Ranking Tweets with Local and Global Consistency Using Rich Features

Zhankun Huang[1,2], Shenghua Liu[1,⋆], Pan Du[1], and Xueqi Cheng[1]

[1] Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China
[2] University of Chinese Academy of Sciences, Beijing, China
{huangzhankun}@software.ict.ac.cn, {liushenghua,dupan,cxq}@ict.ac.cn

Abstract. Ranking tweets is more challenging in Microblog search because of content sparseness and lack of context. Traditional ranking methods essentially using Euclidean distance are limited to local structure. Manifold structure helps to rank with local and global consistency. However such structure is empirically built on content similarity in an unsupervised way, suffering from sparseness while being adopted in tweet ranking. In this paper, we explore rich features to alleviate content sparseness problem, where time locality feature is proposed to consider context dependency. We then propose a supervised learning model that aggregates rich features to construct manifold structure. A learning algorithm is then designed for solving the model by minimizing the loss of labeled queries. At last we conduct a series of experiments to demonstrate the performance on 109 labeled queries from TREC Microblogging. Compared with the well-known baselines and empirical manifold structure, our algorithm achieves consistent improvements on the metrics.

Keywords: time locality, rich features, manifold, supervised learning.

1 Introduction

Statistics from Wikipedia [17] indicate that, there are approximately 340 million tweets posted per day. Besides to reading tweets from their timelines or trending topics, people also need to search tweets on a topic. Different from traditional webpage search, tweet ranking in Microblog search is more challenging. Because of the limitation on length, tweets do not provide sufficient word occurrences. Tweets are context-dependent while lacking context makes context similarity cannot be measured directly. Nevertheless, tweets on a specific topic are usually temporally concentrative, especially when some big event breaks out, like terrorist attack. Hence tweets posted during a short time period can be regarded as having the same context. On the other hand, tweets posted in totally different time periods possibly refer to different topics though they contain some common words. As the examples from TREC Microblogging 2011 show in Table 1, both of them talked about "protest". However since they were posted in quite different

⋆ Corresponding author.

V.S. Tseng et al. (Eds.): PAKDD 2014, Part I, LNAI 8443, pp. 298–309, 2014.

Table 1. Context-dependent tweets examples: both tweets were talking about "protest". But because of different contexts, they were related to different topics.

Relevant Tweet		Query
Post Time	**Content**	
Jan. 29 2011 18:14:21	Al Jazeera: **Protest**ors create a human shield around the Egyptian Museum to protect it.	Egyptian **protest**ers attack museum
Feb. 01 2011 12:46:40	Royal Palace: Jordan's king sacks government in wake of **protest**s.	**protest**s in Jordan

contexts measured by post time, they were about different topics. The former one was related to "Egyptian protesters attack museum" and the latter one was about "protests in Jordan". Thus we use the time locality to approximate the context dependency in this paper.

Traditional ranking algorithms based on Euclidean distances are limited to the local structure [20]. Manifold structure has been employed for ranking problem [3,5,16,20], which maintains local and global consistency simultaneously. Generally [19,20], manifold structure is built on content similarity empirically. Unlike traditional webpages, tweets are extremely short and lack context. Content similarity alone cannot well describe the relevance between tweets. The manifold graph of all related tweets for a query may be disconnected due to sparseness, and mixed with much noise because of different contexts. Therefore, rich features, especially the time locality feature are explored to weight the network of manifold structure.

In this paper, we aggregate rich features to construct the manifold structure in a supervised way, with time locality as an approximation of context dependency. The similarities between query and tweets or tweets themselves are measured from different views in terms of each feature, so that several heterogeneous structures are obtained, which are then linearly combined to model the objective function with local and global consistency. A supervised algorithm is then designed to learn the combination coefficients. As seen later, the manifold structure leant from rich features better benefits the ranking than being determined empirically on a single content similarity. A series of experiments are conducted on the dataset of tweet corpus and labeled queries officially provided by TREC Microblogging. The evaluation results show that, in terms of P@10, P@20, P@30, and AUC (area under ROC curve), our approach can outperform the ranking models of Lucene and RankSVM.

The rest of the paper is organized as follows: Section 2 discusses the related work. Section 3 describes our approach in detail. Section 4 introduces the features we use for ranking. Section 5 demonstrates our evaluation results and we make conclusion in Section 6.

