Proceedings of the Sixth International AAAI Conference on Weblogs and Social Media

Transductive Learning for Real-Time Twitter Search

Xin Zhang and **Ben He** and **Tiejian Luo**
zhangxin510@mails.gucas.ac.cn, {benhe,tjluo}@gucas.ac.cn
Information Dynamic and Engineering Applications Laboratory
Key Laboratory of Computational Geodynamics
Graduate University of Chinese Academy of Sciences

Abstract

Recency is an important dimension of relevance for real-time Twitter search as users tend to be interested in fresh news and events. By incorporating various sources of evidence, the application of learning to rank (LTR) algorithms to real-time Twitter search has shown beneficial in finding not only relevant, but also recent tweets in response to given queries. However, the potential effectiveness brought by LTR may not have been fully exploited due to the lack of labeled data available for properly learning a ranking model, since human labels are expensive in real-world applications. To this end, this paper proposes a transductive algorithm that incrementally aggregate the labeled tweets through an iterative process. Experimental results on the standard Tweets11 dataset show that our approach is able to outperform strong baselines without the use of human labels.

Introduction

With the rapid development of online social networks, Twitter, one of the most popular microblogging services, is attracting more and more attention from Internet users (Kwak et al. 2010). The overwhelming amount of updates on Twitter leads to the difficulty in finding the interesting messages, which are fresh and relevant to the given query, whereby a user's information need is represented by a query issued at a specific time.

To achieve an effective real-time Twitter search, quite a few previous studies attempt to adopt the classical language model by introducing the temporal factors (Li and Croft 2003; Efron and Golovchinksy 2011). However, there exists also the difficulty in integrating multiple features that are intrinsic in online social networks, such as user authority, mentions, retweets, hashtags, and so on (Kwak et al. 2010).

Recently, there have been efforts in applying learning to rank to Twitter search by integrating various sources of evidence of relevance. Learning to rank (LTR), as known as machine-learned ranking, is a family of algorithms and techniques that automatically learn a ranking model from training data, where explicit or implicit evidence of relevance are usually represented by a feature vector. Despite the advantage of combining the predefine features to construct a ranking model, it may also suffer from the lack of labeled examples required for learning an effective ranking model, especially in the case of real-time Twitter search, where manual annotation is usually expensive and time consuming to obtain (Ounis et al. 2011).

In this paper, we propose a transductive algorithm to deal with the lack of labeled examples for learning a ranking model for real-time Twitter search. Transductive learning (Vapnik 1998) is a semi-supervised method for classification by utilizing limited data, which we believe is suitable for learning a ranking model with limited or no training data available. We apply transductive learning to train a ranking model by utilizing the existing training data, and gradually expand the labeled data set from the test examples. The incremental aggregation of labeled examples for LTR has been studied by (Duh and Kirchhoff 2008). Moreover, (Huang et al. 2006) applied a co-training algorithm to improve pseudo relevance feedback for passage retrieval.

The major contributions of this paper are two-fold. First, extracting a variety of features for implying the relevance and freshness of tweets. Various sources of features are incorporated into a learning to ranking paradigm to represent the tweets. Second, proposing a transductive learning algorithm to deal with the lack of training data in real-time Twitter search. As the quality of labeled data is essential to learning an effective ranking model, our proposed approach can significantly improve the feasibility of the LTR approaches.

Related Work

Many previous approaches to real-time retrieval are based on extensions of the content-based weighting models (Li and Croft 2003; Efron and Golovchinksy 2011). Despite the improvement over the classical content-based weighting models, the above described approaches ignore the many aspects and characteristics of the social features in microblogging services (Kwak et al. 2010; Cha et al. 2010; Duan et al. 2010). Recently, there have been research in applying LTR approaches for real-time Twitter search for its advantage in combining different features to automatically learn a ranking model from the training data.

LTR is a family of methods and algorithms that automatically construct a model or function to rank objects. One of the major advantages of LTR is its flexibility in incorporating diverse sources of evidence into the pro-

cess of retrieval (Liu 2009). It is widely accepted that there are three types of LTR algorithms, namely the pointwise, pairwise and listwise approaches. Experimentation in literature in general conclude that the pairwise and listwise approaches have superior retrieval performance than the pointwise approach (Joachims 2002; Cao et al. 2007; Wu et al. 2008). In this paper, we therefore apply three state-of-the-art LTR approaches, including the pairwise Ranking SVM (RankSVM) approach (Joachims 2002), and two recent listwise approaches, namely ListNet (Cao et al. 2007) and LambdaMART (Wu et al. 2008).

