MICROBLOG RETRIEVAL USING TOPICAL FEATURES AND QUERY EXPANSION

Cher Han Lau, YueFeng Li, Dian Tjondronegoro

Queensland University of Technology

ABSTRACT

Retrieving information from Twitter is always challenging given its volume, inconsistent writing and noise. Existing systems focus on term-based approach, but important topical features such as person, proper noun and events are often neglected, leading to less satisfactory results while searching information from tweets. This paper propose a novelty feature extraction algorithm which targets the above problems, and present the experiment results using TREC11 dataset. The proposed approach considers both term-based and pattern-based features and distribute weights accordingly. We experiment four different setting to evaluate different combinations and results show that our approach outperformed traditional method of using term-based or pattern only methods and signify the importance of topical features in microblog retrieval.

1. INTRODUCTION

Microblogging has emerged as one of the primary social media platforms for users to post short messages from personal updates, questions and content of interest. One of the popular microblog service providers is Twitter. Twitter has attracted over 200 million registered users and publishing 50 million tweets per day. Users from Twitter come from different areas include national leaders, celebrities, field experts and general public. The significant use of twitter has been witnessed in various application from stock forecasting [1], event monitoring [2] to natural disasters [3]. It also plays an important role in critical situation where reliable communication is not available such as 2009 Iranian election [4] and Mumbai terrorist attack [5]. i

In addition to information sharing, users refer to Twitter for information about breaking news and real-time events. By searching Twitter, users can obtain instantaneous updates on issues of their interest in a timely manner, and from multiple perspectives. However, it is a challenging task as tweets are unstructured, ungrammatical and prone to noise. The 140 characters word limit also causes users to employ different strategies such as abbreviations and slangs in order to compress more information. Furthermore, vast amount of tweets delivered makes it impossible for human to read and distil useful information, without the help of machines. Hence, there exist the needs for a system to assist users to retrieve information effectively from Twitter.

Most of the current microblog IR system relying on term-based model such as TF-IDF, BM25 and probabilistic model [6, 7, 8]. The advantages of term-based model is efficient computation performance and the maturity of term weighting theories. However, term-based approaches often suffering from problems of polysemy and synonym and very sensitive to term use variation. Topical feature such as phrases and named entities (e.g. person, location and proper nouns) are also often neglected. This problem is even more evident in microblogs due to the amount of noise and causes poor retrieval.

In TREC 2011, we explore the topical feature discovery in Twitter from both data mining and information retrieval perspectives. We present an algorithm to extract topical feature from tweets using pattern mining technique to capture meaningful pattern. We then assign weight to the terms and word pairs based on the term distribution in each pattern. We evaluate our approach using TREC 2011 Microblog track dataset. Experiment results show that our method performs better than existing term-based solutions.

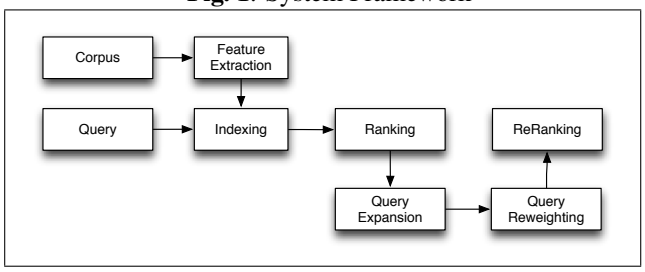
2. RELATED WORK

Twitter has been used to search for information for its large amount of social information and timeliness response. Users searched twitter particularly for answers, timely information (e.g. news, events), people information and topical information [9, 10]. However, search functionality provided by Twitter is limited to keyword-based retrieval to return the most recent posts. Users are unable to explore the results or retrieve more relevant tweets based on the content, and may get lost or become frustrated by the information overload [11].