2 Related Works

Due to the features of short, sparseness and frequency, Microblog search is very different from traditional web search. As Teevan [14] explored, Twitter queries were shorter, but contained longer words, more specialized syntax, and more references to people. In addition, unlike the URLs in social bookmarking sites which have provided high values for search engine, the shorten URLs in tweets can either be high in quality or spam [9]. We will not expand tweets by extracting the contents that the URLs link to, and treat whether a tweet contains URL as a feature.

User oriented search has attracted more and more attentions. Instead of presenting a simple list of tweet messages, TweetMotif [12] provided a facet search interface to allow user to navigate tweets by topics, by which searched tweets are grouped into subtopics extracted by language model. Another personalized tweet ranking method [15] was proposed by ranking the incoming tweets based on the likelihood user may retweet them, and ranking the users given a tweet based on their willingness to retweet it. Using four groups of features: Author-based, Tweet-based, Content-based, User-based, a Coordinate Ascent learning to rank algorithm was trained to rank tweets according to users' need. To address the problem of sparseness Naveed [11] ignored length normalization for tweets. In his point of view, document length normalization for short documents like tweets may introduce an unmotivated bias. Abel [1] enriched the semantics of tweets by extracting facet values from tweets, where external resources were used in order to create facet and facet-value pairs. Massoudi [10] used time-dependent query expansion to overcome the disadvantage of redundancy-based IR methods over short documents retrieval. Duan [6] adopted learning to rank algorithm to select a feature set from both content and non-content features of tweets and proposed a ranking model based on RankSVM algorithm. In our approach, we treat the length of tweet as a feature, and use rich features to alleviate the problem of sparseness. Specifically, the weight of each feature is not given empirically, instead they are determined through a supervised way.

Jabeur [8] measured the conditional probability of relevance using Bayesian network model. In his study, time magnitude of tweet was estimated by query terms occurrence in the temporal neighborhood. His experimental results showed that time magnitude was a primordial feature for ranking tweets. In this paper, we introduce into time locality feature to simulate the context of tweet. For the tweets posted in a short time period, we assume that they are in the same context. Zhang [18] proposed a transductive framework that generated training data while no labeled data was available, and boosted the ranking by adding confident unlabeled data during iteratively training. In our approach, after the manifold structured is learnt, all the tweets to be ranked are treated as no labeled data and the query is treated as labeled data.

In some recent works, manifold structure has been employed for ranking problems and clustering problems[13] in consideration of local and global consistency. Zhou [20] proposed a universal ranking algorithm with respect to the intrinsic manifold structure collectively revealed by a great amount of data. Cheng and Du [3,5] introduced sink point into manifold structure to address diversity as well as relevance and importance. Wan and Yang [16] proposed a novel extractive approach based on manifold-ranking of sentences. Generally, manifold structure is built on content similarity empirically, which suffers from sparseness in tweet ranking. In our research, we leverage rich features to construct tweet manifold, including content, time locality and several intrinsic features, which are combined in a supervised way[4].

3 Our Approach

Ranking on data manifolds was proposed by Zhou [20]. Let f denote a score function, which assigns data x_i with ranking score f_i. And y_i is the initial ranking score for data x_i. For the labeled data $y_i = 1$ and for the unlabeled data $y_i = 0$. Let W denote an affinity matrix where W_{ij} is the similarity between data x_i and x_j. We name W as the similarity matrix. Note that W_{ii} is set to 0 to avoid self-reinforcement. Symmetrically normalize W by $S = D^{-1/2}WD^{-1/2}$ where D is a diagonal matrix with D_{ii} equal to the sum of the i-th row of W. As [20] illustrates, the final score function f^* can be achieved directly by:

$$f^* = (1-\alpha)(I - \alpha S)^{-1} y \; . \tag{1}$$

or in an iterative way using the following iterative function:

$$f_{t+1} = \alpha S f_t + (1-\alpha) y \; . \tag{2}$$

where α is a parameter for trade-off.

For tweet ranking, we then construct a weighted network over the manifold structure of tweets, where the nodes in the network are the unlabeled tweets and labeled query, and edges reflect the similarities between query and tweets or tweets themselves. Then we perform a graph-based semi-supervised method to rank tweets by their distances to the query on the intrinsic manifold structure. In the original thought of manifold regularization framework, the similarity matrix W is empirically built on content similarity. In our approach, we do not determine W in advance. Instead we achieve it through a supervised learning process leveraging rich features.

