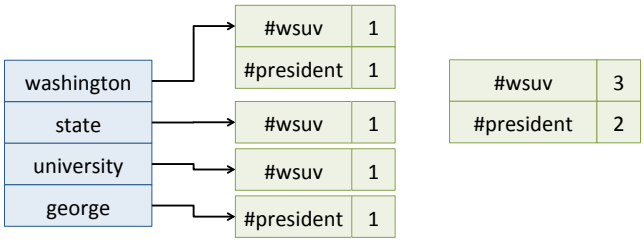

Figure 1: THFM (left) and HFM (right)

value. Figure 1 illustrates the THFM (left) and HFM (right) provided a data set containing two tweets: "washington state university #wsuv" and "george washington #president".

After generating THFM and HFM, the next step is to score hashtags in the data set to find personalized recommendations for a user. Our proposed scoring method utilizes the variation of the TF-IDF scheme, we call Hashtag Frequency-Inverse Hashtag Ubiquity (HF-IHU). HF-IHU has two opposing weighting factors: The first is the frequency with which a hashtag appears with a given term (the hashtag frequency). The second is the hashtag ubiquity which discounts hashtags that are prevalent in all contexts and rewards hashtags that are tightly associated with a narrow subset of terms.

Provided a term t and a hashtag h which co-occurred with t, $hf_{t,h}$ is expressed as follows,

$$hf_{t,h} = \frac{THFM[t][h]}{\sum_{h'} THFM[t][h']} \quad (1)$$

where $THFM[t][h]$ denotes the occurrences of h occurring with t in the corpus. The denominator is the sum of all hashtag frequencies associated with t. Thus, $hf_{t,h}$ measures the association between a term and a hashtag. Intuitively, if many users used a hashtag with a particular term, the hashtag is more likely relevant to the term.

The ihu_h is derived from the following formula:

$$ihu_h = \log \frac{|Corpus_{NH}|}{HFM[h]} \quad (2)$$

where $|Corpus_{NH}|$ denotes the number of all terms in the corpus with hashtags removed. The denominator of ihu_h is the sum of all term frequencies associated with h. Thus ihu_h decreases as the hashtag h becomes associated with a large fraction of terms in the corpus. The intuition is that these ubiquitous tags are less likely to be personally important to any given user, thus they must overcome a larger hurdle than other hashtags to be recommended. This is in contrast to the IDF term in the well-known TF-IDF, where IDF would have decreased the importance of term t, rather than h, contradicting our objective.

Our main hashtag scoring algorithm is shown in Algorithm 1. The algorithm inputs a tweet, which is a list of terms $T = (t_1, ..., t_n)$. For each term $t_i \in T$, we locate all hashtags h_j that co-occurred with it from our THFM and HFM indices. The hashtag-term frequency hf_{t_i,h_j} and the inverse hashtag ubiquity metric ihu_{h_j} are computed across all hashtags to calculate the partial score. These partial scores are aggregated for all hashtags pertaining to t_i before being returned.

3. EXPERIMENTAL EVALUATION

In this section we present an evaluation of our system. We initially describe the characteristics of the Twitter corpus we obtained and will use for evaluation.

Algorithm 1 Scoring with HF-IHU

▷ *Given a tweet with n terms:* $T = (t_1, \ldots, t_n)$
▷ *Recall THFM contains term to hashtag-frequency map*
▷ *Recall HFM contains hashtag to term-frequency map*
for all terms $t_i \in T$ **do**
  ▷ *for each hashtag co-occurring with* t_i
  for each $(h_j, f_{h_j}) \in THFM[t_i]$ **do**
    $hf_{t_i,h_j} \leftarrow \frac{THFM[t_i][h_j]}{\sum_{h'} THFM[t_i][h']}$
    $ihu_{h_j} \leftarrow \log \frac{|Corpus_{NH}|}{HFM[h_j]}$
    $S_{h_j} \leftarrow S_{h_j} + \left(hf_{t_i,h_j} \times ihu_{h_j}\right)$
  end for
end for
return S_{h_j}

3.1 Tweet Corpus

To evaluate our Hashtag Recommendation System, we first obtained the *Tweets2011* corpus, consisting of a collection of tweet identifiers, provided by Twitter for the TREC 2011 Microblog Track 2011 [9].

Twitter users often *mention* one or more users in their own tweets with *@user* to include the other users in their conversation. Although mentions appear many times in the data set, we removed these user handles from the data set because they are generally used to show interest in the mentioned user or the relationship, but not in the user itself. Additionally, we follow common information retrieval preprocessing steps by: (1) removing punctuation and non-alphanumeric symbols; (2) removing common stop-words; (3) transforming all text to lowercase; (4) stemming (we employed the Porter Stemmer from the open source Python library NLTK [10]).