Various retrieval models have been proposed to assist users in retrieving information from Twitter. Magnani et al. proposed a user-based tree model for retrieving conversations from microblogs [12]. A system by [7] performs data mining on users demographic information (twitter clients, geographical location, gender) to visualise underlying properties of tweets for decision making. Query-likelihood retrieval model can be used to identify subtopics for further browsing [13].

Apart from specially designed twitter retrieval systems, different unique properties of Twitter are also exploited for retrieval purpose. Hashtag is commonly used to conduct search for topics of interest (*#greysanatomy*), monitor event devel-

Fig. 1. System Framework



opment (*#emmy*) and discuss trending issues (*#carbontax*). Hashtag has also shown useful for relevance feedback and query expansion [6]. The statistics of tweets published, followers count and following-followers ratio can be used to estimate the authority of users to rank and improve the retrieval result [14].

Current twitter retrieval relies heavily on term-based approach, such bag-of-words (BOW) model. Each tweet is considered as a collection of pre-processed (e.g. normalized, stemmed) terms with weight scores (e.g TF-IDF) assigned. This approach is suitable for tweets as it is topic specific and performance is reliable without any advanced computation. However, it is also shown to be very sensitive to noise [15]. The word limit in tweet is causing users to use different terms and abbreviations, which degrades the retrieval performance. Different strategies have been adopted to overcome the microblog retrieval issues. For instance, query expansion is used to capture more relevant terms from the initial query [16]. Topical features such as named entities is used in clustering task and but its use in retrieval task was not yet explored [17].

Most of the current study on twitter retrieval are using keyword-based technique and the result is not promising (as it is still sensitive to noise). The potential of using other techniques and features such has yet to be studied. In this research, we proposed a twitter retrieval framework focus on using topical feature, combine with query expansion using pseudo-relevance feedback (PRF) to improve microblogs retrieval results.

3. SYSTEM FRAMEWORK

This section will detail the design of our system framework as shown in Figure 1. Given a set of microblog corpus, we first perform feature extraction and index into database. A query will then be matched with the tweet index to retrieve initial resultset. The result is then used to expand the initial query with more relevant terms. Finally the query will be reweighted and used to re-rank the resultset to return the final result.

The following observations are made and taken into considerations during our framework design. Firstly, tweets are limited by 140 characters, and can be noisy and terms usage varying. Secondly, tweet contains concise topical information about person, location and events [10]. Lastly, users initial query are always short and often does not include relevant keywords which leads to low recall. Therefore, we design our framework with the following goals:

- To consider both important terms-based and topical feature, which aims to capture important patterns such as $\{$*barack, obama*$\}$, $\{$*brisbane, australia*$\}$
- To automatically expand initial query with relevant terms to improve retrieval result.

A main control process handles the following tasks during experiments: expand query (details provided in later section), assign feature weights and rank results. Assuming in microblogs retrieval, users are more interested in most recent results, tweets are always sorted by relevance then by time in descending order. The logical flow of the process is detailed in Algorithm 1.

Algorithm 1 Main Control Process

Input

- A set of tweets. $D = \{d_1, d_2, d_3, ...d_n\}$
- Initial query, q

Method

1. Use q to retrieve top 100 tweets in D.
2. Sort the retrieved tweets based on time.
3. Form training set Ω using top 10 of the sorted tweets
4. Form expanded query Q using terms from Ω
5. Use Q to retrieve 1000 tweets from D and sort based on time
6. Use top 30 tweets as the final result

4. FEATURE EXTRACTION AND WEIGHTING

As twitter search is topic specific and not only terms but also named entity, it is important to consider both important terms and pattern in our feature extraction. For term-based feature, we adopt Vector Space Model (VSM) and TF-IDF weighting scheme as shown in Equation 1. We define that tweet document d contains a set of terms where $d = \{t_1, t_2, t_3, ..., t_n\}$, the weight w of term t in d can be calculated as the product of the average term frequency with proportion to the length of tweet and the logarithmic inverse document frequency.

$$w_{t,d} = \frac{tf_{t,d}}{|d|} \times \log(\frac{N}{df_i}) \quad (1)$$

For topical feature, we adopt Frequent Pattern Mining (FPM) approach to find the closed pattern. Frequent pattern mining is to find out the pattern that occurs with a minimum

occurrence (*support*) greater than the preset minimum support (*min_sup*) and a closed pattern is the longest pattern of size n with same support of the pattern of size $n-1$.

