Late Data Fusion for Microblog Search

Shangsong Liang, Maarten de Rijke, and Manos Tsagkias

ISLA, University of Amsterdam
{s.liang, derijke, e.tsagkias}@uva.nl

Abstract. The character of microblog environments raises challenges for microblog search because relevancy becomes one of the many aspects for ranking documents. We concentrate on merging multiple ranking strategies at post-retrieval time for the TREC Microblog task. We compare several state-of-the-art late data fusion methods, and present a new semi-supervised variant that accounts for microblog characteristics. Our experiments show the utility of late data fusion in microblog search, and that our method helps boost retrieval effectiveness.

1 Introduction

Microblogs, such as Facebook and Twitter status updates, aim at capturing what is happening right now. The short length characteristic of the posts is attractive to people for regularly updating their status [9]. This phenomenon leads to fast paced dynamics reflected in rapidly ever-evolving topics [8]. For search in dynamic environments of this kind, content-based similarity between query and document is only one of many aspects that determines relevance. Other ranking criteria include, e.g., recency, user authority, content, existence of hyperlinks, hashtags, retweets. These ranking options can be offered to the user in isolation, or in combination for a better ranking. Prior research focused on combining these options at retrieval time, and has shown that it is a non-trivial problem [2]. We look at the problem as a late data fusion problem [3], where we have to merge ranked lists produced by a diverse set of rankers into one final ranked list. We investigate the utility of several state-of-the-art late data fusion methods, and present a new semi-supervised variant tailored to microblog search.

We focus on a particular microblog search scenario, that developed by the Microblog track in 2011 Text REtrieval Conference (TREC) [2]. The task uses Twitter data, and is defined as follows: given a query with a timestamp, return relevant and interesting tweets in reverse chronological order. Several dozen groups participated in the task, producing 184 individual runs. We consider these 184 runs as different ranking strategies to be merged together. Conceptually, this fusion problem can be thought of as a federated search problem in uncooperative environments where given a query, a ranked list of documents is returned [5].

Late data fusion has a long history [3] with the `combSUM` family of fusion methods being the oldest and one of the most successful ones in many IR tasks [1, 4, 6, 7]. Broadly speaking, late data fusion methods use two types of features: query-dependent, and ranked list-dependent. Query dependent features include the rank or the retrieval score of a document in the ranked list. Ranked list-dependent features aim at capturing the quality of the ranked list [1, 4]. Below, we extend a weight enabled variant

of combMNZ method to account for an another set of features, that are document-dependent, and encode characteristics specific to microblogs.

The research question we aim at answering is whether late data fusion is useful for microblog search, and whether taking into account individual document-specific features and their combination helps performance compared to methods that assign weights only to ranked lists. Our main contribution is a semi-supervised method that generalizes a weight-enabled variant of the combSUM family to take into account document-dependent features.

In Section 2 we describe our method, in Section 3 and 4 we present our experiments and report on our results and analysis, and in Section 5 we conclude.

2 Method

Our method for merging ranked lists of microblog posts works as follows. Given a set of ranked lists $\mathcal{R}$ generated by a set of systems $\mathcal{S}$, let d be a document in a ranked list $r \in \mathcal{R}$ generated from system $s \in \mathcal{S}$ in response to a query q. Our method, $WcombMB$, scores a d as:

$$WcombMB(q,d) = |\{r : d \in r\}| \cdot \sum_r w(r) f(\mathbf{x}^r, d), \tag{1}$$

where $w(r)$ is the weight for ranked list r, and $f(\mathbf{x}^r, d)$ is a linear combination of query- and document-dependent features $\mathbf{x}$:

$$f(\mathbf{x}^r, d) = \sum_{\chi \in \mathbf{x}} \omega(\chi) \cdot score(\chi, d), \tag{2}$$

where our feature set $\mathbf{x} := \{$*hashtag*, *link*, *retweets*, *query*$\}$ will be explained below, $\omega(\chi)$ is the weight of feature $\chi \in \mathbf{x}$, and $score(\chi, d)$ is the linearly normalized score of d for feature χ.

Next, we describe our features and explain how we assign weights and scores to documents. We start with the weight $w(r)$ in (1), a *ranked list-dependent* feature. We follow [1], and use a semi-supervised approach. We evaluate each ranked list against our ground truth, and use its performance measured using P@30 as the weight of the ranked list $w(r)$.

Next, we turn to (2), where there is a single *query-dependent* feature ("*query*"), which takes into account the retrieval score of the document. In the setting of the TREC 2011 Microblog track, documents are ranked in reverse chronological order regardless of their retrieval score. We use the inverse of the rank of the document over the number of returned documents as $score(q, d)$ instead of the retrieval score which is usually used in combSUM and its variants. For *document-dependent* features (*hashtag*, *link*, *retweets*), $score(\chi, d) = 1$ if d has χ, i.e., if it contains at least one hashtag (H) or link (L), or if it has at least one retweet (RT), otherwise it is 0. We optimize the weights $\omega(\chi)$ using grid search; We set the constraint $\sum_\chi \omega(\chi) = 1$, and vary $\omega(\chi)$ from 0.1 to 0.9 with 0.1 step each time.

Once all documents in $\mathcal{R}$ are assigned a score, we rank them by their score in descending order, we keep only the top-30, and re-rank them in reverse chronological order.

3 Experiments

Our experiment aims at answering what is the relative improvement in performance when using late data fusion methods compared to the performance of the best ranked list in $\mathcal{R}$. We use eight late data fusion methods; Two unsupervised data fusion methods, i.e., `combSUM`, and `combMNZ`; Two semi-supervised variants that accept weights for each ranked list, i.e., `WcombSUM`, `WcombMNZ`; And our method `WcombMB` using one document-dependent feature (`-H`, `-L`, `-RT`), and their combination (`-ALL`).

