Understanding and Analysing Microblogs

Pinar Yanardag Delul
« Supervised by: S.V.N. Vishwanathan »
Purdue University
Department of Computer Science
ypinar@purdue.edu, vishy@stat.purdue.edu

ABSTRACT

Microblogging is a form of blogging where posts typically consist of short content such as quick comments, phrases, URLs, or media, like images and videos. Because of the fast and compact nature of microblogs, users have adopted them for novel purposes, including sharing personal updates, spreading breaking news, promoting political views, marketing and tracking real time events. Thus, finding relevant information sources out of the rapidly growing content is an essential task.

In this paper, we study the problem of understanding and analysing microblogs. We present a novel 2-stage framework to find potentially relevant content by extracting topics from the tweets and by taking advantage of submodularity.

Categories and Subject Descriptors

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

General Terms

Algorithms, Design, Experimentation

Keywords

Twitter, Topic Models, Social Media, Personalization, Recommendation, Submodularity

1. INTRODUCTION AND PROBLEM

In the recent years, microblogging services became a popular medium to spread real-time news, to share personal updates, to promote opinions and many more. Analysing the characteristics of the microblog messages is a critical task and it can be useful in many ways, such as personalized content recommendation, friend recommendation, emerging or evolving news detection and viral marketing.

However, due to unique characteristics of microblogs, understanding and analysing the content of the messages is still an ongoing challenge. In particular, texts are short, topics evolve quickly and the language is different than standard written English due to usage of abbreviations, symbols and slang words. Therefore, standard text mining and topic modelling tools do not work well on this data and we need smarter ways to extract topics from microblogs.

WWW 2013 Companion, May 13–17, 2013, Rio de Janeiro, Brazil.
ACM 978-1-4503-2038-2/13/05.

In this paper, we study Twitter, one of the most popular microblogging platform on the Web with more than 500 million users. Users share and discover content by sending and receiving 140-character messages known as 'tweets'. People who subscribe to a user's messages are called 'followers' and people that the user follows are called 'friends' or 'followees'.

Twitter has a rapidly growing content where 340 million tweets are generated daily as of 2012. Due to the fast growth of the service, users are easily overwhelmed by the large amount of text and it is very difficult for users to dig out information of interest. Following too many accounts might easily overwhelm the user by generating too many messages on the timeline. Following very few accounts will cause the user to miss important and interesting pieces of information. Therefore, we need to consider a fundemental balance: we don't want to follow users who are very similar to us (otherwise, our timeline will be flooded with the same type of content), yet we don't want to follow someone who is very different from us (e.g. a musician might not want to follow a programmer).

Finding similar, high quality and reliable information sources on Twitter is a challenging issue. Unlike traditional recommendation systems, we do not have any explicit information available about the user's interests (such as ratings on the items user likes or dislikes). User's followers/followees network and published tweets are the only information available to exploit. Therefore, we want to analyse these implicit feedback provided by the user and suggest other users who might be potentially interesting to this user.

Solving this problem brings two significant benefits. Firstly, it helps users to discover new interesting information sources. Secondly, it improves interaction between similar-tasted users and builds an interest-based social network.

We summarize the contributions in the following: (1) We propose a 5-stage topic extraction pipeline which takes an individual user as an input and semantically enriches the contents of the tweets. (2) We propose a submodular framework which takes benefit of the proposed topic-extraction pipeline to recommend relevant content such as users to follow or tweets to read.

The rest of this paper is organized as follows. In Section 2, we give an overview of previous related work. In Section 3, we present our proposed approach and methodology and motivate it with respect to existing works. Finally, we draw conclusions in Section 4.

2. STATE OF THE ART

Some researchers focused on modifying standard LDA to work with Twitter. Previous efforts include aggregating all the tweets of a user into a single document [8] which follows a author-topic model. However, this model fails to capture the fact that each tweet has its own topic assignment. Latest approaches such as Twitter-LDA [9] tried to overcome this issue, however they assumed that a single tweet is usually about a single topic which conflicts with our assumptions that a tweet is about multiple topics with different layers. Labeled LDA [7] is another LDA-based approach, however their model relies on hashtags, thus it might not cover the topics the user mentions.