Let D be the collection of tweets where $D = \{d_1, d_2, d_3, ..., d_n\}$, and each tweet d contains terms where $d = \{t_1, t_2, t_3, ..., t_n\}$, T denotes all the terms in the collection D. We then define termset X be a set of terms from T, $coverset(X) = \{d | d \in D, X \subseteq d\}$. Its support $sup(X) = |coverset(X)|$. A termset X is called a "*frequent pattern*" if $sup \geq min\_sup$, a pre-defined minimum support.

A pattern $p = \{t_1, ..., t_n\}$ is an unordered list of terms (in this research, we limit maximum n to 3). A pattern $p_2 = \{x_1, \cdots, x_i\}$ is a sub-pattern of another pattern $p_1 = \{y_1, \cdots, y_j\}$, denoted by $p_2 \subset p_1$. With that, we can say that p_2 is a sub-pattern of p_1 and p_1 is super-pattern of p_2. A frequent pattern is defined as "*closed pattern*" if not $\exists$ any super-pattern X_1 of X such that $sup(X_1) = sup(X)$, and $P = \{CP_1, CP_2, \cdots, CP_n\}$ denotes the closed frequent pattern for all tweets.

Next, we develop different weighting strategies. By evaluating term supports based on their appearances in patterns. The evaluation of term supports (weights) in this paper is different from term-based approaches. For a term-based approach, the evaluation of a given terms weight is based on its appearances in tweets. In this research, terms are weighted according to their appearances in discovered pat- terns.

- RUN1 – Terms only (TO): Baseline run using terms frequency without weighting.
- RUN2 (§ 4.1) – Pattern weighted terms (PWT): Terms based and weighing terms using term occurrence in patterns.
- RUN3 (§ 4.2) – Term weighted Patterns (TWP): Patterns only and weighing pattern using weights of the terms within the pattern.
- RUN4 (§ 4.3) – Weighted Terms and Pattern (WTP): Combine second and third weighting scheme. (Main run)

4.1. Pattern Weighted Terms (PWT)

Hypothesis A term t_i is more important than t_j if t_i appears in more pattern than t_j and t_i is more important than t_j if it appears in more frequently appear pattern. The weight $w(t)$ can be calculated by aggregating the support of patterns where t appear, as below:

$$w(t) = \sum_{i=1}^{|P|} |\{p | p \in P_i, t \in p\}| \tag{2}$$

4.2. Terms Weighted Patterns (TWP)

Hypothesis A pattern p_i is more important than p_j if aggregated weight of $t \in p_i$ are more important than aggregated weight of $t \in p_j$. The weight $w(p)$ can be calculated by aggregating the term frequency for all terms in p,

$$w(p) = \sum_{t \in p} w(t) \tag{3}$$

A term t_i is more important than t_j if t_i appears in more pattern than t_j and t_i is more important than t_j if it appears in more frequently appear pattern. Therefore, the weight $w(t)$ can be calculated by aggregating the support of patterns where t appear, as below:

4.3. Weighted Terms and Pattern (WTP)

If the hypothesis from 4.2 and 4.3 hold, term-based feature can be obtained using PWT and topical features can be extracted by pattern mining using TWP. With that, we can deduce that if a tweet is represented by both feature, it will represent both term and topic feature, therefore:

$$t_{term} = \{(t_1, tw_1), (t_2, tw_2), \cdots, (t_n, tw_n)\}$$

$$t_{pattern} = \{(p_1, pw_1), (p_2, pw_2), \cdots, (p_n, pw_n)\}$$

$$t = t_{term} \cup t_{pattern}$$

4.4. Query Expansion

One main problem in microblog retrieval is that query is short and unable to accurately describe users informations needs. As users compose tweets using different terms, many tweets that are relevant but without the query terms inside will not be retrieved. One way to overcome this problem is to perform query expansion which expand the initial query with more relevant terms.

