Ranking Model Selection and Fusion for Effective Microblog Search

Zhongyu Wei[1] * , Wei Gao[2], Tarek El-Ganainy[2], Walid Magdy[2], Kam-Fai Wong[1,3,4]

[1]The Chinese University of Hong Kong, Shatin, N.T., Hong Kong
{zywei, kfwong}@se.cuhk.edu.hk
[2]Qatar Computing Research Institute, Qatar Foundation, Doha, Qatar
{wgao, telganainy, wmagdy}@qf.org.qa
[3]MoE Key Laboratory of High Confidence Software Technologies, China
[4]Shenzhen Research Institute, The Chinese University of Hong Kong

ABSTRACT

Re-ranking was shown to have positive impact on the effectiveness for microblog search. Yet existing approaches mostly focused on using a single ranker to learn some better ranking function with respect to various relevance features. Given various available rank learners (such as learning to rank algorithms), in this work, we mainly study an orthogonal problem where multiple learned ranking models form an ensemble for re-ranking the retrieved tweets than just using a single ranking model in order to achieve higher search effectiveness. We explore the use of query-sensitive model selection and rank fusion methods based on the result lists produced from multiple rank learners. Base on the TREC microblog datasets, we found that our selection-based ensemble approach can significantly outperform using the single best ranker, and it also has clear advantage over the rank fusion that combines the results of all the available models.

Categories and Subject Descriptors

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

Keywords

Microblog search; Twitter; ranker selection; rank fusion; aggregation; re-ranking

1. INTRODUCTION

In recent years, microblogging services witnessed an increase in popularity on the Internet. For example, Twitter has billions of online users who exchange information of interest everyday in the form of short messages called tweets each within 140 characters. Because of the timely fashion of tweets, breaking news or current events are captured and propagated faster over this platform than the traditional news feeds on the Web. Therefore, users are willing to search over the enormous collection of online microblogs to satisfy their information needs for on-going topics.

However, topical ad-hoc search so far is not the most popular search behavior on Twitter. Human factor study [29] found that Twitter users mainly perform search to get updates about some entities or celebrities, find friends, get insight about certain hashtags, and so on. This is not only because of the social nature of the service, but also due to the generally low quality of microblogs. The latter becomes a big obstacle for ad-hoc search since the strict (short) length limit and the colloquial form of expressions in the posts can result in serious word mismatch problem.

It has been found that in general two people use the same term to describe the same concept in less than 20% of times [6]. Word mismatch problem is more severe for short casual queries (like microblog queries) than for long elaborate ones [32]. If documents are very brief such as tweets, the risk of query terms failing to match words observed in relevant documents would be even larger [7]. The problem does not only have the effect of hindering the retrieval of relevant documents, but also naturally produces bad rankings of retrieved relevant documents [5].

In microblog search, some techniques such as query or document expansion have been used to address word mismatch for providing better retrieval effectiveness, and among others, reports showed that the ranking models learned from various relevance features for re-ranking top retrieval results can typically improve the final results [9, 15, 22]. However, all of the reported re-ranking methods merely focused on feature engineering, and none of them on ranking technique itself. Also, the applied models are all based on a single ranker which is typically query-insensitive and may not be universally suitable for different types of queries.

In this paper, we study how to improve re-ranking microblog search results by leveraging ranked list from multiple rankers. We examine some state-of-the-art post-retrieval re-ranking approaches and their variants: (1) We choose the best single ranking model among all the candidate models; (2) For each query, we select the best performed ranking model from the candidate models in a query-sensitive manner; (3) We aggregate the ranked lists of all the candidate models available using different fusion techniques; (4) Instead of selecting the single best ranker for each query or fusing the results of all candidate models, we explore different fusion techniques to combine the outputs of top-k ranking models selected in a query-by-query basis. We compare these approaches

*This work was conducted when the author did his internship at Qatar Computing Research Institute

SoMeRA'14, July 06 - 11 2014, Gold Coast , QLD, Australia.
Copyright 2014 ACM 978-1-4503-3022-0/14/07 ...$15.00.
http://dx.doi.org/10.1145/2632188.2632202.

based on TREC Microblog datasets. Experimental results show that the query-sensitive selection together with the ensemble of multiple rankers can achieve statistically significant improvements over baselines.