On the other hand, traditional methods such as TF-IDF doesn't work on Twitter since it assumes that the indexed documents are at a reasonable length.

Therefore, current models don't work well on Twitter because of the short content of the tweets: they don't contain sufficiently enough word co-occurance information for bag of words representations, thus leads to poor performance.

Therefore, we need a novel approach to understand microblogs.

3. PROPOSED APPROACH AND METHODOLOGY

Our proposed framework consists of two main components: (1) Topic Extraction Pipeline (2) Submodular Framework. We first discuss topic extraction pipeline, and than introduce the submodular framework.

3.1 Topic Extraction Pipeline

Our main assumption is; a tweet consists of multiple topics with different layers where each layer has an associated weight. Our pipeline makes use of Hashtags, Part of Speech tags, topics of URLs, Freebase and a Wikipedia-based search engine as layers and extracts pieces of information from each layer which might be useful as a topic assignment. Therefore, the tweet 'Hacked my Emacs setup with evil, Clojure-mode.' will associate not just with topics explicitly mentioned like Emacs and Clojure, but also ones obliquely referenced like Open Source, Linux, Programming Languages, Text Editors.

Pre-processing: First of all, we discard all the tweets that have less than 30 characters, less than 8 tokens, less than 3 english nouns and more than 5 english stop words. We also discard all replies and tweets that include smileys since they often indicate personal messages.

After that, we extract multiple entities that are helpful to detect the importance and emphasis of a given tweet as follows:

Hashtags: Hashtags are an alternate way to associate a topic with a tweet, by simply placing a hash symbol (#) in front of a topic. We obtain hashtags of the tweet by simply extracting words start with "#" character.

Part of Speech Tags: Part of speech tags are critical entities that can give very important feedback about the topic of a tweet. We obtain part of speech tags with NLTK [6] toolkit and we only consider NNS (Noun, plural), NNP (Proper noun, singular) and NNPS (Proper noun, plural) tags for topic modelling since they often indicate useful keywords for a given tweet.

URLs: Similarly, we treat URLs as valuable, external sources of information and we extract topics given the content of the URL with the help of AlchemyAPI [1]. This will help us to extract topics that are mentioned indirectly in a tweet.

Twitter allows users to forward a tweet through their network with *Retweet* (RT) functionality. Even though a RT is not written by the user in question, we intuitively think that a retweeted message still has an importance since the user wanted to share it with his/her network. Therefore, when analysing the profile of a user, we still consider Retweets but we give a lower weight to them comparing to user's actual tweets.

After our pre-processing pipeline, we obtain a set of terms consist of hashtags, POS tags and URLs' topic assignments per tweet and we feed the information obtained in pre-processing step to external sources.

Augmenting topics with external sources: After the pre-processing step, we obtain a new representation for the tweet t_i as t'_i which doesn't have any unnecessary entities. Our main intuition for topic augmentation is to treat each tweet t'_i as a search engine query and to use a search engine to retrieve relevant results for our query.

For this purpose, we built a search engine using ElasticSearch (a Lucene based search engine) [2] and we fed it with the complete English Wikipedia corpus. Our search engine takes a pre-processed tweet t'_i as an input and returns a set of Wikipedia entities which are relevant for the analysed tweet. The result set includes a relevance score per Wikipedia article and we treat the article titles as topics as well as adopting the relevance scores as weights for the corresponding topics.

After we get topic assignments from Wikipedia, we use Freebase to address the ambiguity problem as well as augmenting topic assignments to a new extend. In particular, we would like to taxonomically extend topics into a set of relevant topics. For example, *Emacs* and *Vim* are two widely used open-source text editors, but there is no way for us to know that they both belong to *text editor* category and they are both *open-source softwares*. Therefore, we use Freebase, one of the most widely used data sources on the Linked Data Web to augment the assigned topics for the tweet.