We were able to download approximately 8.3 million tweets. Pre-processing eliminated 1% of these tweets because some consisted of only stop words or user mentions. We then found that approximately 13% of tweets contained at least one hashtag. We split the pre-processed data into a training set (90%) and a test set (10%): The training set contains approximately 8.1 million tweets, and $900,000$ of these contain at least one hashtag. The test set contains about $100,000$ tweets, and all of which include at least one hashtag.

3.2 Experimental Setup

To evaluate HF-IHU, we compared it with three other recommendation methods: Cosine Similarity with k-Nearest Neighbour (kNN), Overall Popularity, and Naïve Bayes. The descriptions of each tested method are briefly explained below:

- **kNN with Cosine Similarity:** Provided a tweet in the training set, t_1 and another tweet t_2 from the test set, this method computes the Cosine Similarity:

$$\cos(t_1, t_2) = \frac{t_1 \cdot t_2}{\| t_1 \| \| t_2 \|} \quad (3)$$

  For each tweet in the test data, we iterated through all tweets in the training data and computed the Cosine Similarity between them. We found the k-Nearest Neighbors ($k = 200$) of the test tweet and used these neighbors to produce a ranked list of recommended hashtags.

- **Naïve Bayes:** This method makes recommendations based on the results of a Multinomial Naïve Bayes model that is standard for text documents with large vocabularies and sparse

data. In this model, the hashtag ranking depends on the posterior probability of a hashtag H_i given a tweet composed of a set of terms t_j each with frequency f_{t_j}:

$$P(H_i|t_1,...,t_n) \propto P(H_i)\prod_j P(t_j|H_i)^{f_{t_j}} \quad (4)$$

We use Laplacian smoothing to deal with edge conditions in the conditional probability tables.

- **Overall Popularity:** This method simply recommends the most frequently occurring (popular) hashtags in the training set for each test tweet. This ranking method is not designed to make personalized recommendation, and therefore the recommendations are consistently the same hashtags for any given tweets.

To evaluate the performance of the above methods, we examined each method's ability to recall hashtags from our ground-truth tweets in the test set. Formally, let $T = \{T_1, ..., T_n\}$ denote the set of all tweets in our test set. Each tweet T_i is composed by a set of terms, $T_i = \{t_1, ..., t_m\}$, and a set of hashtags $H_i = \{h_1, ..., h_k\}$. For each method, we use the set of terms T_i to produce a set of ranked recommended hashtags $S_i = \{s_1, ..., s_p\}$ for that tweet. We then compare the recommended hashtags S_i to the ground-truth hashtags, H_i, that we removed from the original tweet. A well-functioning recommendation system will generate S_i such that most, or all, of the hashtags from H_i are highly ranked. To measure this, we define:

$$1_{H_i}(s_j) = \begin{cases} 1, & \text{if } s_j \in H_i \\ 0, & \text{otherwise} \end{cases} \quad (5)$$

Then, the total recall, R_{total}, can be computed as follows,

$$R_{total}(p) = \sum_{i=0}^{n}\sum_{j=0}^{p} 1_{H_i}(s_j) / \sum_{i=0}^{n}|H_i| \quad (6)$$

where n is the total number of tweets in the test set and p is the number of ranked recommendations provided by the method under examination.

3.3 Experimental Results

We ran all the ranking methods introduced above on the testing data and plotted $R_{total}(p)$ for various values of p (along the horizontal axis). Results are shown in Figure 2. At $p = 1$, each method returns only its top recommended hashtag for each tweet in the test set, while at $p = 100$ each method returns its top 100 recommended hashtags for that tweet. Note an ideal method will not be able to obtain 100% recall. Rather, the maximum recall ceiling lies at roughly 74% when $p = 1$ and increases to approximately 81% when $p \geq 6$. The maximum recall ceiling lies below 100% because not all tags in the test set occur in the training data. Thus, some tags in the test tweets could never be recommended. Moreover, since many test tweets have more than one hashtag, for small values of p, some tags will necessarily go unmatched, even if they would be recalled for larger values of p. The maximum recall ceiling reaches an asymptote near $p = 6$ since very few tweets in the test set have more than 6 hashtags to recall.