For each feature ϕ, we can define some similarity measurements. Then the similarity matrix W for our approach is defined as follow:

$$W = a^T \mathcal{F} = \sum_{i=1} a^i F^i \; . \tag{3}$$

where each F^i is a similarity matrix achieved by a kind of similarity measurement over some feature, and a is the linear coefficient vector needed to be learnt.

Given a corpus $CO = \{(q^i, T^i) | i = 1, \dots, N\}$, where q^i denotes the i-th query and T^i denotes the set of labeled tweets for q^i, $P^i = \{p_1, \dots, p_j\}$ and $O^i = \{o_1, \dots, o_j\}$ denotes the set of relevant and irrelevant tweets in T^i. To find out the optimal solution for coefficient a, we construct the following optimization problem:

$$min_a L(a) = \frac{1}{2}\lambda \|a\|^2 + \sum_{i=1}^{N} \sum_{\substack{o \in O^i \\ p \in P^i}} l(f_o^i - f_p^i) \; . \tag{4}$$

where f^i is the final score function for q^i obtained from Eqn. (1) directly or Eqn. (2) iteratively, and λ is the regularization parameter that trades off between the model complexity and the fit of the model. $l(x)$ is the loss function which assigns a non-negative penalty to the violation of constraint $f_p^i > f_o^i$. Here we use Wilcoxon-Mann-Whitney (WMW) loss $l(x) = (1 + e^{-\frac{x}{b}})^{-1}$ since it shows good performance in our approach.

To minimize (4) with respect to parameter a, we use gradient descent method to minimize the loss and find the optimal solution for a. The gradient of $L(a)$ with respect to a is as follow:

$$\frac{\partial L(a)}{\partial a} = \lambda a + \sum_{i=1}^{N} \sum_{\substack{o \in O^i \\ p \in P^i}} \frac{\partial l(\delta_{op})}{\partial \delta_{op}} \left(\frac{\partial f_o^i}{\partial a} - \frac{\partial f_p^i}{\partial a} \right) . \tag{5}$$

where $\delta_{op} = f_o^i - f_p^i$. Next we need to know $\frac{\partial f^i}{\partial a}$, $i.e. \frac{\partial f^*}{\partial a}$.

When f_t in Eqn. (2) converges to f^*, we have:

$$\begin{aligned} f^* &= \alpha S f^* + (1 - \alpha) y \; . \\ \frac{\partial f^*}{\partial a} &= \alpha \left(\frac{\partial S}{\partial a} f^* + S \frac{\partial f^*}{\partial a} \right) . \\ \frac{\partial f^*}{\partial a} &= \alpha (I - \alpha S)^{-1} \left(\frac{\partial S}{\partial a} f^* \right) . \end{aligned}$$

where $\frac{\partial S}{\partial a}$ is a supervector, elements in which are matrices, and can be expressed as: $\frac{\partial S}{\partial a} = \left[\frac{\partial S}{\partial a^1}, \frac{\partial S}{\partial a^2}, \ldots, \frac{\partial S}{\partial a^k} \right]$. Here we denote D^i as a diagonal matrix with (j, j)-element equal to the sum of the j-th row of the i-th feature matrix F^i for a specific query, then we have:

$$\begin{aligned} \frac{\partial S}{\partial a^i} &= \frac{\partial D^{-\frac{1}{2}}}{\partial a^i} W D^{-\frac{1}{2}} + D^{-\frac{1}{2}} \frac{\partial W}{\partial a^i} D^{-\frac{1}{2}} + D^{-\frac{1}{2}} W \frac{\partial D^{-\frac{1}{2}}}{\partial a^i} \\ &= -\frac{1}{2} D^{-\frac{3}{2}} D^i W D^{-\frac{1}{2}} + D^{-\frac{1}{2}} F^i D^{-\frac{1}{2}} - \frac{1}{2} D^{-\frac{1}{2}} W D^{-\frac{3}{2}} D^i \; . \end{aligned}$$