After augmenting topics with Wikipedia, we have a weighted bag of terms per each tweet with an associated relevance score. In particular, this bag of terms is constructed with hashtags, topics that we got from URL extraction process, part of speech tags, topics obtained from Elastic Search and topics from Freebase (see Figure 1).

Once our framework extracts topics from different aspects with an associated relevance score, we then combine all of those topic assignments into a single topic distribution. In order to reduce the variability of the topic set, we use Porter stemmer to stem the topic assignments and we sum up the weights of the topics who fall into same topic name. Figure 2 shows two topic clouds for a Ruby programmer who works at Google (image is generated via wordle.net). The topic cloud on the left is the raw representation of the user's tweets which were generated with all of the tweets without any processing. The topic cloud on the right side of Figure 2 shows our framework's representation. We can see that on the left side, topic cloud is very noisy and dominated by non-topical words where on the right side of the image, topics are cleanly represent the interests of the user.

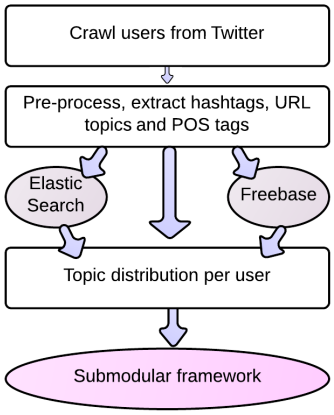

Figure 1: Topic extraction pipeline and submodular framework

We also associate a weight with each information source. In particular, we think that topics extracted from hashtags are the most relevant ones since user speficially used a hashtag in order to indicate the topic of the tweet. Therefore, hashtags have the highest weight among others. After the hashtags, we assume URL topics are the second most important topic assignments. Finally, we weight Elastic search topics and Freebase topics with the same weight followed by part of speech tags.

Cumulative score of the topic per tweet, $tscore(t_{ij})$ is calculated as follows:

$$tscore(t_{ij}) = \phi_{ht}.(w_{ht(ij)})+\phi_{url}.(w_{url(ij)})+\phi_{wiki}.w_{wiki(ij)}+ \phi_{freebase}.w_{freebase(ij)} + \phi_{pos}.w_{pos(ij)} \quad (1)$$

where $\phi_{ht(i)}$, $\phi_{url(i)}$, $\phi_{wiki(i)}$, $\phi_{freebase(i)}$, $\phi_{pos(i)}$ are user defined parameters in order to give different weights for each factor and $\phi_{ht(i)} = 4$, $\phi_{url(i)} = 3$, $\phi_{wiki(i)} = \phi_{freebase(i)} = 2$ and $\phi_{pos(i)=1}$.

After obtaining the combined score of a topic i given a tweet j, we then weight all the topic assignments in the tweet by the score of the tweet, where score of a tweet is calculated as a linear combination of number of people who retweeted the item, whether it is a Retweet or original content of the user. After obtaining final score of all the topics in a tweet, we then sum up all the weights given a single topic.

Given a user u and a set of tweets with their topic assignments, we combine weights of each term as follows:

$$weight(u_k) = \sum_{i=1}^{ntweets} \sum_{j=1}^{nterms} tscore_{ij}.iscore_i$$

where *ntweets* indicates the number of tweets the user has, and *nterms* indicates the number of terms each tweet has. Here, $tscore_{ij}$ is the total weight of the $term_j$ given $tweet_i$ and $iscore_i$ is the importance score of the tweet.

Finally, we choose top 1000 topics and finalize our topic distribution.

3.2 Submodular Framework

We first introduce some background information about submodularity. $f : 2^V \rightarrow \mathbb{R}$ is a set function which maps subsets $S \subseteq V$ of a finite ground set V to real numbers. $f(\cdot)$ is called normalized if $f(\varnothing) = 0$, and it is monotone if $f(S) \leq f(T)$ whenever $S \subseteq T$. $f(\cdot)$ is called submodular if for any $S, T \subseteq V$, if we have the following:

$$f(S \cup T) + f(S \cap T) \leq f(S) + f(T) \quad (2)$$

A provably equivalent definition of submodularity property is diminishing returns, where $f(\cdot)$ is submodular if for any $R \subseteq S \subseteq V$ and $s \epsilon V \setminus S$, we have the following:

$$f(S \cup \{s\}) - f(S) \leq f(R \cup \{s\}) - f(R) \quad (3)$$

.