2. RELATED WORK

Several studies have investigated the nature of microblog search compared to other search tasks. Naveed et al. [24] illustrated the challenges of microblog retrieval, where documents are very short and typically focused on a single topic. Teevan et al. [29] highlighted the differences between microblog queries and Web search queries: firstly, microblog queries represent users' interest to find updates about a given event or person as opposed to relevant pages on a given topic in Web search; secondly, the length of microblog queries are much shorter (with only 1.64 words on average) as compared to that of Web queries (with 3.08 words on average).

TREC introduced a track for ad-hoc microblog search starting from 2011 [25, 28, 18]. Many different approaches were proposed while only a few of them presented good retrieval effectiveness. Typical methods could be summarized as using query or document expansion for retrieval, performing post-retrieval re-ranking based on various relevance features, or the combination of both. Among the effective approaches, many of them used learning to rank algorithms [19] for re-ranking [9, 22, 15]. However, these works only focused on feature engineering and none of them examined the ranking techniques more deeply. They suffered from the following issues: (1) some work had very small training set that is not sufficient to learn a powerful ranker [9]; (2) some of them used only a very limited feature set with just 10 or so features [22, 15]; (3) all of them simply employed a single ranking model which is query-insensitive. There leaves much room for further improvement by using more sophisticated techniques.

3. RANKER SELECTION AND FUSION

To improve the effectiveness of re-ranking of retrieved tweets, we present three considerations different from previous work: (1) Instead of employing only one ranking model, we can resort to multiple ranking models and combine the results produced from them for re-ranking; (2) The model selection could be query-sensitive, aiming to choose the multiple top rankers in a unsupervised query-by-query basis for improving the re-ranking for the entire topic set; (3) A metasearch fusion algorithm can be adopted to aggregate the preferences of multiple ranking models based on either the selected top ranking models or all available ranking models.

Inspired by the supervised model selection strategy [26], we propose a re-ranking system that allows to select multiple ranking models and combine their results on a per-query basis. Suppose we have learned a set of rankers $R = \{R_b, R_1, ..., R_m\}$ where R_b is a *base ranker* that produces an initial ranked list, such as the language-modeling-based IR model or query expansion based on it, and the other m are *candidate rankers* for re-ranking the initial results. For an unseen test query q', we want to select some most effective candidate rankers from $R - \{R_b\}$ for q'.

Given the test query q', we first select L nearest training queries from the training query set $Q = \{q_1, q_2, ..., q_n\}$ to predict the performance of each candidate ranker on ranking the retrieved tweets of q'. Then we choose the top-k best candidate rankers based on their performance estimated for q', and combine the ranked lists produced by them to re-rank the tweets. Therefore, the system includes two main components, i.e., ranker performance prediction for model selection and rank aggregation for the selected rankers. Figure 1 shows the architecture of the proposed system.

3.1 Ranking Model Selection

For identifying some nearest neighbors of q', we extend the model selection method described in [26]. Our extension makes two major progresses over theirs: (1) Not using the KL-divergence [16] between the ranking scores of the current candidate ranker and the base ranker, we utilize the divergence scores between ranking scores obtained from the current candidate ranker and all other rankers (including the base ranker) to form a vector for identifying some training queries similar to q'; (2) Instead of choosing only top-one ranker, for each query, we choose multiple top candidate rankers with highest estimated performance scores on that query and aggregate their results for re-ranking.

A divergence vector is obtained for assessing the similarity between q' and the training queries. For any query q and a candidate ranker R_i, let the vector $\mathbf{D}(R_i, q)$ denote a distribution of divergence values between the ranked list of q produced by R_i and those by other rankers in R, which is presented as

$$\mathbf{D}(R_i, q) = [D(R_b||R_i, q), D(R_1||R_i, q), ..., D(R_m||R_i, q)]$$

where each element is the normalized divergence score between two specific rankers over the retrieved tweets of q, $D(R_b||R_i, q)$ indicates the extent that R_i can alter the order of the initial ranking, and the rest of the elements indicate the divergence between the ranked lists of two candidate rankers. The normalized divergence score is computed as $D(R_j||R_i, q) = \frac{1}{Z}\sum_t |s_j(t) - s_i(t)|$, where $s_i(t)$ is the ranking score of tweet t provided by ranker R_i, and Z is the normalization constant so that the sum of all elements in $\mathbf{D}(R_i, q)$ equals to 1 (so that $\mathbf{D}$ becomes a distribution).

