Concept Based Tie-breaking and Maximal Marginal Relevance Retrieval in Microblog Retrieval

Kuang Lu
Department of Electrical and Computer Engineering
University of Delaware
Newark, Delaware, 19716
lukuang@udel.edu

Diego Roa
Systems and Computing Engineering Department
Universidad de los Andes
Colombia
df.roa34@uniandes.edu.co

Hui Fang
Department of Electrical and Computer Engineering
University of Delaware
Newark, Delaware, 19716
hfang@udel.edu

ABSTRACT

There are enormous tweets posted on any given day, and the number keeps increasing. As a result, the needs of effectively retrieving tweets depending upon user's information need, and summarizing tweets pertaining to a given topic have become increasingly important. In this paper, Wikipedia concepts [1] was introduced in tie-breaking to perform ad-hoc microblog retrieval. The Maximal Marginal Relevance (MMR) [2] criterion is deployed to summarize relevant tweets.

1. INTRODUCTION

Microblog, as an extremely popular type of social media, has become increasingly involved in almost everyone's life. The need of finding relevant tweets also has become more prominent. Additionally, summarizing relevant tweets in order to offer users a concrete picture about a certain topic also has become more and more important. This year, a new task, called Tweet Timeline Generation is introduced which specifically focuses on this problem.

For microblog ad-hoc retrieval, as indicated in the previous work [6], tie-breaking seems to be an appropriate solution. This year, we tried to include other new signal into our tie-breaking framework to improve the search performance. The signal we used is "Wikipedia concepts". The basic idea is that, given a query, we find out the Wikipedia concepts mentioned in the query as well as in the candidate tweets, and then use the number of concepts of the tweets as a new signal. Besides tie-breaking, other IR approaches, such as query expansion and learning-to-rank, are also likely to be effective, thus were tried by us.

Considering summarizing several tweets, it is clear that solely using classical IR models, such as Okapi BM25, might not work. The reason is that these models evaluate the relevance of documents and rank them independently. In generating tweet summery, retrieving highly relevant but redundant tweets might not be helpful since redundant information certainly may not be useful for a user.

In this year's track, we continued leveraging tie-breaking strategy [7] for ranking tweets as last year. Moreover, the concepts in tweets were also used as a new retrieval signal to enhance search results. Besides tie-breaking, query expansion and learning-to-rank implemented in Terrier [5] were also deployed as additional ranking method. For the second Tweet Timeline Generation task, Maximal Marginal Relevance [2] based ranking, combined with tie-breaking, were deployed to capture both relevance and novelty.

2. AD-HOC MICROBLOG RETRIEVAL

2.1 Concept Based Tie-breaking

As proposed in [7], tie-breaking is a retrieval method combining retrieval signals in a simple way which explicitly differentiates the impact amongst IR signals. The basic idea is that, from pre-selected candidate IR signals, first choose only one of them to rank the tweets. For documents with the same score, another singal will be used to rank these documents to break the ties, but the relative orders of other documents against these documents remain the same. The tie-breaking step above is repeatedly applied to further break ties until all candidate signals are applied and the ranking is finalized. Ranking tweets in this way, signals that are applied earlier have more decisive impact on the final ranking than those applied latter.

In last year's microblog track [6], only the term frequency (TF), the inverse document frequency (IDF), the document length (DL), the number of followers of the account posted the tweets (NOF) were used and the implementations of them are shown in Table 1. The order of implementation of last year is $IDF \bigoplus TF \bigoplus NOF \bigoplus DL$. We chose this as the baseline method for this year. However, since tweets are short in nature, solely using classic IR signals might not be sufficient. Therefore, this year we incorporate a new signal, which is "Wikipedia concepts" in both queries and tweets. These concepts are extracted by the toolkit Wikimantic [1]. What this toolkit does is that for every string sent to the toolkit, it will detect the Wikipedia concepts related to it. The Wikipedia concepts are defined as the title of Wikipedia articles. We chose to use original query string and tweets as input, and selected the longest non-overlapping concepts detected by Wikimantic as the concepts of queries and

Table 1: Implementation of Signals Given a query Q and a tweet D

Retrieval signal	Implementation
TF(Q,D)	$\sum_{t \in Q \cap D} \frac{c(t,D)}{c(t,D)+1}$
IDF(Q,D)	$\sum_{t \in Q \cap D} log(1 + \frac{N}{df(t)})$
DL(Q,D)	$\frac{1}{dl(D)}$
NOF(Q,D)	$log(NOF(D))$

t : *a term t*
$c(t, D)$: *number of occurances of t in D*
N : *number of tweets in the collection*
$df(t)$: *number of tweets containing term t*
$dl(D)$: *the length of tweet D*
$NOF(D)$: *number of followers of the account that posted D*