A typical query expansion technique is pseudo-relevance feedback (PRF). PRF is an automatic relevance feedback process based on local analysis, which assumes top retrieved documents are relevant and extract the terms from the top retrieved documents and add them back to the query. The benefits of using PRF is that it improves retrieval performance without external interaction [18], which is ideal for twitter retrieval since it is realistically difficult to perform multi-iteration relevance feedback.

The query expansion is done using a Vector Space Model (VSM) with TF-IDF weighting. Given an initial query q, we first retrieve 100 relevant tweets and rank it reverse chronologically. The top 10 tweets are then selected as the training set Ω, and all the terms in Ω are used to expand q and form Q. Expanded query Q is then used to retrieve 1000 tweets and rank reverse chronologically. The top 30 tweets are then selected as the final result.

Table 1. Dataset Statistics

Total Tweets	16,141,812
Null Tweets	1,204,053
ReTweets	2,596,642
Non-english Tweets	5,204,053
Indexed Tweets	4,952,843
Total Tokens	27,240,636

4.5. Similarity Metric

In vector space model, all queries and documents are represented as vectors in dimensional space V, where V represents all distinct terms in the collection. The similarity of documents is determined by the similarity of their content vector. This has led to three problems: (i) Low frequency terms in the collection will be assigned relatively high weight and (ii) The similarity score is low due to the absence of terms in query, and (iii) semantically related terms that does not appear in query will not be retrieved. Therefore, cosine similarity will not be used as the similarity and instead Jaccard Index will be used. Equation denotes the similarity function between a query q and a tweet d:

$$Sim(q, d) = \frac{|q \cap d|}{q \cup d} \tag{4}$$

5. EVALUATION

The performance of our microblog retrieval will be evaluated based on both standard Information Retrieval (IR) metrics, using a set of 50 topics provided by TREC. The topics is manually selected and by expert users. TREC11 microblog dataset will be used for the evaluation.

5.1. Dataset Description

Dataset from TREC'11 microblog track is used for our evaluation. The dataset consists about 16 million tweets collected during 2010-01-23 to 2010-02-08 Tweets are filtered by language and only English tweets are used in the experiment. We then perform noise removal and stopwords removal on tweets. The statistics of the dataset is detailed in Table 1.

5.2. Results and Discussions

This section we will discuss our evaluation from both TREC wide results and intra-run results. The TREC wide results describe and compare the performance of our system compared to other system. Intra-run results detail the characteristics of our runs and the pros and cons.

Table 2. Performance for highly relevance

RunID	MAP	R-PREC	P@30
Run1A	0.0753	0.1114	0.1347
Run2A	0.0486	0.0846	0.1694
Run3A	0.0673	0.1191	0.2034
Run4A	0.0486	0.1040	0.1973

Table 3. Performance for highly relevance

RunID	MAP	R-PREC	P@30
Run1A	0.0797	0.0983	0.0374
Run2A	0.0419	0.0566	0.0424
Run3A	0.0646	0.0846	0.0556
Run4A	0.0353	0.0513	0.0515

Table 4. Topics with poor performance

TopicID	Topic Title
MB002	2022 FIFA soccer
MB005	NIST computer security
MB011	Kubica crash
MB014	release of "The Rite"
MB015	Thorpe return in 2012 Olympics
MB016	release of "Known and Unknown"
MB017	White Stripes breakup
MB018	William and Kate fax save-the-date
MB030	Keith Olbermann new job
MB038	protests in Jordan
MB048	Egyptian evacuation

5.3. Results

Table 2 and Table 4 shows the Mean Average Precision (MAP), Recall Precision (R-PREC) and the official Precision when 30 tweets are retrieved (P@30) for All Relevance (ALLREL) judgement and Highly Relevant (HIGHREL) judgement. ALLREL includes 49 topics and HIGHREL includes only 30 topics contain highly relevant tweets, on the TREC relevance scale. Topic 50 is removed from the evaluation due to insufficient relevant tweets. Table **??** shows the topics which our system did not perform well.