Based on the vectors of divergence distribution, the similarity between the unseen query q' and any training query $q \in Q$ can be computed as negative KL-divergence between $\mathbf{D}(R_i, q')$ and $\mathbf{D}(R_i, q)$, that is, $sim(q', q) = -\mathcal{KL}\left(\mathbf{D}(R_i, q'), \mathbf{D}(R_i, q)\right)$. According to the similarity scores, we choose L training queries from Q that are closest to q', denoted as $\{q^{(l)}|q^{(l)} \in Q; l = 1, 2, \ldots, L\}$.

Let $ps(q^{(l)}, R_i)$ be the evaluation performance score of R_i obtained for the neighboring query $q^{(l)}$, and then the performance score of R_i for q' can be estimated as $\frac{1}{L}\sum_{l=1}^{L} ps(q^{(l)}, R_i)$. Therefore, we can select top-k ranking models according to the estimated performance scores and then aggregate their ranking results in a query-sensitive way. Note that the L training queries and k best models are query-dependent and the parameters L and k can be fixed during training.

3.2 Rank Aggregation

Given multiple ranked lists resulting from the selected models, we can combine these results by using rank fusion methods to aggregate these individual lists. Popular fusion models are estimated based on either relevance score or rank of the results or both of them. In this work, we investigate four representative fusion techniques that are shown effective in general information retrieval tasks. However, their effectiveness is yet unclear in the specific fusion task for microblog search. The fusion can be applied either overall all available candidate rankers or on the top-k candidate rankers selected in query-dependent manner.

CombMNZ [11] is a traditional and effective fusion method, where the final score of a tweet is calculated as the sum of its relevance scores received from different rankers weighted by the number of rankers that "retrieved" it: **CombMNZ**$(t) = |\{r \in R'|rank_r(t) \leq c\}| \times \sum_r score_r(t)$, where R' is the set of rankers used for combination, c is the cut-off rank, $rank_r(t)$ and $score_r(t)$ are the rank and the relevance score of tweet t given by ranker r, respectively. Note that c is used to control how deep we want to look into the ranked lists (by assuming that the items ranked below the

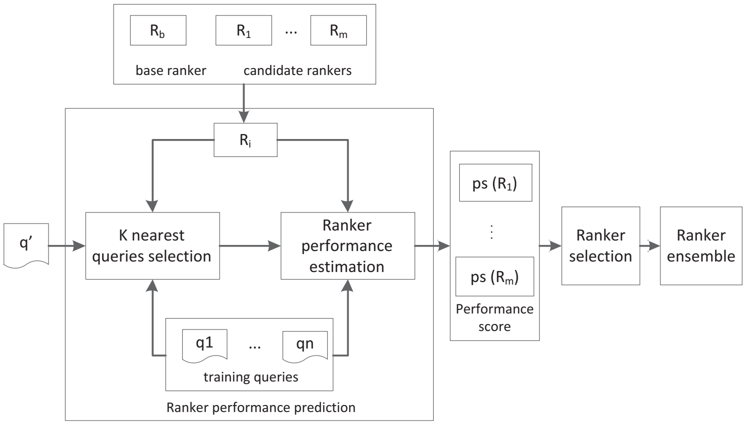

Figure 1: The architecture of our query-sensitive ranker selection and ensemble method

threshold are not retrieved). CombMNZ will become CombSUM when we do not consider the cut-off rank. CombMNZ was found useful in the sense that different runs "retrieve" similar set of relevant documents but different set of non-relevant documents [17].

Weighted Borda-fuse [1] and **weighted Condorcet-fuse** [23] are two voting-based fusion algorithms that derive the final scores by weighted ranks across the given ranking models or by counting the number of pairwise wins by majority vote among the ranked lists. Note that the former is based on pointwise vote while the latter is pairwise. These two were considered standard fusion methods in metasearch due to their effectiveness. For both approaches, we used the mean average precision (MAP) of the list as the weight of the corresponding ranking model following [1, 23].