There have been efforts in applying LTR for real-time Twitter search (Metzler and Cai 2011; Miyanishi et al. 2011), especially as demonstrated in the TREC 2011 Microblog track (Ounis et al. 2011). Despite the effectiveness of LTR reported for Twitter search, labeled examples are required to facilitate the LTR methods. In addition to the cost of the human labels, the experimentation in the Microblog track does not show evident advantage brought by LTR over conventional retrieval methods (Ounis et al. 2011; Amati et al. 2011). We argue that the limited improvement brought by LTR is mainly due to the lack of training data. In particular, the human labels from the TREC participants may not be adequate for learning an effective ranking model. To address this problem, a transductive algorithm is put forward in the paper to incrementally generate relevance labels.

Learning to Rank with Transduction

Features Extraction describes the predefined features that are used for representing the tweets. *Transductive Learning* presents the algorithm of our proposed transductive learning.

Features Extraction

Our predefined features are organized around the basic entities for each query-tweet tuple to distinguish between the relevant and irrelevant messages. Table 1 presents all the features exploited in this paper. More specifically, five types of features are defined as follows.

- **Content-based relevance** is the content-based relevance score. In this paper, four popular content-based retrieval models and their corresponding query expansion or pseudo relevance feedback methods are applied. The parameters in the above content-based models are optimized using Simulated Annealing (Kirkpatrick, Gelatt, and Vecchi 1983) on the Blogs06 collection, which is a median crawl of blog posts and feeds used by the TREC Blog track 2006-2008 (Ounis, Macdonald, and Soboroff 2008).
- **Content richness** indicates how informative a tweet is.
- **Recency** refers to those features that indicate the temporal relationships between the query's submitted time and the tweet's posted time.
- **Authority** measures the influences of the author's tweets to others.
- **Tweet specific** features are those specific to given tweets, like RT, mentions and hashtags.

Input

R: initial retrieved tweets returned by content-based retrieval models for a batch of query topics
U: a set of unlabeled tweets
I: a set of labeled tweets, if any
F: a set of features representing the tweets
k: the number of first and last ranked tweets to be initially added to I, if I is empty
N: the number of iterations

Output

I: an expanded set of labeled tweets

Initialization

If I is empty, add the first and last ranked k tweets in R into I as positive and negative examples, respectively

Method

loop (for N iterations){
(1) Learn a ranking model C from I using F to represent the tweets
(2) Use C to rank tweets in U represented by F
(3) For all the topics, select the very first and last ranked tweets T from U
(4) Add T to I and remove T from U }

Figure 1: The transductive learning algorithm.

Transductive Learning

In this paper, a transductive learning algorithm, denoted by TLTR, is devised to facilitate learning to rank by generating a set of training data from limited or no human labels. Using transduction, it is not necessary to generate a general model to predict the label of any unobserved point during the process of learning. Before training, it is only required to predict the labels of a given test set examples (El-yaniv and Pechyony 2006).

A general description of the proposed method is given in Figure 1. The underlying idea of our proposed method is similar to that of pseudo relevance feedback (Rocchio 1971). In our proposed TLTR approach, a small amount of labeled examples are input to initialize the transduction process. In case of the unavailability of labeled data, the common highly ranked tweets from the four content-based retrieval models as given in Table 1 are selected as the initial positive training examples. We assume that the first returned tweets are highly relevant to the given query topic. In contrast, the last ranked tweets in the initial retrieval results are selected as the negative examples, which are assumed to be off-topic and irrelevant. Ranking of the remaining tweets are estimated by a model learned from the labeled examples. Such a process repeats until the number of iterations exceeds a predefined threshold. In each iteration, the most highly and bottom ranked tweets are added to the labeled data set. Finally, the expanded training data set is used for facilitating the LTR approaches for real-time Twitter search.

In this paper, k is arbitrarily fixed to 3 to reduce the cost in the parameter tuning. N, the number of iterations, is treated as a free parameter that require a grid search to locate a safe range of its values.

Table 1: Features predefined for representing the tweets.

Feature Type	Feature Name	Description
Content-based relevance	BM25	relevance score given by BM25
	PL2	relevance score given by PL2
	DirKL	relevance score given by the KL-divergence language model with Dirichlet smoothing
	KLIM	scores applying the KLIM model
	BM25QE	relevance score given by query expansion on top of BM25
	PL2QE	relevance score given by query expansion on top of PL2
	DirKLQE	relevance score given by query expansion on top of DirKL
	KLIMQE	relevance score given by query expansion on top of KLIM
Content richness	Content Length	the length of a tweet in words
	URL	whether the tweet contains URL
	OOV	the ratio of the unique words in the tweet
	RTRATIO	the ratio of appended comments after retweet
Authority	Global RT Count	the number of the retweets given an author, no matter what topic the author is talking about
	Local RT Count	the number of the retweets given an author and a particular topic.
Recency	DAY_QUERY_DIF	difference in days between the time of the query's issuing and the tweet's posted time
	DAY_BURST_DIF	differences in days between the burst time of an event and a tweet's posted time
	Ratio_DAY_QUERY_DIF	the ratio of DAY_QUERY_DIF of a given tweet in a day
	Ratio_DAY_BURST_DIF	the ratio of DAY_BURST_DIF of a given tweet in a day
Tweet specific	BEGIN RT	whether the tweet begins with a RT tag
	CONTAIN RT	whether the tweet is retweeted with appended comments
	BEGIN AT	whether the tweet is a replied tweet, i.e. beginning with an @
	AT COUNT	how many users are mentioned in the tweet with an @
	English?	whether the tweet is written in English