Figure 2 shows that HF-IHU consistently reproduced the removed-hashtags over the other three methods. The result of Overall Popularity method simply reflects the percentage of popular hashtags occurring in our test set as expected. One surprise in the results is how poorly kNN performs. One of the strengths of the HF-IHU method over kNN is that it examines the weight of all of the candidate hashtags, whereas kNN only examines at the terms-level and simply returns hashtags that occur with similar tweets. Thus, all

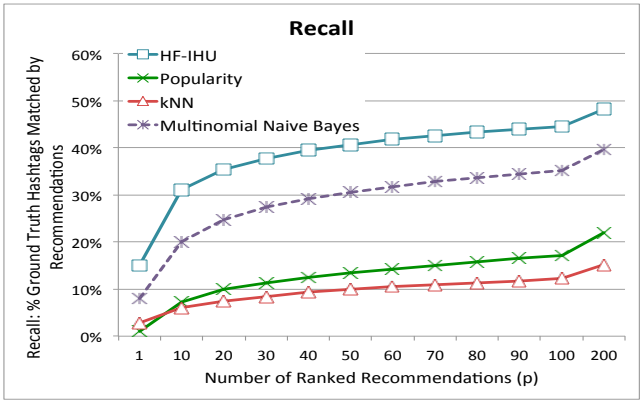

Figure 2: Total Recall for three ranking methods

hashtags in those similar tweets are ranked equally. For example, to find hashtags for a test tweet "george washington" with HF-IHU, it first computes the score for all hashtags that occurred with the term "george" (576 hashtags) and then computes for hashtags that occurred with "washington" (641 hashtags), accumulates hashtag scores, and it finally returns the top n high-scored hashtags.

With the kNN method, however, only similar tweets are used to determine recommended tags. Thus, the individual terms have little direct contribution. Rather, it is the set of terms that will determine the recommendations. For example, in our test tweet "george washington", tweets that have exactly these terms and one or more hashtags will have a perfect similarity score, while tweets that differ only in one word will be close neighbors. However, tweets that have both "george" and "washington", but also contain other terms will *not* be close neighbors. The result is that fewer tweets will be taken into account to determine recommendations. This will tend to bias the statistical relationships in an unpredictable and often undesirable manner.

As the last element of our analysis, we want to show qualitative evidence of the effectiveness of the recommendation system. To this end, we retrieved a list of the most prolific users (tweeted most frequently) in our data set. From this list, we then selected three sample users with clear interests: *@XboxSupport*, *@jewishblogger*, and *@freeprojectinfo*. *@XboxSupport* is a twitter account set up to provide support for XBox users. *@jewishblogger*, according to their profile page are a "worldwide leader in Jewish and Israeli blogs". *@Freeprojectinfo* tweets about freelance job postings. These sample users were selected because their tweets seem to focus on a relatively narrow range of topics and thus we should be able to manually validate recommendations provided by our system with a reasonable amount of confidence.

Given the tweets by each sample user as input, Table 1 lists the top-10 recommended hashtags ranked with our proposed method. For each recommended hashtag, we determined if the tag was clearly related to the topics covered based on the account profile, and if so we marked that hashtag as a *hit*. When a recommended hashtag had no intuitive semantic value, we performed a web search to provide a first order approximation on the meaning associated with the tag before determining whether it qualified as a hit.

Table 1 shows that the recommended hashtags ranked with HF-IHU include many pertinent hashtags for *@jewishblogger* and *@freeprojectinfo*, but only a few relevant hashtags for *@XboxSupport*. `#vuze` was the only tag that did not have an intuitive semantic value. A cursory search indicates that Vuze is a program that allows users to stream music and videos through devices, such as

@XboxSupport	*@jewishblogger*	*@freeprojectinfo*
#vuze •	#israel •	#jobs •
#kinect •	#jewish •	#freelance •
#egypt	#obama	#webdevelopment•
#jan25	#israeli •	#job •
#jobs	#telaviv •	#egypt
#fb	#synagogue •	#design •
#sissyboys	#gasztro •	#jan25
#xbox •	#parashat •	#fb
#ff	#jan25	#seo •
#nowplaying	#jerusalem •	#wordpress •
Hits: 3	**8**	**7**

Table 1: Top 10 Recommended hashtags ranked with HF-IHU

XBox consoles, so it was deemed a hit.

Unlike the hashtags recommended by HF-IHU, kNN fails to identify *any* intuitively salient tags for our three sample users (due to space constraints, the results are not shown). Moreover, most of recommended hashtags by kNN are in the top 50 popular hashtags. As observed in the evaluation with retweets, the performance of kNN method is directly affected by retweets in the data set. Since there are more terms that were tweeted with popular hashtags, it is more probable that tweets containing popular hashtags score high with Cosine Similarity.

4. RELATED WORKS

Zangerle, *et al.* compare three different hashtag ranking methods in Recommending #-Tags in Twitter [7]. Receiving a user's tweet, they first find similar tweets in their data set using TF-IDF, and retrieve a set of candidate hashtags that appeared in these most similar tweets. They rank the hashtags based on the overall popularity of candidate hashtags, the frequency of candidate hashtags within the most similar tweets, and the similarity score of the most similar tweets. The reported results show that the third method performed the best in recommending hashtags. Their approach solely relies on tweets similarities and those hashtags occurred in the most similar tweets are recommended to users, whereas our approach focuses more on terms in tweets and the relevance of those terms to candidate hashtags.