Reciprocal rank fusion [4] sorts the tweets according to this formula: $RRF(t) = \sum_{r \in R'} \frac{1}{\kappa + rank_r(t)}$, where κ is a constant used to mitigate the impact of high-ranked tweets by outlier rankings. The intuition of the formula is to reduce the influence of documents ranked unreasonably high while giving chance to lower-ranked documents to influence. It was shown state-of-the-art effectiveness in TREC ad hoc task as well as Web search task [4].

3.3 Base Rankers

We provide two base retrieval models (i.e., base rankers) under different settings (The re-ranking was done to reorder the retrieved tweets returned by the two base rankers). Beside a language-model-based retrieval model [27] denoted as LM, we also adopt a significantly improved base ranker based on pseudo-relevance feedback method using Web search results [10] denoted as $\text{LM}_{\textbf{webprf}}$.

The main challenge in finding relevant tweets to a given topic is word mismatch between search query and tweet text. Many TREC reports in Microblog track showed that query expansion helps in improving the microblog retrieval effectiveness since it enriches the query with additional terms that lead to better matching with more relevant tweets [25, 28, 18]. The base model $\text{LM}_{\textbf{webprf}}$ utilizes web search results as external resource to find concurrent information about the search topic which is proven both efficient and effective for query expansion [8, 10]. For completeness, here we provide some details of the process which is shown in Figure 2 and described stepwise as follows:

- The original query Q_0 is used to search in the tweets collection in an initial step. The most frequent n_t terms (excluding stop words) appearing in the top retrieved n_D tweets are extracted in a standard PRF process [33]. Extracted expansion terms are denoted as Q_{PRF}.

- Q_0 is used to search the Web via search engine in the same time frame of the query for the concurrent results, in which we extract two types of information: (1) The title of the topmost search result is extracted and pruned by removing stop words and website name. The title part usually contains delimiters like '-' and '|' that separate the real title content and the domain name of the webpage, e.g., "... | CNN.com", "... - Wikipedia, the free encyclopedia". Only the real title is used for expansion, referred to as Q_{title}. (2) Both titles and snippets of the top-10 ranked results are collected. Then all terms appearing more than n_w times are extracted and used for expansion, referred to as Q_{web}.

- All expansion terms are combined and appended with a given weight to the original query as follows: $Q_{exp} = (1-\alpha)Q_0 + \alpha(Q_{PRF} \cup Q_{title} \cup Q_{web})$, where Q_{exp} is the final expanded query used for searching tweets at the second time and α is the weight assigned to the expansion terms.

The final formulated query Q_{exp} is expected to be richer in information about the topic than the original query, and potentially leads to better search results. We empirically set the parameters $\alpha = 0.2$, $n_D = 50$, $n_t = 12$, and $n_w = 3$. The details of the parameter tunning process are reported in [10].

3.4 Candidate Rankers

We employ six learning to rank algorithms as the candidate rankers for selection and fusion: RankNet [3], RankBoost [12], Coordinate Ascent [21], MART [13], LambdaMART [31] and RandomForests [2] using RankLib package[1]. Based on these algorithms, we train eight rankers: (1) A Rankboost model is trained without validation set; (2) A MART model is learned using 80% training queries for training and 20% training queries for validation; (3) A RandomForest model is learned in the same way as (2); (4) A RankNet model is learned in the same way as (2); (5) Two Coordinate Ascent models are learned in the same way as (2) but one of them optimizes MAP and the other optimizes P@30; (6) Two LambdaMART models are learned in the same way as (5).

[1] http://sourceforge.net/p/lemur/wiki/RankLib/

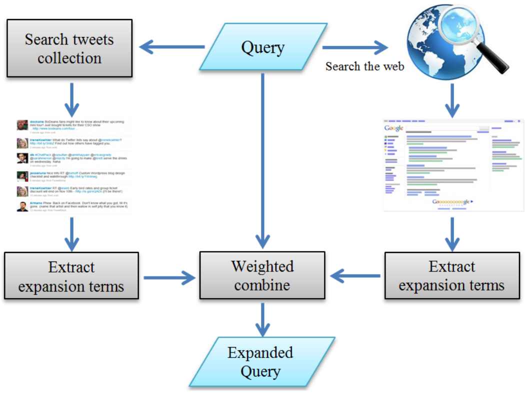

Figure 2: Web search results based query expansion approach

3.4.1 Feature Design

To learn these candidate rankers, we define a set of 19 ranking features belonging to three categories by referring to [22, 9, 15], including content-based features, Twitter-specific features, and user-based features. Recent study [29, 30, 14] showed that people often search Twitter to find temporally relevant information, such as current events, trending topics and real-time information. Considering the importance of time factors, we add two temporal features described as follows:

- **Recency_Degree** indicates whether the post is published recently according to the query time: $Recency\_Degree = Time_{query} - Time_{post}$, where $Time_{query}$ and $Time_{post}$ stand for the time stamps (in millisecond) the query is issued and the tweet is posted, respectively.

