University of Tehran at RepLab 2014

Abolfazl Aleahmad[a], Payam Karisani[a], Masoud Rahgozar[a], Farhad Oroumchian[a,b]

[a] Database Research Group, Control and Intelligent Processing Center Of Excellence, School of Electrical and Computer Engineering, University of Tehran
[b] University of Wollongong in Dubai

Abstract. In this paper, we present our approach to author ranking subtask; which is a part of author-profiling task in RepLab 2014. In this subtask, systems are expected to detect influential authors and opinion makers on Twitter website. The systems' output, for a given domain, must be a ranked list of authors according to their probability of being an influential author or opinion maker. Our system utilizes a Time-sensitive Voting algorithm, which is based on the hypothesis that influential authors tweet actively about topics of their interest. In this method, hot topics of each domain are extracted and a time-sensitive voting algorithm ranks each authors on their respective topics.

Keywords: Microblog retrieval, twitter profile ranking, social networks.

1 Introduction

Twitter has become a common means of spreading personal and public information in recent years. Users in all ranges of social status are twitting about different subjects on the Internet. The upward trend of using the websites like Twitter is confirmed by [5], which shows Twitter had 200 million users until February 2013. On the other hand, the information need of the internet users should be addressed in this important category of social media that has been subject of many researches in the information retrieval field. The key question in textual information retrieval is how to compute the relevance probability of a document with regard to a user query. Three major factors are generally used in effective retrieval models [6]: term frequency, document length and term inverse document frequency.

From these three factors, term frequency and document length normalization are not meaningful in microblog retrieval due to the short length of users' posts. On the other hand, there are some other parameters such as users' hash tags or retweets that are far more important and have been exploited by microblog retrieval techniques. These facts show special and different nature of microblogs that should be considered in information retrieval algorithms.

The third year of Replab campaign addresses Online Reputation Management systems. It comprises of two major tasks, which are Reputation Dimensions and Author Profiling. The Author Profiling task itself consists of two subtasks: Author Categorization and Author Ranking. This paper describes our experiments in the author ranking subtask. This subtask is presented by Replab organizers as below:

> "Systems will be expected to find out which authors have more reputational influence (who the influencers or opinion makers are) and which profiles are less influential or have no influence at all. For a given domain (e.g. automotive or banking), the systems' output will be a ranking of profiles according to their probability of being an opinion maker with respect to the concrete domain, optionally including the corresponding weights"

Our aims in the author ranking task is to verify the hypothesis that influential authors tweet more about hot topics in their domain compared to the other users.

The rest of this paper is organized as follows: section 2 describes the collection and preparation process of the provided RepLab 2014 dataset and our experimental setup, section 3 presents our proposed algorithm, section 4 reports our experimental results, and section 5 concludes the paper.

2 Experimental Setup

The dataset of the author profiling task consists of nearly 7500 English and Spanish twitter profiles which are categorized into automotive, banking and Miscellaneous domains. Every profile has at least 1000 followers and at the crawling time, the last 600 published tweets of each profile are crawled.

The dataset is split into two training and test sets that contain around 33% and 67% of the profiles respectively. The training set consists of 1185 and 1315 profiles from automotive and banking domains and the test set contains 2345, 2500, 146 profiles from automotive, banking and miscellaneous domains respectively. Table 1 shows the dataset features.

Table 1. RepLab 2014 dataset features

Feature	Description
tweet_id	the related tweet id
profile_id	the related profile id
domain_id	domain of the profile
tweet_url	tweet's URL
language	tweet's language
Timestamp	tweet's published time

The evaluations are carried out based on manual judgments of reputation experts. The outputs are stored in the standard TREC format and the traditional information retrieval criteria (MAP, R-Precision, and P@N) are used to evaluate the performance of each system.

The author ranking collection of RepLab 2014 contains approximately 4.5 million tweets from 7491 profiles. This collection was downloaded directly from Twitter; Table 2 contains properties of the crawled collection:

Table 2. Some statistics from the crawled collection

Description	Value
Total number of profiles	7491
Total Number of tweets	4486868
Number of English tweets	3192787
Number of Spanish tweets	1211714
Number of unknown language tweets	82367
Number of tweets not crawled by our crawler	127255
Tweets starting date	May 24, 2010
Tweets ending date	February 8, 2014

We used the Twitter's standard API to get the number of the followers for each profile. But because of the limitations of the Twitter API, we developed a tool to download the HTML page of each tweet which is used to extract the text, the retweet count, and the favorite count of each tweet. Our proposed algorithm does not use any external resources.