For the corpus we are going to introduce in the following, tweets are categorized into highly relevant, minimally relevant and irrelevant. To adapt to this multi-label situation, Eqn. (4) and Eqn. (5) are needed to be modified as follow:

$$min_a L(a) = \frac{1}{2} \lambda \|a\|^2 + \sum_{i=1}^{N} \sum_{\substack{o \in O^i \\ m \in M^i}} l(f_o^i - f_m^i) + \sum_{i=1}^{N} \sum_{\substack{o \in O^i \\ p \in P^i}} l(f_o^i - f_p^i) + \sum_{i=1}^{N} \sum_{\substack{m \in M^i \\ p \in P^i}} l(f_m^i - f_p^i) \; . \tag{6}$$

$$\frac{\partial L(a)}{\partial a} = \lambda a + \sum_{i=1}^{N} \sum_{\substack{o \in O^i \\ m \in M^i}} \frac{\partial l(\delta_{om})}{\partial \delta_{om}} \left(\frac{\partial f_o^i}{\partial a} - \frac{\partial f_m^i}{\partial a} \right) + \sum_{i=1}^{N} \sum_{\substack{o \in O^i \\ p \in P^i}} \frac{\partial l(\delta_{op})}{\partial \delta_{op}} \left(\frac{\partial f_o^i}{\partial a} - \frac{\partial f_p^i}{\partial a} \right)$$
$$+ \sum_{i=1}^{N} \sum_{\substack{m \in M^i \\ p \in P^i}} \frac{\partial l(\delta_{mp})}{\partial \delta_{mp}} \left(\frac{\partial f_m^i}{\partial a} - \frac{\partial f_p^i}{\partial a} \right). \tag{7}$$

where O^i, M^i and P^i denote the set of irrelevant, minimally relevant and highly relevant tweets for q^i respectively.

Finally the optimization problem in (6) can be solved with a standard quasi-Newton method. Here Broyden-Fletcher-Goldfarb-Shanno (BFGS) [2] is employed for the learning process. Note that the optimization problem in Eqn. (6) is generally non-convex, we try several different start points to find a good solution which though may not be a global optimal solution.

Table 2 shows the overview of the learning algorithm for manifold structure, in which derivative of $L(a)$ is denoted as $dL(a)$.

Table 2. Learning algorithm of manifold structure for ranking tweets

Algorithm. Learning of Manifold Structure
1: Given an initial point a=a_0 and an initial Hessian matrix $B = B_0 = I$.
2:**loop**
3: evaluate $dL(a)$ using Eqn.(7).
4: if $norm(dL(a)) < \varepsilon$:
5: break.
6: else:
7: obtain a descent direction p by solving $Bp = -dL(a)$.
8: perform a linear search to find an acceptable step size s.
9: update a using $a = a + sp$.
10: update Hessian matrix B as BFGS algorithm says.
11:**end loop**
12:**return** fina a.

4 Features Description

To represent the similarity matrix W, we explore rich features to measure the similarities between query and tweets or tweets themselves from different views in terms of all these features, so that rich features can be exploited. The available features can be derived from, for example, content, context, length etc.

Number of co-occurrence word, cosine similarity, Dice coefficient and Jaccard similarity are employed to measure the content similarity from different views. Note that by weighting the words in tweets and query using different approaches, we compose different types of features and similarities, such as cosine similarity using tf weight and cosine similarity using tf-idf weight. All these similarities will be combined together to better capture the content similarity.

As addressed in Section 1, for ranking tweets, time locality is an important feature that can help us address the criterion of context awareness. Thus the context dependency can be approximated as the following equation (8) in terms of time locality.

$$1 - \frac{|pt_i - pt_j|}{max\{|pt_i - pt_j||\forall i > j\}} \tag{8}$$

where pt_i is the post time of tweet t_i. For the query, the post time is represented as querying time.

Several intrinsic features are extracted as well. Here we only introduce some of them. "*oov_ratio*" is the out-of-vocabulary word count over the total word count, "*unique_word_ratio*" is similar to "*oov_ratio*", "*begin_with_at*" means whether the content of tweet starts with "@", "*has_rt*" means whether the tweet is posted by retweeting others. On the other hand, we combine these intrinsic features into vector as a new feature, which is denoted as "*intrinsic_vector*", and use cosine similarity on it. Because of lack of intrinsic features for the query, the similarities between query and tweets in terms of intrinsic features are set to 0.

The rich features for similarity measurement derived from content, time locality and intrinsic features are listed in Table 3. Note that we may not know in advance which feature is better than others, or whether a feature is useless and can be removed. Through a supervised learning approach, we learn the combination of these features and assign useful features higher weights.