- **Is_Peak** is a binary feature indicating whether the target tweet is posted at the peak time of the queried topic. Peak-finding algorithm [20] is used to identify the peak time for the query. Following the strategy used in the real-time tweet search system [14], we apply peak-finding for the top 1000 search results and treat the first and second largest peaks as the real peaks of the query.

4. EVALUATION

We evaluate our approach using TREC ad-hoc microblog search task[2] which was initiated from 2011. Two different tweets collections and three sets of queries have been released so far. The tasks of the first two years share the same collection Tweets2011 which contains 16,141,809 tweets. Then a much larger collection Tweets2013 containing 243,271,538 tweets was newly constructed. There are 3 different query sets, one for each year, which are denoted as QS2011, QS2012 and QS2013 containing 50, 60 and 60 queries, respectively. The statistics of these two tweets collections and relevance judgement of query sets are shown in Table 1 and Table 2, respectively. Following the track benchmark, we report P@30 as the major evaluation metric, and we will also report mean average precision (MAP) for reference.

4.1 Experimental Setting

We trained the eight candidate rankers using the following setup: for test on TREC2011 data, we used 2012 data for training; for test on 2012 data, we used 2011 data for training; and for test on 2013 data, we trained the models using both 2011 and 2012 data. All the parameters of model selection (such as L and k for each query) and fusion (such as c and κ) were validated using 20% of the corresponding training data.

Table 1: The statistics of tweets collections

Collection	# of tweets	# of terms	Average length
Tweets2011	16,141,809	155,562,660	9.64
Tweets2013	243,271,538	2,928,041,436	12.04

Table 2: The statistics of relevance judgement

Query set	# of queries	# of annotated tweets	# of relevant
QS2011	50	40,855	2,864
QS2012	60	73,073	6,286
QS2013	60	71,279	9,011

We implemented the following re-ranking schemes for systematic comparison: (1) *BestSingle*: Use the single best ranker among all the candidate rankers in a query-insensitive way like a common existing approach; (2) *PMO*: Apply the model selection method by Peng et al. [26] that chooses the best single ranker for each query; (3) *Best-sel*: Choose the best single ranker for each query using our extension of the model selection method (see Section 3.1); (4) *CMNZ-all*, *Borda-all*, *Condorcet-all*, and *RRF-all*: Combine all the eight available candidate rankers using CombMNZ, weighted Borda-fuse, and weighed Condorcet-fuse, and Reciprocal Rank Fusion, respectively; (5) *CMNZ-sel*, *Borda-sel*, *Condorcet-sel*, and *RRF-sel*: Combine our selected top rankers (see Section 3.1) using the four corresponding fusion models. We also report the performance of best systems of TREC in each year from 2011 to 2013 for comparison.

4.2 Results and Discussions

Tables 3 and 4 present the re-ranking results using our methodology compared to the baselines and different re-ranking schemes. As shown, the LM achieved an average-level score compared to other results in the microblog track, while the LM$_{\textbf{webprf}}$ achieved among the highest scores of automatic runs according to the TREC reports [25, 28, 18]. We aim to examine how much performance gain different re-ranking techniques could obtain over these two baselines whose performances have such a large gap, to justify the effectiveness of model selection and fusion. Based on the results of all the six groups of experiments in the two tables (based on the three-year data in each table), we have the following findings:

– Almost all results show that re-ranking can improve the search results of the two base retrieval models that have large performance gap. So re-ranking is generally a right direction to go. But the effectiveness varies considerably with different re-ranking approaches. Overall, our query-sensitive model selection and fusion (denoted as the "*x-sel*" rows) consistently outperforms other re-ranking schemes according to P@30 values.