The tweet messages are stored in TREC format and then indexed in Terrier 3.5 [7]. Our submitted runs are experimented using the Terrier retrieval engine with default settings (stopword removal and porter stemmer is applied, etc.). Also, we used the Stanford Topic Modeling Toolbox [3] to predict the significant topics in each domain which is discussed in the next section.

3 Our Algorithm

Table 3 shows the steps of our algorithm, repeated identically for each domain Di (e.g. automotive, banking):

Table 3. Our algorithm

1- Topic Creation: topics are extracted for the domain Di, as follows:

a) Hashtags (words starting with "#" that state topics of discussions in twitter) are extracted from tweets belonging to the domain Di. Then the number of profiles that used each hashtag is calculated.

b) Let HashDi be hashtags of Di sorted based on the number of profiles that used them.

c) Let QDi be top N most frequent hashtags in HashDi that are not present in HashDj for all j | i<>j.

2- Retrieval: Let Ri(Qj) be the set of 1000 tweets retrieved by PL2 model (implemented in Terrier) for the topic QDi,j.

3- Topic based profiles ranking: Let ProfileTopicRanki,j be the set of profiles ranked by Time-sensitive Voting algorithm based on each list Ri(Qj).

4- Topics' weight calculation: Let Weighti,j be the precision that is calculated for each Ri(Qj) based on the relevance judgments of the training dataset.

5- Calculate final author rankings: Let T be a constant weighting threshold. Let ProfileRanki be ProfileTopicRanki,j lists with Weighti,j ≥ T merged by weighted averaging using weights Weighti,j.

Finally, ProfileRanki,j will contain the final rank of profile j in each domain Di.

4 Official Runs

We submitted 5 different runs to the author ranking sub-task of RepLab 2014. The following table describes each run briefly:

Table 4. Description of the submitted runs

Run Name	Description
UTDBRG_AR_1	This is the output of the algorithm in table 1 with N=100. But instead of using Time-sensitive Voting, Local method of [9] is used and in step 1 of the algorithm, instead of tweets hashtags, all tweets terms are considered.
UTDBRG_AR_2	This is the output of the algorithm in table 1 with N=50. But only tweets which are retweeted more than 100 times are considered in step 1.
UTDBRG_AR_3	This is the output of the algorithm in table 1 with N=50 and instead of Time-sensitive Voting, the Voting method of [8] is used.
UTDBRG_AR_4	This run uses the number of followers to re-rank the result of the first run named 'UTDBRG_AR_1'
UTDBRG_AR_5	This is the output of the algorithm in table 1 with N=100

The threshold T is considered zero in all the above runs. In the remaining part of this section we compare the submitted runs based on the official results released by the track organizers. Table 5 compares the 5 submitted runs based on the MAP measure:

Table 5. Comparison of the official runs based on MAP for the two domains

	UTDBRG AR_1	UTDBRG AR_2	UTDBRG AR_3	UTDBRG AR_4	UTDBRG AR_5
Automotive	0.7047	0.4565	0.6767	0.7206	0.6871
Banking	0.3961	0.3689	0.3208	0.4103	0.3183

Also the following figures compare the submitted runs based on Precision-Recall and P@N measures:

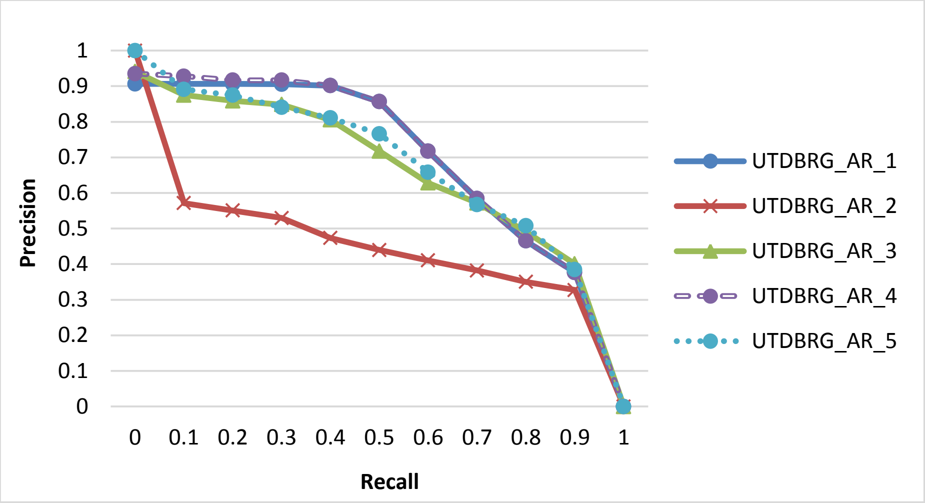

Fig. 1. Precision-Recall comparison of the runs in automotive domain

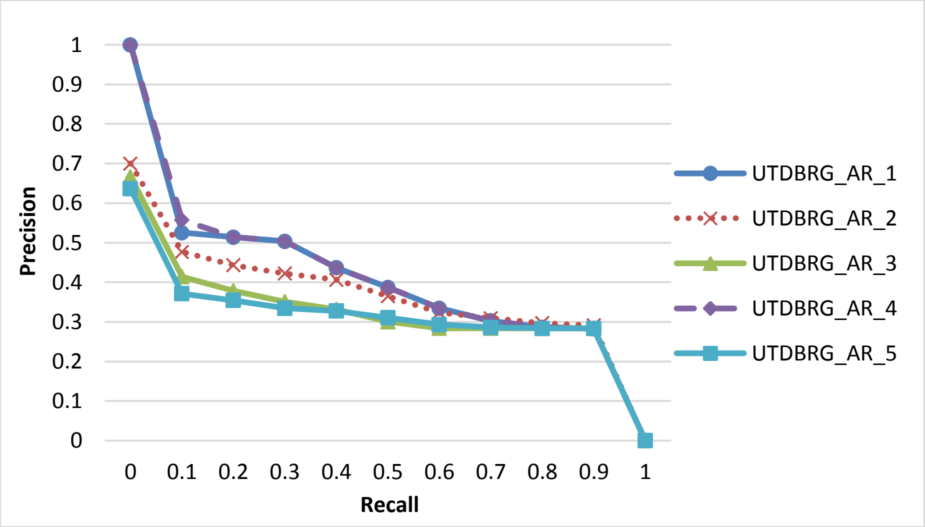

Fig. 2. Precision-Recall comparison of the runs in banking domain

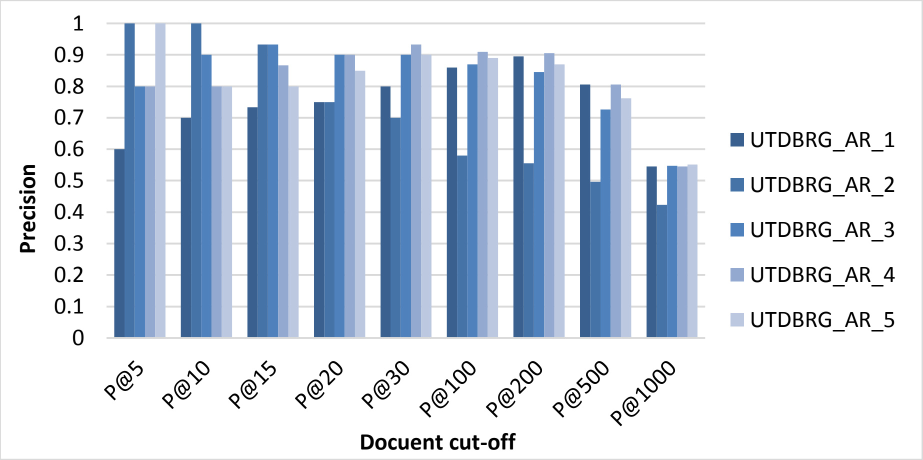

Fig. 3. P@N comparison of the runs in automotive domain

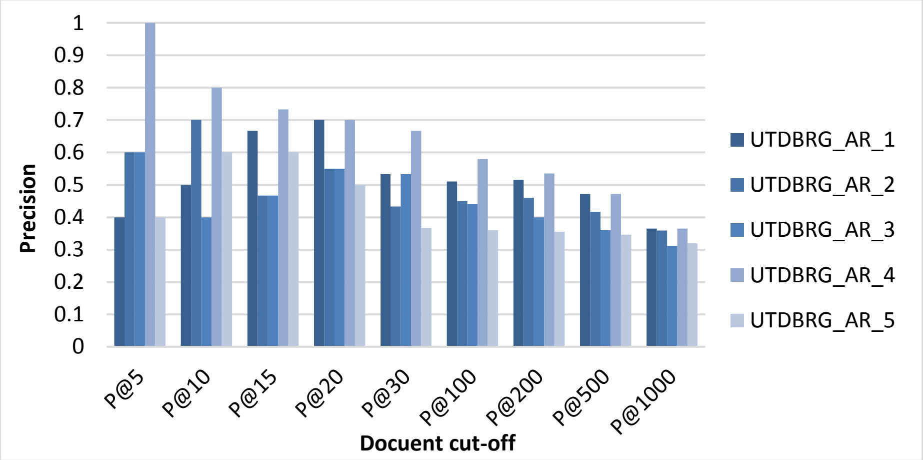

Fig. 4. P@N comparison of the runs in banking domain

5 Further Experiments

It is clear that the topic creation step plays an important role in the final performance of our algorithm. So, we decided to improve the algorithm further by changing the first step of the algorithm. For this purpose two categories of topic sets are created as follows:

- **Grouping hashtags:** The extracted keywords in step 1 of the proposed algorithm are grouped together to form a number of representative topics in each domain. Here is the process:

(i) The algorithm of table 1 is run once. So, some keywords are extracted and weighted in step 1 and step 4 respectively. These keywords are ranked in descending order of their weight.
(ii) The ordered list of keywords is split into different groups and each group is considered as a topic. In other words, each topic consists of a number of hashtags grouped together.

– **Using topic modeling:** Latent Dirichlet Allocation (LDA) [1] is used to create some other topics for each domain. A number of topics are extracted from the training set using LDA for each domain. For this purpose we took advantage of Stanford Topic Modeling Toolbox [2].

After creating the two topic sets, they are combined to form a unique list of topics. Then this list of topics is considered as the output of the first step and in step 2, top 10000 tweets are retrieved. The rest of the algorithm is executed as discussed in table 1. The following table compares the performance of the modified algorithm with the performance of UTDBRG_AR_5 based on MAP in Automotive and Banking domains.

Table 6. Comparison of UTDBRG_AR_5 with the improved algorithm based on MAP

	Improved Algorithm	**UTDBRG_AR_5**
Automotive	0.7833	0.6871
Banking	0.6525	0.3183

Also the following figures compare the performance of the modified algorithm with the performance of UTDBRG_AR_5 based on precision-recall and P@N measures.

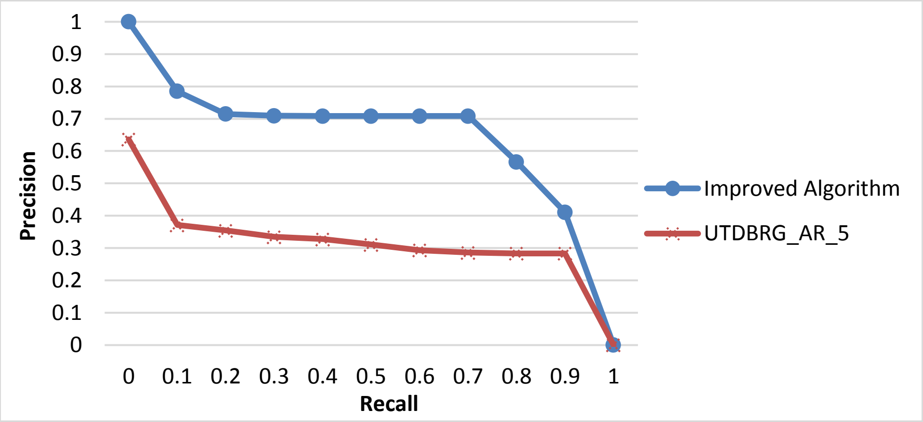

Fig. 5. Comparison of UTDBRG_AR_5 with the improved algorithm based on precision-recall in banking domain

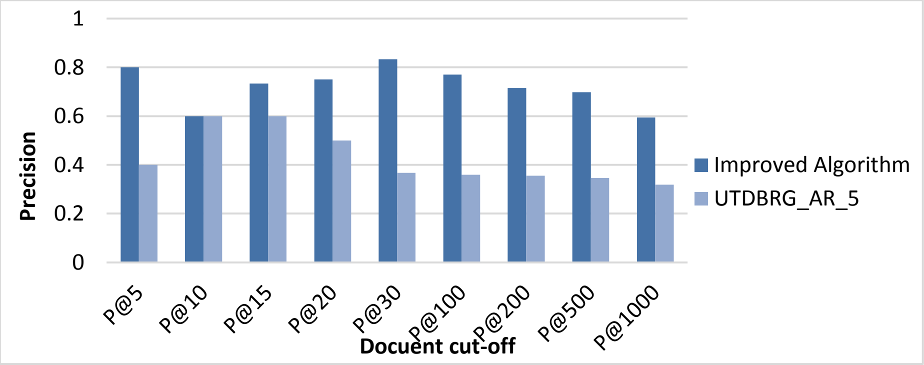

Fig. 6. Comparison of UTDBRG_AR_5 with the improved algorithm based on P@N in banking domain