Table 3. Rich features for similarity measurement derived from content, time locality and intrinsic features

Content Features	Time Locality Feature	Intrinsic Features
co_occurrence_bool co_occurrence_tf co_occurrence_idf co_occurrence_tfidf cos_sim_bool cos_sim_tf cos_sim_tfidf dice_bool dice_idf jaccard_bool jaccard_idf	time_locality	oov_ratio doc_len unique_word_ratio entropy has_url has_hashtag begin_with_at has_rt intrinsic_vector

5 Experiments

Our experiments are conducted based on the tweet corpus and labeled queries officially provided by TREC Microblogging. We compare our approach with the well-known baseline methods including different ranking models in Lucene, RankSVM and conventional manifold approach. We then evaluate the effectiveness of the features we selected for our approach, such as time locality and intrinsic features.

5.1 Corpus

From the TREC Microblogging, we have got about 113,928 labeled tweets on 109 topic queries, out of 7,443,387 tweets totally. The first 49 topics are treated as training set and the remained 60 query topics are for testing. We preprocessed the labeled tweets simply by removing the "@*user*", "#*tag*#", URL, stop words, punctuations and transferring all the letters into lower case. The tweets are officially labeled as highly relevant, relevant, and irrelevant with integers 2, 1, and 0 respectively.

5.2 Baseline Methods

Apache Lucene is a well-known information retrieval software library, which has been widely recognized for its utility in the implementation of Internet search engines. We compare our approach with Lucene 4.2 using default, BM25 and LMDirichlet models, which are denoted as **Lu-Default**, **Lu-BM25**, and **Lu-LMD** respectively for comparisons. It is worth noticing that real-time constraint is added to make sure that only the tweets posted earlier than querying time are returned. The parameters of BM25 and LMDirichlet have been well tuned.

We use RankSVM as another baseline method. Support vector machines (SVMs) has been proved to be effective for classification and regression analysis. RankSVM is an instance of structured SVM used for ranking problem. For RankSVM, we select almost the same features employed in our approach to represent each tweet including content, time locality and intrinsic features.

Lastly, we also implement conventional manifold approach as baseline, by constructing manifold structure on single content similarity. And the edges with similarity less than a threshold $\varepsilon \in (0, 1)$ are dropt. We implement **Manifold-Cosine** using cosine similarity with tfidf weight "*cos_sim_tfidf*" in Table 3.

5.3 Evaluation Metrics

We use the Microblog track task metrics for evaluation. The main metrics for the task are the receiver operating characteristic (ROC) curve [7] and P@K. The ROC curve shows precision versus fallout for every possible score threshold while P@K gives a simple measure of ranking effectiveness.

To evaluate the result that the ROC curve explains, area under the curve (AUC) metric is employed. The formulation to calculate the AUC value for a ROC curve is as follow:

$$AUC = \frac{\sum_{i \in positive\ class} rank_i - \frac{M \times (M+1)}{2}}{M \times N}$$

where M is the number of positive samples and N is the number of negative samples.

As people usually prefer to browse the results in the top returned by search engines, we only consider the top K tweets in ranking list. The value P@K is calculated as follows:

$$P@K = \frac{|\{relevant\, tweets\, in\, top\, K\, results\}|}{K}$$

5.4 Evaluation Results

In the experiments, we list P@10, P@20 and P@30 for precision metrics and denote our approach as **Manifold-Rich**.

Table 4. Comparison with baselines

Methods	P@10	incr(%)	P@20	incr(%)	P@30	incr(%)	AUC	incr(%)
Lu-Default	0.1966	44.8	0.1695	49.0	0.1475	52.8	0.7970	11.6
Lu-BM25	0.2271	25.4	0.1932	30.7	0.1678	34.3	0.7985	11.4
Lu-LMD	0.1576	80.6	0.1254	101.4	0.1192	89.1	0.7838	13.4
RankSVM	0.2627	8.4	0.2102	20.1	0.1836	22.8	0.8909	-0.2
Manifold-Cosine	0.0695	309.6	0.0907	178.4	0.0819	175.2	0.8711	2.1
Manifold-Rich	0.2847	–	0.2525	–	0.2254	–	0.8892	–