In the transductive learning process, after the positive examples are appended to the labeled set, we assign preference values according to the temporal distance between the tweet's timestamps and the query's submission. The larger the preference value is, the higher the tweet is relevant to the given query. This labeling strategy is mainly due to the fact that recency is a crucial factor of relevance in real-time Twitter search. The fresh tweets are favored over those outdated.

Experiments

Dataset

In 2011, a new task called the Microblog track is introduced to provide a benchmark for research in Twitter search (Ounis et al. 2011). The task of the Microblog Track is to retrieve relevant tweets for each query at a specific time. Thus, for each query all relevant tweets that are ordered from the newest to the oldest should be returned, and all tweets must be posted before the query is issued. We experiment on the Tweets11 collection, which consists of a sample of tweets over a period of about 2 weeks spanning from January 24, 2011 to February 8, 2011. In the TREC 2011 Microblog track, this collection is used for evaluating the participating real-time Twitter search systems over 50 official topics. In addition, 12 example topics are given for prior training without human labels. The official measure in the TREC 2011 Microblog Track, namely precision at 30 (P30), is used as the evaluation metric in our experiments. Standard stopword removal and Porter's stemmer are applied during indexing and retrieval. All experiments are conducted by an in-house version of the Terrier platform (Ounis et al. 2006). Our crawl of the Tweets11 collection consists of 13,401,964 successfully downloaded unique tweets. Note that because of dynamic nature of Twitter, there is a discrepancy between the number of tweets in our crawl and the figures reported by other participants in the Microblog track 2011. This does not affect the validity of the conclusions drawn from our experiments as the proposed approach and the baselines are evaluated on the same dataset with consistent settings.

Table 2: Results obtained by four content-based models with query expansion.

	BM25QE	PL2QE	DirKLQE	KLIMQE
P30	0.3429	0.3537	0.3571	**0.3782**

Results and Discussions

The aim of our experiments is to evaluate the effectiveness of the proposed transductive learning algorithm in combination with LTR approaches for real-time Twitter search. Table 2 outlines the precisions at 30 given by the four baselines. As shown in this Table, KLIMQE turns out to have the best effectiveness over other baselines.

We firstly compare our proposed approach (TLTR) to the content-based retrieval models with query expansion. The results obtained by our proposed method over 62 topics and leave-one-out cross-validation with the optimal setting of N are shown in Table 3. In this table, a star indicates a statistically significant improvement over KLIMQE, which has the best retrieval performance among the four content-based baselines. As shown by the results, using all the 62 topics for simulating the training data set (TLTR_All_No) leads to significant improvement over the baseline with all three LTR algorithms. On the other hand, using the leave-one-out cross-validation (TLTR_L1_No), the transductive algorithm significantly outperforms the baseline with RankSVM, while no significant improvement is observed with ListNet and lambdaMART. As TLTR_All_No leads to in general better retrieval performance than TLTR_L1_No, in the rest of this paper, we only report the results obtained by TLTR_All_No.

Next, our proposed transductive learning algorithm is compared to the state-of-the-art LTR approaches, which learns the ranking models from the official relevance as-

Table 3: Comparison of the transductive learning with the KLIMQE baseline. No labeled data are used for training.

KLIMQE	RankSVM	ListNet	LambdaMART
	TLTR_All_No		
0.3782	0.3959, +4.68*%	0.3966 +4.87*%	**0.4212, +11.37*%**
	TLTR_L1_No		
0.3782	**0.4014,+6.13*%**	0.3728, -1.43%	0.3796, +0.37%

Table 4: Comparison of TLTR with the LTR approaches using relevance assessments.

Full relevance info.			No relevance info.
LTR_10_Full	LTR_5_Full	LTR_2_Full	TLTR_All_No
Results obtained using RankSVM			
0.3912	0.3748	0.3571	**0.3959,+1.20%**
Results obtained using ListNet			
0.3633	0.3517	**0.3735**	**0.3966,+6.18*%**
Results obtained using LambdaMART			
0.3912	0.2782	0.3238	**0.4212,+7.67*%**

sessments. 2, 5, and 10-fold cross-validations are conducted to evaluate the effectiveness of the LTR approaches. Our proposed approach, without the use of relevance information, is compared to the best results obtained by each of the three LTR approaches in Table 4. The results show that our proposed transductive learning method is able to significantly outperform the state-of-the-art LTR approaches with a proper setting of the parameter *N*.