– *BestSingle* made significant improvements over the baseline in only two groups of results in terms of P@30. *PMO* performs even worse than *BestSingle* in five groups of results on P@30 although not significantly worse. This is because *PMO* only uses the base ranker for calculating query similarity which is not fine-grained or accurate. The overall performance is improved a little, but not significantly better than *PMO*, by using our extension *Best-sel* that resorts to all candidate rankers for query similarity assessment. Overall, using a single ranker for re-ranking has its limitation according to the results.

[2] `https://sites.google.com/site/microblogtrack/`

Table 3: Re-ranking results base on LM (*Italic*: diff. with LM p <0.05; Bold: diff. with LM p <0.01; *: diff. with PMO p <0.05; $\sharp$: diff. with BestSingle p <0.05; $\dagger$: diff. between best x-sel and best x-all p <0.05; Underline: max value)

	TREC2011		TREC2012		TREC2013	
	P@30	MAP	P@30	MAP	P@30	MAP
LM	0.4231	0.3897	0.3559	0.2329	0.4700	0.2731
BestSingle	***0.4673***	$\underline{\boldsymbol{\mathit{0.4015}}}$	***0.3887***	0.2399	0.4867	$\underline{0.2803}$
PMO [26]	***0.4633***	0.3873	*0.3814*	$\underline{\mathit{0.2406}}$	0.4833	0.2641
CMNZ-all	***0.4510***	0.3621	*0.3814*	0.2326	0.4878	0.2664
Borda-all	***0.4483***	0.3646	*0.3819*	0.2362	0.4800	0.2708
Condorcet-all	***0.4633***	0.3925	***0.3870***	0.2391	0.4828	0.2759
RRF-all	***0.4558***	0.3672	***0.3915***	0.2376	0.4844	0.2732
Best-sel	***0.4633***	0.3940	***0.3859***	0.2337	*0.4900*	0.2746
CMNZ-sel	***0.4653***	0.3733	***0.3932****	0.2374	*0.4922*	0.2733
Borda-sel	***0.4633***	0.3940	***0.3859***	0.2337	0.4894	0.2669
Condorcet-sel	$\underline{\boldsymbol{\mathit{0.4701}}}$	0.3784	$\underline{\boldsymbol{\mathit{0.3960}}}^{*}$	$\underline{\mathit{0.2406}}$	$\underline{\mathit{0.4933}}$	0.2753
RRF-sel	***0.4633***	0.3940	***0.3927****	0.2363	*0.4917*	0.2622

Table 4: Re-ranking results based on $\mathrm{LM_{webprf}}$ (*Italic*: diff. with $\mathrm{LM_{webprf}}$ p <0.05; Bold: diff. with $\mathrm{LM_{webprf}}$ p <0.01; *: diff. with PMO p <0.05; $\sharp$: diff. with BestSingle p <0.05; $\dagger$: diff. between best x-sel and best x-all p <0.05; Underline: max value)

	TREC2011		TREC2012		TREC2013	
	P@30	MAP	P@30	MAP	P@30	MAP
$\mathrm{LM_{webprf}}$	0.4905	0.4651	0.4356	0.2960	0.5350	0.3454
BestSingle	0.5075	0.4611	0.4514	0.2971	0.5494	$\underline{\mathit{0.3559}}$
PMO [26]	0.4966	0.4710	0.4452	0.2980	0.5567	0.3459
CMNZ-all	0.4884	0.4517	0.4452	0.2935	0.5589	0.3437
Borda-all	0.4932	0.4588	0.4345	0.2927	0.5600	0.3432
Condorcet-all	0.5048	$\underline{0.4721}$	0.4463	0.2976	0.5494	0.3515
RRF-all	0.5014	0.4632	0.4492	0.2953	0.5561	0.3481
Best-sel	0.5102	0.4662	0.4531	0.3013	0.5561	0.3478
CMNZ-sel	***0.5143****	0.4656	*0.4571**	0.2976	$\underline{\mathit{0.5700}}^{\sharp}$	0.3479
Borda-sel	*0.5102*	0.4670	0.4548	0.2977	*0.5678*	0.3493
Condorcet-sel	$\underline{\boldsymbol{\mathit{0.5197}}}^{*\sharp\dagger}$	0.4671	$\underline{\boldsymbol{\mathit{0.4605}}}^{*}$	$\underline{\mathit{0.3002}}^{\dagger}$	*0.5639*	0.3472
RRF-sel	*0.5136**	0.4575	0.4554	0.2999	*0.5695*$^{\sharp}$	0.3501
TREC Best System	0.4551	0.3350	0.4695	0.3469	0.5544	0.3506