According to Eqn. 3, the *value* of s never increases when the context gets larger, which satisfies the property of diminishing returns.

We have a set of users $V = u_1, u_2, \ldots, u_n$ where certain user pairs are similar and similarity of the $user_i$ and $user_j$ is measured by a non-negative function. Given a $user_i$, we want to select a high quality and compact subset S of users by maximizing a submodular function.

Based on the definition of submodular functions, we can define the utilization of submodularity for answering our problem. Basically, given a $user_i$, we want select a subset of users from the category that are very similar to $user_i$ but at the same time diverse from each other as much as possible. In other words, we want to maximize the similarity between selected users and $user_i$ while minimizing the redundancy between selected users. Therefore, not only we will select the most representative users but also the selected subset will cover different aspects of the category.

For this purpose, we adopt a similar approach to MMR (Maximal Marginal Relevance) [3] which maximizes information coverage and minimizing the redundancy, as follows:

$$MMR = argmax_{D_i \in R \setminus S} [\lambda(Sim_1(D_i, Q) - (1\text{-}\lambda)max_{D_i \in S} Sim_2(D_i, D_j)) \quad (4)$$

We use the following objective function [5] that was inspired by MMR:

$$f_{rel}(S) = \sum_{i \in V \setminus S} \sum_{j \in S} sim_{i,j} - \lambda \sum_{i,j \in S: i \neq j} sim_{i,j} \quad (5)$$

where $\lambda \geq 0$.

Following a similar intuition to MMR method, a user has a high marginal relevance if it is both relevant to $user_i$ and contains minimal similarity to previously selected users. Notice that first term of f_{rel} measures the similarity between

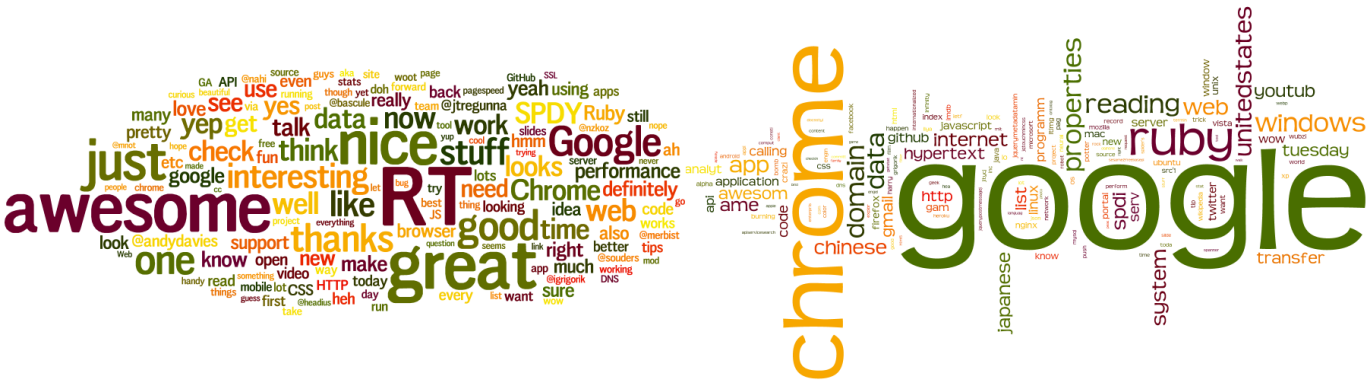

Figure 2: Word cloud generated for a Ruby programmer who works at Google where size of the words are weighted by their importance (Left image: Raw representation of the tweets Right image: Topics extracted by our framework)

selected subset S and rest of the users $V \setminus S$ and the second term is sum of pairwise similarity of selected user versus already selected users' similarity.