Table 2: Ad-hoc Result

Run Method	R-precision	MAP	P@30
UDInfoTB	0.3639	0.3386	0.5394
UDInfoTBRR	0.2594	0.2184	0.5733
UDInfoQE	0.4414	0.4154	0.6115
UDInfoLTR	0.2270	0.1926	0.2455

tweets. The reason we chose to include this signal is that the concepts, in our opinions, are more informative and can more accurately represent the information of tweets and queries. Therefore, using it could result in higher precision, meaning reducing the number of retrieved non-relevant tweets only containing query terms instead of containing query concepts. The way we implemented this signal is shown in Figure 1: Given a query and a document, we use Wikimantic to detect the number of Wikipedia concepts of them; count the number of matching concepts between the query and the tweet; and use the number of matching concepts as the signal value (denoted as CM).

After Wikipedia concept was decided to be used, the following step should be deciding how it should be applied in the tie-breaking framework. Since the Wikipedia concepts are widely recognized by a considerable number of Wikipedia users, we thought the Wikipedia matching concepts between queries and tweets (CM) should be viewed as the most important signal, and thus we planed to apply it first. However, since it took a very long time to compute the concepts in all tweets, we only used it to re-rank the top 100 tweets of the baseline run to generate another run. After the submission, we were able to compute the CM signal for all tweets and include it as the first signal in the tie-breaking framework as originally planned to test its effectiveness, which will be discussed later. In addition to tie-breaking based runs, we also tried two runs using the $Terrir$ [5] toolkit to explore its built-in learning-to-rank and query-expansion methods.

2.2 Experiment Setup

The tweet pool is generated by retrieving tweets through the API[1] provided by the organizers. Specifically, for each original query, we searched it on the Yahoo! search engine, which returned some query suggestions for the query. The original queries as well as their query suggestions were used to search against the API, and the number of results required to return was set to 10,000. The returned tweets were then used to build tweet collection for each original query respectively. The maximum id numbers were set to be the same as original queries in order to prevent retrieving feature tweets (posted after the query time), which will be deemed irrelevant.

Since non-English tweets will be treated as irrelevant regardless of whether it is relevant or not, we filtered the non-English tweets before building tweet index. In order to detect non-English tweets, the python package *lang_id.py* [4] was used to identify the language of tweets. Only the tweets detected by the tool as English were kept, or otherwise discarded.

2.3 Results and Analysis

We submitted four runs for the ad-hoc task this year. The results are reported in Table 2:

- **UDInfoTB:** The baseline run same as last year[6], which is the tie-breaking with the order $IDF \bigoplus TF \bigoplus NOF \bigoplus DL$.
- **UDInfoTBRR:** In this run, we re-ranked the top 100 tweets of the results of $UDInfoTB$ based on the CM signal.
- **UDInfoLTR:** The learning-to-rank implemented in $Terrir$ [5] was used to generate this run. The signals we used were $BM25$, $PL2$ [3] and TF_IDF.
- **UDInfoQE:** The built-in pseudo relevance feedback based query expansion tool in $Terrir$ was used to generate this run. Top 3 documents of each query were used to extract 10 most informative terms to expand the oroginal query.

After the evaluation was out, we also tested how useful the CM signal is in the tie-breaking framework. First, we added it to our baseline run's framework as the first signal to be applied to examine the usefulness of it (tie-breaking with the order $CM \bigoplus IDF \bigoplus TF \bigoplus NOF \bigoplus DL$). Thereafter, we tried different orders of signal application of theis experiment to test whether the performance could be improved. The results of the experiments are shown in Table 3. As can be seen, the new signal CM is helpful in enhancing the performance (runs with this signal is better than the UDInfoTB run, which does not have the signal) and using a more appropriate order could further improve the performance.