– Aggregating results using all ranking models seems over aggressive which is not advantageous over *BestSingle*, and its outcome is sensitive to the fusion method used. Among six groups of results, the best *x-all* model outperforms *BestSingle* in three groups of them on P@30 without significant difference. Hence, adequate selectivity towards the models rather than aggregating all of them is necessary and expected beneficial to the fusion.

– The fusion approach based on our model selection method demonstrates superior effectiveness in all the six groups of results. Our method consistently demonstrates significant improvement over the base rankers on P@30 (in *italic* and **bold**).

– Compared to *BestSingle*, most P@30 results of *x-sel* models are better, and significant improvement can be found in two groups of results (with superscript '$\sharp$'). It is worth noting that such significance is obtained with the stronger base model $\mathrm{LM_{webprf}}$ on 2011 and 2013 data, which reveals that our re-ranking method is more effective in combination with query expansion that provides higher retrieval accuracy.

– Compared to *PMO*, the *x-sel* models achieve significant improvement in three groups of results (with superscript '*'), two of which are based on query expansion. This implies that our extension on the model selection is effective when combined with different fusion models.

– By using ranker selection, all the four fusion techniques give better results than their counterparts using all the ranking models without model selection. Among them, Condorcet-based technique performs the best by achieving best P@30 values in five groups of experiments. Condorcet vote considers pairwise preference rather than absolute position of the rank like other fusion techniques, which is more precise in general. But the number of pairs can be polynomial with respect to the number of lists and the list length, which may contain lots of noise resulting from less effective models. Our model selection reduces the noise and unleashes the advantage of Condorcet vote.

– Combining our proposed query expansion method and re-ranking strategy, our microblog retrieval pipeline outperforms best systems of TREC2011 and TREC2013 with a large margin and performs comparably to the best system of TREC2012 in terms of P@30.

5. CONCLUSION AND FUTURE WORK

In this paper, we explore the use of model selection and fusion methods for re-ranking microblog search results based on multiple learned ranking models. We extend a query-sensitive model selection method whose outputs from multiple selected rankers are combined with state-of-the-art rank fusion techniques in a query-by-query basis. Experimental results on TREC microblog datasets show that our re-ranking approach based on query-sensitive model selection and fusion performs the best which significantly outperforms the best single ranker and the existing ranker selection method, provided that the base retrieval effectiveness is good enough. Results also show that simply aggregating results of all ranking models is not advantageous over the best single ranking model and its outcome is sensitive to the fusion technique used. In our approach, Condorcet-fuse is especially appealing due to its pairwise nature.

High effectiveness of microblog search results would be critical to many applications on social media. In the future, we plan to deploy our system to other related tasks such as microblog filtering and summarization.

ACKNOWLEDGEMENT

Zhongyu Wei and Kam-Fai Wong are supported by General Research Fund of Hong Kong (417112) and Shenzhen Fundamental Research Program (JCYJ20130401172046450).

6. REFERENCES

[1] J.A. Aslam and M. Montague. Models for Metasearch. In *Proceedings of SIGIR*, pp.276–284, 2001.

[2] L. Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[3] C. Burges, T. Shaked, E. Renshaw, A. Lazier, M. Deeds, N. Hamilton, and G. Hullender. Learning to rank using gradient descent. In *Proceedings of ICML*, pp.89–96, 2005.

[4] G. Cormack, C. Clarke, and S. Buettcher. Reciprocal rank fusion outperforms condorcet and individual rank learning methods. In *Proceedings of SIGIR*, pp.758–759, 2009.

[5] G. Crestani. Combination of similarity measures for effective spoken document retrieval. *Journal of Information Science*, 29(2):87–96, 2003.

[6] S. Deerwester, S. Dumais, G. Furnas, T. Landauer, and R. Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6):391–407, 1990.

[7] M. Efron, P. Organisciak, and K. Fenlon. Improving retrieval of short text through document expansion. In *Proceedings of SIGIR*, pp.911–920, 2012.