Therefore, we greedily select the next most relevant user to $user_i$ and at the same time avoiding redundancy by removing users that are too similar to other users that are already selected. Thus, we incrementally build a userlist for $user_i$ by choosing the next user with the highest marginal relevance. λ is a parameter that controls the level of importance of choosing relevant users versus diverse users. In order to benefit from submodular optimization, the objective function measuring the similarity must be submodular (if both terms in f_{rel} is submodular, then our function is also submodular). Hence, we use adopt Jaccard index and use a weighted version of it to calculate the similarity. Finally, we use a greedy algorithm adopted from [4] to perform the user selection.

4. RESULTS

To do our experiments, we first crawled top 600 people from 10 different categories at Wefollow.com, a user-powered Twitter directory. After getting a total of 6000 users in music, news, politics, tv, sports, entrepreneur, comedy, social media, programmer and blogger categories, we crawled up to 3200 tweets per each user. Then, we fed our topic-extraction pipeline with each user and obtained the topic assignments per user. We then fed our submodular framework with a sample user and ask algorithm to select 3 similar users yet diverse users.

Based on our experiments, topic extraction pipeline works reasonably good as can be seen from Figure 2. Submodular framework is also working as expected based on our preliminary experiments. We expect to extend our experiments in very near future.

5. CONCLUSIONS AND FUTURE WORK

Understanding and analysing microblogs is a challenging issue. Finding high quality and reliable information sources on Twitter is a very beneficial problem which can be useful in many ways, such as personalized content recommendation, friend recommendation, emerging or evolving news detection and viral marketing.

In this paper, we proposed a 2-stage framework to perform user recommendation task on Twitter. To the best of our knowledge, this is the first paper which uses submodularity to do user recommendation on Twitter. We used a 5-layered topic extraction pipeline and we used a diversity-oriented submodular function, and our results seems promising so far. Currently we are working towards user studies and experiment the framework with larger datasets.

As a future work, we plan to develop a web-based interface where users can give explicit feedback about the recommendations and we plan to utilize a wider spectrum search engine to query pre-processed tweets. We also plan to perform our experiments to recommend tweets.

6. REFERENCES

[1] Alchemyapi. http://www.alchemyapi.com.

[2] Elasticsearch. http://www.elasticsearch.org.

[3] H.˜Dang. Overview of duc 2005. In *Proceedings of the Document Understanding Conference*, 2005.

[4] J.˜Leskovec, A.˜Krause, C.˜Guestrin, C.˜Faloutsos, J.˜VanBriesen, and N.˜Glance. Cost-effective outbreak detection in networks. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 420–429. ACM, 2007.

[5] H.˜Lin and J.˜Bilmes. Multi-document summarization via budgeted maximization of submodular functions. In *North American chapter of the Association for Computational Linguistics/Human Language Technology Conference (NAACL/HLT-2010), Los Angeles, CA, June*, 2010.

[6] E.˜Loper and S.˜Bird. Nltk: The natural language toolkit. In *Proceedings of the ACL-02 Workshop on Effective tools and methodologies for teaching natural language processing and computational linguistics-Volume 1*, pages 63–70. Association for Computational Linguistics, 2002.

[7] D.˜Ramage, D.˜Hall, R.˜Nallapati, and C.˜Manning. Labeled lda: A supervised topic model for credit attribution in multi-labeled corpora. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1-Volume 1*, pages 248–256. Association for Computational Linguistics, 2009.

[8] M.˜Steyvers, P.˜Smyth, M.˜Rosen-Zvi, and T.˜Griffiths. Probabilistic author-topic models for information discovery. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 306–315. ACM, 2004.

[9] W.˜Zhao, J.˜Jiang, J.˜Weng, J.˜He, E.˜Lim, H.˜Yan, and X.˜Li. Comparing twitter and traditional media using topic models. *Advances in Information Retrieval*, pages 338–349, 2011.