The precision comparisons between Manifold-Rich and the baseline methods are listed in Table 4. The increasing percentages, "incr(%)", of Manifold-Rich are listed right next to the corresponding metric columns. It is seen that Manifold-Rich outperforms Lucene significantly no matter it uses default, LMDirichlet or BM25 models. Compared with Lu-LMD, Manifold-Rich achieves significant improvements of 80.6%, 101.4% and 89.1% respectively on P@10, P@20 and P@30. And compared with Lu-Default, Manifold-Rich gets improvements of 44.8%, 49.0% and 52.8% respectively. Lastly compared with Lu-BM25 which gets the best precision among the three Lucene models, the improvements of our approach on P@10, P@20 and P@30 are 25.4%, 30.7% and 34.3% respectively. It is seen that only using content similarity between query and tweet in Lucene is not sufficient for ranking tweets due to the content sparseness. By leveraging the local and global structure on rich features, we can achieve much better ranking results. Compared with RankSVM, Manifold-Rich achieves improvements of 8.4%, 20.1% and 22.8% with respect to P@10, P@20 and P@30. The manifold baseline Manifold-Cosine is constructed by dropping those edges with weight less than an empirical threshold 0.05. It is seen that Manifold-Rich outperforms Manifold-Cosine with improvements of 309.6%, 178.4%, 175.2% with respect to P@10, P@20, P@30, which indicates that the manifold structure leant from rich features better benefits the tweet ranking than being empirically determined on a single content similarity. It worth noticing that with the statistical significance testing of t-test, the p-value is less than 0.01, which indicates that our approach outperforms others significantly. It is seen that in terms of AUC value, our approach receives much better results than the ranking models in Luene. Especially compared with Lu-BM25, the AUC improvement of Manifold-Rich is 11.4%. The AUC metrics of manifold approaches are close to each other. Though the ROC curve of RankSVM gains a little more AUC value than that of Manifold-Rich, considering the impressive improvements on precision, our approach is still reliable.

5.5 Effectiveness of Rich Features

To show the effectiveness of the rich features, we conduct a set of experiments with some selections of features in our approach, such as time locality feature and intrinsic features. Traditional ranking approaches mainly rely on the content similarity between document and query for ranking. Thus we implement **Manifold-Content** to evaluate the performance of our approach only with content feature without an empirical threshold as Manifold-Cosine does. We use all the content features in Table 3 to measure the content similarity. To evaluate the effectiveness of time locality feature for context awareness, we then use both content features and time locality feature in Table 3 to construct the manifold structure, and compare it with Manifold-Content. This version is denoted as **Manifold-Content-Context**.

The comparisons between different feature selections of our approach are demonstrated in Table 5. The cells of "incr(%)" give the increasing percentages of Manifold-Rich comparing to others. Compared with Manifold-Content, Manifold-Rich achieves improvements of 33.3%, 41.9% and 40.0% with respect to P@10, P@20 and P@30. It shows that instead of only using content similarity, with rich features to construct the manifold structure, the ranking results are much better. Comparing Manifold-Content-Context with Manifold-Content, we find that with time locality feature taken into account for context awareness, Manifold-Content-Context achieves promising improvements of 22.2%, 24.7% and 18.6% on P@10, P@20 and P@30 respectively, which indicates that the time locality feature has somehow captured the missing context of tweets and is valuable for constructing the manifold. Therefore, by considering both content and time locality features, we can address the criteria of content relevance and context awareness simultaneously for ranking tweets.

Table 5. Comparison between our approaches with different features

Methods	P@10	incr(%)	P@20	incr(%)	P@30	incr(%)	AUC	incr(%)
Manifold-Content	0.2136	33.3	0.1780	41.9	0.1610	40.0	0.8811	0.9
Manifold-Content-Context	0.2610	9.1	0.2220	13.7	0.1910	18.0	0.8876	0.2
Manifold-Rich	0.2847	–	0.2525	–	0.2254	–	0.8892	–

6 Conclusions

In this paper, we employ rich features to construct manifold structure on tweets in a supervised way and then rank tweets on it. The similarities between query and tweets or tweets themselves are estimated from different views in terms of each feature, resulting in heterogeneous structures, which are then combined to model the objective function. Especially, by combining content similarity and context similarity together, we can address the criteria of content relevance and context awareness simultaneously. A learning algorithm is designed to solve the model to minimize the loss of the labeled ranked tweets. Experimental results demonstrate that our approach outperforms the ranking models in Lucene significantly and achieves higher precision than RankSVM trained on the common relevance features. In addition, we experimentally show that manifold structure learnt from rich features better benefits the ranking than being determined empirically only on content similarity.