Kywe, *et al.* proposed a method that recommends hashtags retrieved from similar users and/or similar tweets [11]. They compute the preference weight of a user towards a hashtag in the data set using the TF-IDF scheme, and then select the top n users who scored high in cosine similarity between a user and another user. The top m similar tweets are selected in a similar manner. Their approach basically adds more hashtags (used by similar users) to the list of candidate hashtags retrieved by the method proposed by Zangerle, *et al.*. However, when target users have never used hashtags before, the recommendations only include hashtags from similar tweets. Although user similarity is taken into account in this method, many of recommended hashtags may be from similar tweets because the majority of tweets do not contain hashtags [11,12]. Additionally, their approach still focuses on similarities in terms and used hashtags, while our approach does not rely on similarities.

Godin, *et al.* observe the challenge of ranking hashtags based on the tweet's similarity and recommending hashtags existing in similar tweets due to the sparseness of hashtags [6]. Their approach focuses on detecting hidden topics for the tweets and then suggest the use of those general topics as hashtags using a Latent Dirichlet Allocation (LDA) model. Although both ours and their approach take into account the sparseness of micro-blog data, the fundamental difference is that their approach limits the suggestions to general topics. Our approach, in contrast, attempts to retrieve relevant and emerging hashtags in the data set.

5. CONCLUSION

The objective of this paper was to implement an effective hashtag recommendation system that automatically suggests a list of personalized hashtags emerging real-time for Twitter users. We proposed a ranking method, *Hashtag Frequency-Inverse Hashtag Ubiquity* (HF-IHU), which is a variation of the TF-IDF weighting scheme to score hashtag relevancy while also taking into account data sparseness of Twitter data set. Our experiments on a large Twitter data set demonstrated that our proposed method performed better than other methods that rely only on hashtag popularity and tweet similarity. We conducted experiments on the top 10 high-scored hashtags for selected users. Compared with a ranking method based on k-Nearest Neighbors, the experiments exhibited that our system consistently assigned high score on hashtags that interests the user.

While our research has demonstrated promising results on recommending personalized hashtags, the scope of the research can be extended in several other directions in the future. For example, text sentiment could potentially be used to detect user's interests more accurately and make better hashtag recommendations. Some previous efforts show sentiment analysis on the whole tweet [13,14]. Zhang, *et al.* propose sentiment analysis at the entity level [15]. We could exploit this analysis so that entities with positive sentiment have a greater impact on the hashtag recommendations than entities with negative or no sentiment.

6. REFERENCES

[1] T. Schreiner, "New compete study: Primary mobile users on twitter." https://blog.twitter.com/2013/new-compete-study-primary-mobile-users-on-twitter.

[2] J. Heggestuen, "One in every 5 people in the world own a smartphone, one in every 17 own a tablet."

[3] H. Tsukayama, "Twitter turns 7: Users send over 400 million tweets per day," Mar. 2013.

[4] B. Sriram, D. Fuhry, E. Demir, H. Ferhatosmanoglu, and M. Demirbas, "Short text classification in twitter to improve information filtering," in *SIGIR'10*, ACM, 2010.

[5] S. Garcia Esparza, M. P. O'Mahony, and B. Smyth, "Towards tagging and categorization for micro-blogs," 2010.

[6] F. Godin, V. Slavkovikj, W. De Neve, B. Schrauwen, and R. Van de Walle, "Using topic models for twitter hashtag recommendation," in *WWW '13 Companion*, pp. 593–596, 2013.

[7] E. Zangerle, W. Gassler, and G. Specht, "Recommending#-tags in twitter," in *SASWeb 2011*, pp. 67–78, 2011.

[8] K. S. Jones, "A statistical interpretation of term specificity and its application in retrieval," *Journal of Documentation*, 1972.

[9] NIST, "Tweets2011 twitter collection." http://trec.nist.gov/data/tweets, August 2011.

[10] S. Bird, E. Klein, and E. Loper, *Natural Language Processing with Python*. O'Reilly Media, Inc., 1st ed., 2009.

[11] S. M. Kywe, T.-A. Hoang, E.-P. Lim, and F. Zhu, "On recommending hashtags in twitter networks," in *Social Informatics*, pp. 337–350, Springer, 2012.

[12] M. Efron, "Hashtag retrieval in a microblogging environment," in *SIGIR '10*, pp. 787–788, ACM, 2010.

[13] A. Go, L. Huang, and R. Bhayani, "Twitter sentiment analysis," *Entropy*, vol. 17, 2009.

[14] T. Wasserman, "Twitter sentiment to light up london's ferris wheel," July 2012.

[15] A. Mudinas, D. Zhang, and M. Levene, "Combining lexicon and learning based approaches for concept-level sentiment analysis," in *WISDOM '12*, pp. 5:1–5:8, ACM, 2012.