The highest MAP score in RUN1 is 1.00 for topic MB018 - *"William and Kate fax save-the date"*, where MB040 - *"Beck attacks Piven"* topped RUN2 and RUN3 with MAP score of 0.25 and MB012 - *"Assange Nobel peace nomination"* scored the highest with 0.30 in RUN4. In overall, RUN1 with terms frequency only performs the best and performance of RUN2 – Weighted Terms (PWT) performance is the worst. Weighted Pattern (TWP) is acceptable however it does not improve the performance of PWT as we expected. The rea-

son behind why PWT performance is worst might be caused by the amplification effect of the original terms only (TO) approach.

In P@30 measurement, WTP performs similarly to PWT, and both significantly outperformed both term-based approach. The highest precision score in RUN1 is 0.67 for topic MB020 - *"Taco Bell filling lawsuit"*, and topic MB009 - *"Toyota Recall"* achieved best result in RUN2, RUN3 and RUN4 with precision score 0.73.

6. CONCLUSION AND FUTURE WORK

The work presented in this paper investigated four different settings to evaluate the effectiveness of term-based and pattern-based feature, for use in microblogs retrieval. We addressed the short user query problem by using pseudo-relevance feedback to expand initial query and apply different weights for different features. A similarity metric is recommended to consider only the terms appear in query, to prevent query result to be affected by non-existed terms. While the retrieval performance are still far from ideal when compared with performance of traditional IR in full length document, our work has definitely shed some light in microblogs retrieval field.

In our approach, we considered pattern mining as a key technique to extract topical feature in a natural way. Important terms are always together to indicate importance. In addition, we considered different weighting scheme to experiment the validity of our hypothesis and the results agree with our hypothesis that topical features is good for microblogs.

It is also worthwhile to mention that none of our proposed method uses any external and future evidence. This is because we strongly believe that whatever belongs to the data it has to be coming out from the data. The data itself is the best descriptors to describe the properties and nature of data at the point of time when a query is issued.

Due to the fact that we believe there are still possibility to use simple approach to solve the microblog retrieval problem. In addition, complex model with various parameters estimation and adjustment required high computational power and makes the system more sensitive to noise, therefore it is not encouraged to do so.

In the future, we plan to include more twitter related characteristics to improve the result. Another key consideration is also whether or not we should change the algorithm to improve the quality of pseudo-relevance tweets returned. We do not intend to change the feature extraction methodology as we want to keep it as simple and straightforward as possible, however the model can be further improved to achieve higher retrieval performance.

7. REFERENCES

[1] S. Asur and B.A. Huberman, "Predicting the future with social media," in *2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*. IEEE, 2010, pp. 492–499.

[2] David A. Shamma, Lyndon Kennedy, and Elizabeth F. Churchill, "Tweet the debates: understanding community annotation of uncollected sources," in *WSM '09: Proceedings of the first SIGMM workshop on Social media*, New York, NY, USA, 2009, pp. 3–10, ACM.

[3] Takeshi Sakaki, Makoto Okazaki, and Yutaka Matsuo, "Earthquake shakes twitter users: real-time event detection by social sensors," in *WWW '10: Proceedings of the 19th international conference on World wide web*, New York, NY, USA, 2010, pp. 851–860, ACM, p851-sakaki.pdf.

[4] A. Burns and B. Eltham, "Twitter free iran: an evaluation of twitters role in public diplomacy and information operations in irans 2009 election crisis," in *Record of the Communications Policy & Research Forum*, 2009, pp. 298–310.