We also aim at capturing the effect on the performance of the number and the quality of the ranked lists we consider. We randomly sample $\{5, 10, 20, 40\}$ ranked lists out of 184, and record the P@30 of the best ranked list in the sample. We merge these sampled ranked lists using the methods above, and record the relative difference in P@30 over the best individual ranked list in the sample. The relative differences are recorded after optimizing the features weights $\omega(\chi)$ for each method. We repeat this procedure 10 times, and report on the average relative differences in P@30.

For evaluation we use the TREC Microblog 2011 task (TMB2011) [2]; we use the collection in JSON format. Out of the 49 queries in the ground truth, 19 are kept for training our semi-supervised methods (i.e., $w(r)$ in (1)), and 30 are used for testing. Our pool of ranked lists consists of the 184 systems submitted to TMB2011. We optimize for, and report on the official TMB2011 measure, P@30.

4 Results

We illustrate our results in Fig. 1. Our method, `WcombMB`, that uses document-dependent features shows higher average relative improvement on P@30 over `combSUM`, `combMNZ`, and their weigh- ted variants. *Links* shows to be the most important document-dependent feature (`WcombMB-L`) marking performance close to when using all document-dependent features (`WcombMB-ALL`). This is probably due to the way the ground truth was assembled; interesting tweets are deemed those that contain a hyperlink.

We find that the number of merged ranked lists plays an important role in performance. We achieve higher improvements when considering 5, and 10 ranked lists. For larger numbers, the gains in improvement become lower. An interesting pattern is that of `combSUM` which marks its best performance for 10 ranked lists coming close to the best performance from `WcombMB-ALL`. Among all methods, `WcombMB-ALL` shows to be the most robust to changes in the number of ranked lists.

5 Conclusions

We have looked at late data fusion for microblog search. We explored the potential of traditional data fusion methods, their weighted variants, and extended a weighted method to incorporate document-dependent features. We found that considering the weight of ranked lists, the document-dependent features and their combination in a specific way can boost the performance of microblog search. In future work we envisage more elaborate methods for scoring document-dependent features, and weighting our sets of features using machine learning methods. For better understanding the effect of the quality of individual ranked lists, we plan to bias sampling when selecting lists.

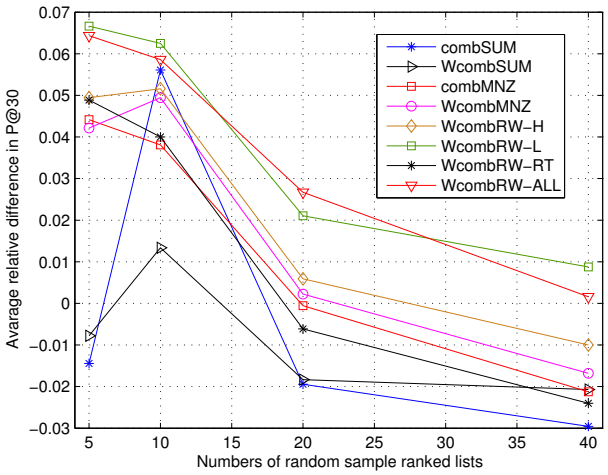

Fig. 1. Average relative difference in P@30 for eight late data fusion methods.

Acknowledgments. This research was partially supported by the European Union's ICT Policy Support Programme as part of the Competitiveness and Innovation Framework Programme, CIP ICT-PSP under grant agreement nr 250430, the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreements nr 258191 (PROMISE Network of Excellence) and 288024 (LiMoSINe project), the Netherlands Organisation for Scientific Research (NWO) under project nrs 612.061.814, 612.061.815, 640.004.802, 727.011.005, 612.001.116, HOR-11-10, the Center for Creation, Content and Technology (CCCT), the BILAND project funded by the CLARIN-nl program, the Dutch national program COMMIT, the ESF Research Network Program ELIAS, and the Elite Network Shifts project funded by the Royal Dutch Academy of Sciences.

6 References

[1] D. He and D. Wu. Toward a robust data fusion for document retrieval. In *IEEE NLP-KE'08*, 2008.

[2] J. Lin, C. Macdonald, I. Ounis, and I. Soboroff. Overview of the TREC 2011 Microblog track. In *TREC 2011*. NIST, 2012.

[3] J. A. Shaw and E. A. Fox. Combination of multiple searches. In *TREC 1992*, pages 243–252. NIST, 1993.

[4] D. Sheldon, M. Shokouhi, M. Szummer, and N. Craswell. Lambdamerge: merging the results of query reformulations. In *WSDM '11*, pages 795–804. ACM, 2011.

[5] L. Si and J. Callan. Modeling search engine effectiveness for federated search. In *SIGIR '05*, pages 83–90. ACM, 2005.

[6] C. G. M. Snoek, M. Worring, and A. W. M. Smeulders. Early versus late fusion in semantic video analysis. In *MULTIMEDIA '05*, pages 399–402. ACM, 2005.

[7] M. Tsagkias, M. de Rijke, and W. Weerkamp. Linking online news and social media. In *WSDM '11*, pages 565–574. ACM, 2011.

[8] J. Yang and J. Leskovec. Patterns of temporal variation in online media. In *WSDM '11*, pages 177–186. ACM, 2011.

[9] D. Zhao and M. B. Rosson. How and why people twitter: the role that micro-blogging plays in informal communication at work. In *GROUP '09*, pages 243–252. ACM, 2009.