[8] A. Din and W. Magdy. Web-based pseudo relevance feedback for microblog retrieval. In *Proceedings of TREC*, 2012.

[9] Y. Duan, L. Jiang, T. Qin, M. Zhou, and H.Y. Shum. An empirical study on learning to rank of tweets. In *Proceedings of COLING*, pp.295–303, 2010.

[10] T. El-Ganainy, Z. Wei, W. Magdy, and W. Gao. QCRI at TREC 2013 Microblog Track. In *Proceedings of TREC*, 2013.

[11] E.A. Fox and J.A. Shaw. Combination of Multiple Searches. In *Proceedings of TREC*, pp.243–252, 1994.

[12] Y. Freund, R. Iyer, R.E. Schapire, Y. Singer. An efficient boosting algorithm for combining preferences. *Journal of Machine Learning Research*, 4:933–969, 2003.

[13] J.H. Friedman. Greedy function approximation: a gradient boosting machine. *The Annals of Statistics*, 29(5):1189–1232, 2001.

[14] W. Gao, Z. Wei, and K.F. Wong. Microblog Search and Filtering with Time Sensitive Feedback and Thresholding based on BM25. In *Proceedings of TREC*, 2012.

[15] Z. Han, X. Li, M. Yang, H. Qi, S. Li, and T. Zhao. HIT at TREC 2012 microblog track. In *Proceedings of TREC*, 2012.

[16] S. Kullback. Information theory and statistics. Dover Publications Inc. (1997).

[17] J.H. Lee. Analyses of multiple evidence combination. In *Proceedings of SIGIR*, pp.267–275, 1997.

[18] J. Lin and M. Efron. Overview of the TREC2013 Microblog Track. In *Proceedings of TREC*, 2013.

[19] T.Y. Liu. Learning to rank for information retrieval. *Foundations and Trends in Information Retrieval*, 3(3):225–331, 2009.

[20] A. Marcus, M. Bernstein, O. Badar, D. Karger, S. Madden, and R. Miller. Twitinfo: aggregating and visualizing microblogs for event exploration. In *Proceedings of SIGCHI*, pp.227–236, 2011.

[21] D. Metzler and W.B. Croft. Linear feature-based models for information retrieval. *Information Retrieval*, 10(3):257–274, 2007.

[22] D. Metzler and C. Cai. USC/ISI at TREC 2011: Microblog track. In *Proceedings of TREC*, 2011.

[23] M. Montague and J.A. Aslam. Condorcet fusion for improved retrieval. In *Proceedings of CIKM*, pp.538–548, 2002.

[24] N. Naveed, T. Gottron, J. Kunegis, A.C. Alhadi. Searching microblogs: coping with sparsity and document quality. In *Proceedings of CIKM*, pp.183–188, 2011.

[25] I. Ounis, C. Macdonald, J. Lin, and I. Soboroff. Overview of the TREC-2011 microblog track. In *Proceedings of TREC*, 2011.

[26] J. Peng, C. Macdonald, and I. Ounis. Learning to select a ranking function. In *Proceedings of ECIR*, pp.114–126, 2010.

[27] J. Ponte and W.B. Croft. A language modeling approach to information retrieval. In *Proceedings of SIGIR*, pp.275–281, 1998.

[28] I. Soboroff, I. Ounis, C. Macdonald, and J. Lin. Overview of the TREC-2012 microblog track. In *Proceedings of TREC*, 2012.

[29] J. Teevan, D. Ramage, and M.R. Morris. # twittersearch: a comparison of microblog search and web search. In *Proceedings of WSDM*, pp.35–44, 2011.

[30] Z. Wei, W. Gao, L. Zhou, B. Li, and K.F. Wong. Exploring tweets normalization and query time sensitivity for twitter search. In *Proceedings of TREC 2011*.

[31] Q. Wu, C.J. Burges, K.M. Svore, and J. Gao. Adapting boosting for information retrieval measures. *Information Retrieval*, 13(3):254–270, 2010.

[32] J. Xu. Solving the word mismatch problem through automatic text analysis. PhD Dissertation, University of Massachusetts, Amherst, 1997.

[33] C. Zhai and J. Lafferty. Model-based feedback in the language modeling approach to information retrieval. In *Proceedings of CIKM*, pp.403–410, 2001.