3. TWEET TIMELINE GENERATION

[1]https://github.com/lintool/twitter-tools/wiki/TREC-2013-API-Specifications

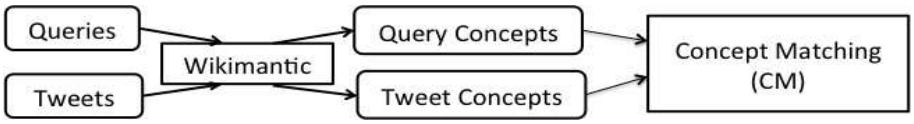

Figure 1: Wikipedia Concept Matching Implementation

Table 3: Performance of Different Signal Implementation Order

Run Method	R-precision	MAP	P@30
$IDF \bigoplus TF \bigoplus NOF \bigoplus DL$ ($UDInfoTB$)	0.3639	0.3386	0.5394
$CM \bigoplus IDF \bigoplus TF \bigoplus NOF \bigoplus DL$	0.3732	0.3393	0.5570
TF $\bigoplus$ CM $\bigoplus$ IDF $\bigoplus$ NOF $\bigoplus$ DL	**0.3963**	**0.3701**	**0.5945**

3.1 Maximal Marginal Relevance

As mentioned previously, solely using existing IR model might not be helpful in the Tweet Timeline Generation (TTG) task. Intuitively, a good summery of a topic should cover as many distinct aspects of the topic as possible, meaning that returning redundant information about only few aspects of the topic might not be favorable. Unsurprisingly, according to this year's evaluation metrics for the TTG task, returning redundant tweets will actually decrease the evaluation score (F-measure based) of a result list. Therefore, for the idea result list, every tweet should be relevant to the original query, and each of them should carry some novel information that is not mentioned in any other tweets.

We noticed that the classical Maximal Marginal Relevance [2] framework might be suitable for our task. The MMR framework computes the ranking score of a tweet according to (1) the relevance score of it against the original query, and (2) the maximal similarity score between the current tweet and the tweets already retrieved. Notice that the second part explicitly captures how much novel information a tweet brings with regard to already retrieved tweets. By using MMR based ranking, novelty and relevance are both considered when deciding whether a tweet should be retrieved or not, which will, ideally, produce a tweet list that fits the TTG task.

Based on MMR, we built a general process for all of our runs. First, use the top 50 tweets for each query of the run $UDInfoQE$ in the first task as the candidate tweets since we were most confident with this run. Second, use all the first-ranked tweets of each query as part of the result. Third, we slightly modified the MMR scoring function in [2] by using the same method to compute query-tweet and tweet-tweet similarity as the ranking method to iteratively select the highest ranked tweet for each query into the result. The ranking function can be described as Equation 1:

$$MMR \stackrel{def}{=} \underset{D_i \in R/S}{\text{argmax}}[\lambda(Sim(D_i, Q) - (1-\lambda)\max_{D_j \in S} Sim(D_i, D_j))] \quad (1)$$

In the equation, R is the whole tweet collection; S is the tweets set already retrieved; $R \backslash S$ is the set of tweets in R not yet retrieved; Q is the query a user searches about; Sim is the measurement for query-tweet and tweet-tweet similarity; λ is the predefined constant coefficient. We planed to use tie-breaking as the similarity measure. However, tie-breaking cannot offer numerical similarity scores, which are required by the MMR ranking function. We overcame it by using Equation 2, 3 to compute similarity scores:

$$Sim_i(Q, D) = \begin{cases} 0, & \text{i=0} \\ \mu * Sim_{i-1}(Q, D) + S_i, & \text{i=1,2,...,k} \end{cases} \quad (2)$$

$$Sim(Q, D) = Sim_k(Q, D) \quad (3)$$

Given a query Q, a tweet D, and a set of k signals, we denote the total score after the ith signal is applied as $Sim_i(Q, D)$ and let $Sim_0(Q, D)$ be zero. The score computed only for the ith signal is denoted as $S_i(Q, D)$. Before ith signal is applied, we multiply the ranking score for the signals applied earlier ($S_{i-1}(Q, D)$) with a big factor μ (set to 10000) so that the effect of signals applied latter cannot invert that of the signals applied ealier. After the score of current signal is computed, it is added to the amplified score described above to finally get $Sim_i(Q, D)$. After all k signals are applied, the $Sim_k(Q, D)$ is used as the similarity score of the tweet ($Sim(Q, D)$). In this way, the ranking will be the same as that produced by the normal tie-breaking described in 2.1 and we are still able to obtain similarity scores. Finally, the tweet selection process will stop according to pre-defined stopping criteria, which could be different among different runs.