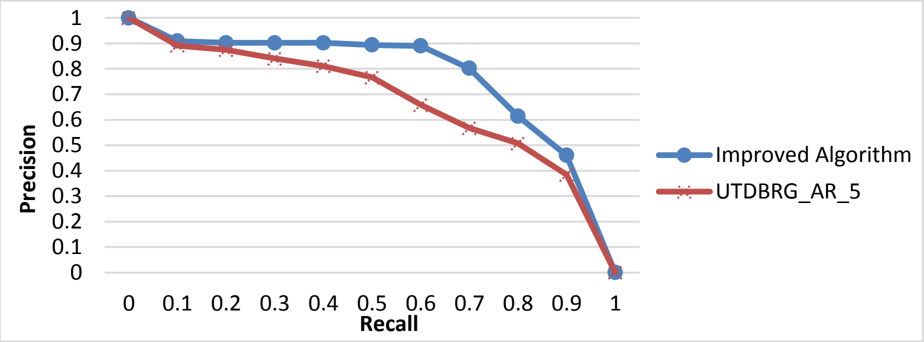

Fig. 7. Comparison of UTDBRG_AR_5 with the improved algorithm based on precision-recall in automotive domain

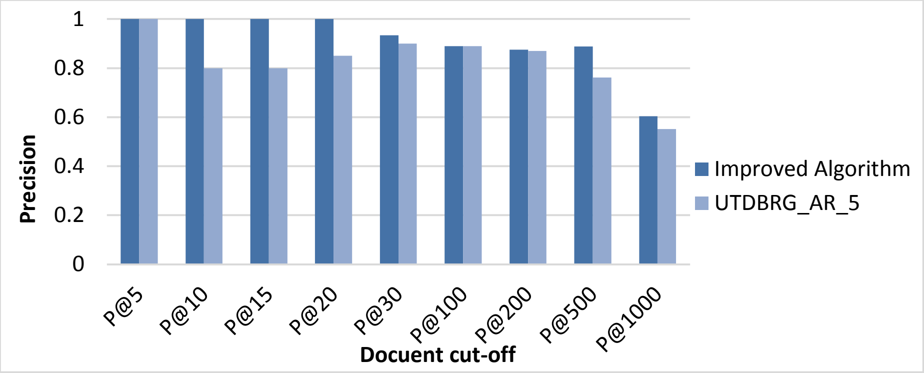

Fig. 8. Comparison of UTDBRG_AR_5 run with the improved algorithm based on P@N in automotive domain

6 Conclusion

In the author ranking task of RepLab2014, we tried to present a new algorithm based on the voting algorithm [8]. The official evaluation results of RepLab 2014 show the proposed algorithm outperforms other algorithms in automotive domain. The topic creation step of our algorithm used simple keywords, so it could not perform well in banking domain that contains more diverse tweets. So, we used topic modeling in addition to tweet hashtags to amend the topic creation step of our algorithm. Evaluation of the improved algorithm shows it works even better than the previous algorithm.

Analysis of our five official runs shows that the fourth run, named UTDBRG_AR_4, performed better than the others. The main reason is usage of more keywords (N=100). Also, it shows that the number of followers is a good feature for detecting influential people. So, we would like to investigate other structural features like people's centrality. Also, it's worth mentioning that we made use of the number of retweets in our experiments but the feature was not helpful. May be the main reason for this fact is that all authors of the collection have more than 1000 followers and the feature is not very discriminative. This feature should be investigated more in future.

7 References

1. David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent dirichlet allocation. J. Mach. Learn. Res. 3 (March 2003), 993-1022.
2. Stanford Topic Modeling Toolbox, http://nlp.stanford.edu/software/tmt/tmt-0.4, Last visited on 5 june 2014.
3. Text REtrieval Conference (TREC) TREC_Eval tool: http://trec.nist.gov/trec_eval, Last visited on 5 june 2014.
4. http://twittercounter.com/pages/100, Last visited on 5 june 2014.
5. http://blog.twitter.com/2013/03/celebrating-twitter7.html, Last visited on 5 june 2014.
6. Modern Information Retrieval, 2ed edition, chapter 3, baeza yates, 2011.
7. Terrier IR Platform version 3.5, http://terrier.org, Last visited on 5 june 2014.
8. Craig Macdonald and Iadh Ounis. 2006. Voting for candidates: adapting data fusion techniques for an expert search task. In Proceedings of the 15th ACM international conference on Information and knowledge management (CIKM '06). ACM, New York, NY, USA, 387-396.
9. Yeha Lee, Seung-Hoon Na, and Jong-Hyeok Lee. 2012. Utilizing local evidence for blog feed search. Inf. Retr. 15, 2 (April 2012), 157-177.