[5] Onook Oh, Manish Agrawal, and H. Rao, "Information control and terrorism: Tracking the mumbai terrorist attack through twitter," *Information Systems Frontiers*, vol. 13, pp. 33–43, 2011, 10.1007/s10796-010-9275-8.

[6] Miles Efron, "Hashtag retrieval in a microblogging environment," in *Proceeding of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, New York, NY, USA, 2010, SIGIR '10, pp. 787–788, ACM.

[7] Marc Cheong and Vincent Lee, "Integrating web-based intelligence retrieval and decision-making from the twitter trends knowledge base," in *SWSM '09: Proceeding of the 2nd ACM workshop on Social web search and mining*, New York, NY, USA, 2009, pp. 1–8, ACM.

[8] Jilin Chen, Rowan Nairn, Les Nelson, Michael Bernstein, and Ed Chi, "Short and tweet: experiments on recommending content from information streams," in *Proceedings of the 28th international conference on Human factors in computing systems*, New York, NY, USA, 2010, CHI '10, pp. 1185–1194, ACM.

[9] M. Efron and M. Winget, "Questions are content: A taxonomy of questions in a microblogging environment," *Proceedings of the American Society for Information Science and Technology*, vol. 47, no. 1, pp. 1–10, 2010.

[10] J. Teevan, D. Ramage, and M.R. Morris, "#twittersearch: a comparison of microblog search and web

search," in *Proceedings of the fourth ACM international conference on Web search and data mining*. ACM, 2011, pp. 35–44.

[11] Michael Bernstein, Lichan Hong, Sanjay Kairam, H. Chi, and Bongwon Suh, "A torrent of tweets: Managing information overload in online social streams," in *In Workshop on Microblogging: What and How Can We Learn From It? (CHI 10*, 2010.

[12] Matteo Magnani, Danilo Montesi, Gabriele Nunziante, and Luca Rossi, "Conversation retrieval from twitter," in *Advances in Information Retrieval*, Paul Clough, Colum Foley, Cathal Gurrin, Gareth Jones, Wessel Kraaij, Hyowon Lee, and Vanessa Mudoch, Eds., vol. 6611 of *Lecture Notes in Computer Science*, pp. 780–783. Springer Berlin / Heidelberg, 2011.

[13] Brendan O'Connor, Michel Krieger, and David Ahn, "Tweetmotif: Exploratory search and topic summarization for twitter," in *ICWSM*, 2010.

[14] R. Nagmoti, A. Teredesai, and M. De Cock, "Ranking approaches for microblog search," in *Web Intelligence and Intelligent Agent Technology (WI-IAT), 2010 IEEE/WIC/ACM International Conference on*. IEEE, 2010, vol. 1, pp. 153–157.

[15] Nasir Naveed, Thomas Gottron, Jérôme Kunegis, and Arifah Che Alhadi, "Searching microblogs: coping with sparsity and document quality," in *Proceedings of the 20th ACM international conference on Information and knowledge management*, New York, NY, USA, 2011, CIKM '11, pp. 183–188, ACM.

[16] K. Massoudi, M. Tsagkias, M. de Rijke, and W. Weerkamp, "Incorporating query expansion and quality indicators in searching microblog posts," *Advances in Information Retrieval*, pp. 362–367, 2011.

[17] Fabian Abel, Ilknur Celik, Geert-Jan Houben, and Patrick Siehndel, "Leveraging the semantics of tweets for adaptive faceted search on twitter," in *The Semantic Web ISWC 2011*, Lora Aroyo, Chris Welty, Harith Alani, Jamie Taylor, Abraham Bernstein, Lalana Kagal, Natasha Noy, and Eva Blomqvist, Eds., vol. 7031 of *Lecture Notes in Computer Science*, pp. 1–17. Springer Berlin / Heidelberg, 2011.

[18] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze, *Introduction to Information Retrieval*, Cambridge University Press, 1 edition, July 2008.