Conclusions and Future Work

In summary, we have proposed a transductive learning algorithm that generates the training data while no labeled data is available. As shown by experiments on the standard Tweets11 dataset, our proposed approach can outperform the classical content-based retrieval models. Additional experiments with the full relevance assessment information demonstrate that our proposed method can not only facilitate the state-of-the-art learning to rank approaches for real-time Twitter search, but also provide at least comparable retrieval performance of the ranking models learned from full relevance assessments. In the future, we plan to improve the robustness of the proposed approach by introducing an adaptive halting criterion of the iterative process. For example, we can introduce a quality factor that monitors the cohesiveness or purity of the training data set generated. The iterative transduction process stops when the quality factor is lower than a given threshold. We will also investigate the possibility of applying other semi-supervised learning paradigms, e. g. co-training, to further improve the proposed algorithm.

Acknowledgements

This work is supported in part by the NSFC Project 61103131/F020511, the President Fund of GUCAS (Y15101FY00), and the CAS R&E Project (110700EA12).

References

Amati, G.; Amodeo, G.; Bianchi, M.; Celi, A.; Nicola, C. D.; Flammini, M.; Gaibisso, C.; Gambosi, G.; and Marcone, G. 2011. FUB, IASI-CNR, UNIVAQ at TREC 2011. In *TREC*.

Blum, A., and Mitchell, T. M. 1998. Combining labeled and unlabeled data with co-training. In *COLT*, 92–100.

Cao, Z.; Qin, T.; Liu, T.-Y.; Tsai, M.-F.; and Li, H. 2007. Learning to rank: from pairwise approach to listwise approach. In *ICML*, 129–136.

Cha, M.; Haddadi, H.; Benevenuto, F.; and Gummadi, K. P. 2010. Measuring user influence in twitter: The million follower fallacy. In *ICWSM*.

Duan, Y.; Jiang, L.; Qin, T.; Zhou, M.; and Shum, H.-Y. 2010. An empirical study on learning to rank of tweets. In *COLING*, 295–303. Beijing, China: Tsinghua University.

Duh, K., and Kirchhoff, K. 2008. Learning to rank with partially-labeled data. In *SIGIR*, 251–258.

Efron, M., and Golovchinksy, G. 2011. Estimation methods for ranking recent information. In *SIGIR*, 495–504. New York, NY, USA: ACM.

El-yaniv, R., and Pechyony, D. 2006. Stable transductive learning. In *COLT*, 35–49.

Huang, X.; Huang, Y. R.; Wen, M.; An, A.; Liu, Y.; and Poon, J. 2006. Applying data mining to pseudo-relevance feedback for high performance text retrieval. In *ICDM*.

Joachims, T. 2002. Optimizing search engines using clickthrough data. In *KDD*, 133–142. ACM.

Kirkpatrick, S.; Gelatt, C.; and Vecchi, M. 1983. Optimization by simulated annealing. *Science* 220(4598).

Kwak, H.; Lee, C.; Park, H.; and Moon, S. 2010. What is twitter, a social network or a news media? In *WWW*, 591–600.

Li, X., and Croft, W. B. 2003. Time-based language models. In *CIKM*, 469–475. ACM.

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

Metzler, D., and Cai, C. 2011. Usc/isi at trec 2011: Microblog track (notebook version). In *TREC*.

Miyanishi, T.; Okamura, N.; Liu, X.; Seki, K.; and Uehara, K. 2011. Trec 2011 microblog track experiments at kobe university. In *TREC*.

Ounis, I.; Amati, G.; Plachouras, V.; He, B.; Macdonald, C.; and Lioma, C. 2006. Terrier: A high performance and scalable information retrieval platform. In *SIGIR OSIR*.

Ounis, I.; Macdonald, C.; Lin, J.; and Soboroff, I. 2011. Overview of the TREC 2011 microblog track. In *TREC*.

Ounis, I.; Macdonald, C.; and Soboroff, I. 2008. On the TREC blog track. In *ICWSM*.

Rocchio, J. 1971. *Relevance feedback in information retrieval*. Prentice-Hall Englewood Cliffs. 313–323.

Vapnik, V. N. 1998. *Statistical learning theory*. New York: Wiley.

Wu, Q.; Burges, C.; Svore, K.; and Cao, J. 2008. Ranking boosting and model adaptation. Technical report, Microsoft.