Acknowledgments. Thanks a lot for the supply of corpuses from TREC Microblogging. This paper is partially supported by National Grand Fundamental Research 973 Program of China (No. 2012CB316303, 2013CB329602), National Natural Science Foundation of China (No. 61202213, 61202215, 61173064, 61173008, 61232010), Projects of Development Plan of the State High Technology Research (No. 2012AA011003), Beijing nova program (No. Z121101002512063).

References

1. Abel, F., Celik, I., Houben, G.-J., Siehndel, P.: Leveraging the semantics of tweets for adaptive faceted search on twitter. In: Aroyo, L., Welty, C., Alani, H., Taylor, J., Bernstein, A., Kagal, L., Noy, N., Blomqvist, E. (eds.) ISWC 2011, Part I. LNCS, vol. 7031, pp. 1–17. Springer, Heidelberg (2011), http://dx.doi.org/10.1007/978-3-642-25073-6_1
2. Broyden, C.G.: A Class of Methods for Solving Nonlinear Simultaneous Equations. Mathematics of Computation 19, 577 (1965)
3. Cheng, X.Q., Du, P., Guo, J., Zhu, X., Chen, Y.: Ranking on data manifold with sink points. IEEE Transactions on Knowledge and Data Engineering 25(1), 177–191 (2013)
4. Du, P., Guo, J., Cheng, X.: Supervised Lazy Random Walk for Topic-Focused Multi-document Summarization. In: IEEE International Conference on Data Mining, pp. 1026–1031 (2011)
5. Du, P., Guo, J., Zhang, J., Cheng, X.: Manifold ranking with sink points for update summarization. In: International Conference on Information and Knowledge Management, pp. 1757–1760 (2010)
6. Duan, Y., Jiang, L., Qin, T., Zhou, M., Shum, H.Y.: An Empirical Study on Learning to Rank of Tweets. In: International Conference on Computational Linguistics, pp. 295–303 (2010)
7. Fawcett, T.: An introduction to roc analysis. Pattern Recognition Letters 27(8), 861–874 (2006)

8. Jabeur, L.B., Tamine, L., Boughanem, M.: Uprising microblogs: a bayesian network retrieval model for tweet search, pp. 943–948 (2012)
9. Kandylas, V., Dasdan, A.: The utility of tweeted urls for web search. In: Proceedings of the 19th International Conference on World Wide Web, pp. 1127–1128. ACM (2010)
10. Massoudi, K., Tsagkias, M., de Rijke, M., Weerkamp, W.: Incorporating Query Expansion and Quality Indicators in Searching Microblog Posts (2011)
11. Naveed, N., Gottron, T., Kunegis, J., Alhadi, A.C.: Searching microblogs: coping with sparsity and document quality. In: Proceedings of the 20th ACM International Conference on Information and Knowledge Management, pp. 183–188. ACM (2011)
12. O'Connor, B., Krieger, M., Ahn, D.: TweetMotif: Exploratory Search and Topic Summarization for Twitter. In: International Conference on Weblogs and Social Media (2010)
13. Souvenir, R., Pless, R.: Manifold Clustering. In: International Conference on Computer Vision, vol. 1, pp. 648–653 (2005)
14. Teevan, J., Ramage, D., Morris, M.R.: TwitterSearch: a comparison of microblog search and web search. In: Web Search and Data Mining, pp. 35–44 (2011)
15. Uysal, I., Croft, W.B.: User oriented tweet ranking: a filtering approach to microblogs, pp. 2261–2264 (2011)
16. Wan, X., Yang, J., Xiao, J.: Manifold-Ranking Based Topic-Focused Multi-Document Summarization. In: International Joint Conference on Artificial Intelligence, pp. 2903–2908 (2007)
17. Wikipedia: Twitter. `en.wikipedia.org/wiki/Twitter`
18. Zhang, X., He, B., Luo, T.: Transductive learning for real-time twitter search. In: Sixth International AAAI Conference on Weblogs and Social Media (2012)
19. Zhou, D., Bousquet, O., Lal, T.N., Weston, J., Schölkopf, B.: Learning with local and global consistency. In: Advances in Neural Information Processing Systems, vol. 16(753760), p. 284 (2004)
20. Zhou, D., Weston, J., Gretton, A., Bousquet, O., Sch, B.: Ranking on Data Manifolds. In: Neural Information Processing Systems (2004)