3.2 Results and Analysis

According to the framework described previously, we came up with 4 runs. The differences among the runs lie on the similarity measures of the MMR ranking function and stopping criteria. The results of submitted 4 runs are shown in Table 4:

- **UDInfoMMRA:** Use Equation 2, 3 to compute the similarity scores with signals and applying

Table 4: TTG Results

Run Method	Unweighted Recall	Weighted Recall	Precision
UDInfoMMR5	0.0743	0.2035	0.5055
UDInfoMMRA	0.0652	0.1806	0.5314
UDInfoMMRWC5	0.0900	0.2191	0.5709
UDInfoMMRWCA	0.0852	0.2010	0.5919

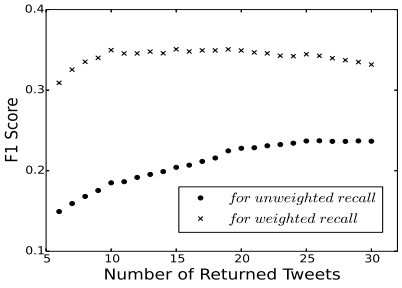

Figure 2: F1 Score Increases When Returning More Tweets for $UDInfoMMRWC5$ run

order as $IDF \bigoplus TF \bigoplus NOF \bigoplus DL$. The stopping criterion was set to be when the score difference between two consecutively retrieved tweets is higher than 30% of the one retrieved earlier.

- **UDInfoMMR5:** Same similarity measure as $UDInfoMMRA$. The stopping criterion is returning 5 tweets for every query.
- **UDInfoMMRWCA:** The similarity measure is similar to $UDInfoMMRA$ with signals and applying order as $CM \bigoplus IDF \bigoplus TF \bigoplus NOF \bigoplus DL$ mentioned in 2.1. The stopping criterion is when the score difference between two consecutively retrieved tweets is higher than 1% of the one retrieved earlier.
- **UDInfoMMRWC5:** Use the same similarity measure as $UDInfoMMRWCA$. The stopping criterion is returning 5 tweets for every query.

As can be seen, we achieved high precision but relatively low recall. It is expected since we chose to only use 50 tweets as the candidate tweet pool for each query; return very few results for each query; and the dominating signal CM, which was implemented as the first signal, is likely to be helpful in improving precision with some sacrifice on recall. Therefore, after evaluation was out, we tried to increase the number of returned tweets for out best run $UDInfoMMRWC5$ in order to examine whether the performance would be improved. The result of this experiment is shown in Figure 2. As can be seen, the F1 score is indeed improved after returning more tweets.

4. CONCLUSION

Tie-breaking is an effective way of combining different retrieval signals in a simple way. We have explored the potential of it this year by adding a new signal (Wikipedia concepts) and there is still room for exploration: different signals, different signal implementations, or even combining it with other IR techniques such as query expansion could be further experimented. For the Tweet Timeline Generation task, even though we did not achieve high recall since we aimed at achieving high precision, relaxing the stopping criteria can help us achieve better results. Thus, we are likely to explore the MMR approach in the TTG task in the future.

5. REFERENCES

[1] C. Boston, H. Fang, S. Carberry, H. Wu, and X. Liu. Wikimantic: Toward effective disambiguation and expansion of queries. *Data & Knowledge Engineering*, 90(0):22 – 37, 2014. Special Issue on Natural Language Processing and Information Systems (NLDB 2012).

[2] J. Carbonell and J. Goldstein. The use of mmr, diversity-based reranking for reordering documents and producing summaries. In *Proc. of SIGIR '98*.

[3] B. He and I. Ounis. Term frequency normalisation tuning for bm25 and dfr models. In *Proceedings of the 27th European Conference on Advances in Information Retrieval Research*, ECIR'05, pages 200–214, Berlin, Heidelberg, 2005. Springer-Verlag.

[4] M. Lui and T. Baldwin. langid.py: An off-the-shelf language identification tool. In *ACL 2012 System Demonstrations*, pages 25–30.

[5] I. Ounis, G. Amati, V. Plachouras, B. He, C. Macdonald, and D. Johnson. Terrier information retrieval platform. In *Proceedings of the 27th European Conference on Advances in Information Retrieval Research*, ECIR'05, pages 517–519, Berlin, Heidelberg, 2005. Springer-Verlag.

[6] Y. Wang, J. Darko, and H. Fang. Tie-breaker: A new perspective of ranking and evaluation for microblog retrieval. In *Proceedings of the 2014 TREC conference*, 2014.

[7] H. Wu and H. Fang. Tie breaker: A novel way of combining retrieval signals. In *Proc. of ICTIR'13*, 2013.