Quality-biased Ranking of Short Texts in Microblogging Services

Minlie Huang
Dept. of Computer Science and
Technology, Tsinghua University
Beijing 100084, China
aihuang@tsinghua.edu.cn

Yi Yang
School of Software
Beihang University
Beijing, China
yangyiycc@gmail.com

Xiaoyan Zhu
Dept. of Computer Science and
Technology, Tsinghua University
Beijing 100084, China
zxy_dcs@tsinghua.edu.cn

Abstract

The abundance of user-generated content comes at a price: the quality of content may range from very high to very low. We propose a regression approach that incorporates various features to recommend short-text documents from Twitter, with a bias toward quality perspective. The approach is built on top of a linear regression model which includes a regularization factor inspired from the content conformity hypothesis - documents similar in content may have similar quality. We test the system on the Edinburgh Twitter corpus. Experimental results show that the regularization factor inspired from the hypothesis can improve the ranking performance and that using unlabeled data can make ranking performance better. Comparative results show that our method outperforms several baseline systems. We also make systematic feature analysis and find that content quality features are dominant in short-text ranking.

1 Introduction

More and more user-generated data are emerging on personal blogs, microblogging services (e.g. Twitter), social and e-commerce websites. However, the abundance of user-generated content comes at a price: there may be high-quality content, but also much spam content such as advertisements, self-promotion, pointless babbles, or misleading information. Therefore, assessing the quality of information has become a challenging problem for many tasks such as information retrieval, review mining (Lu et al., 2010), and question answering (Agichtein et al., 2008).

In this paper, we focus on predicting the quality of very short texts which are obtained from Twitter. Twitter is a free social networking and microblogging service that enables its users to send and read other users' updates, known as ”Tweets”. Each tweet has up to 140 characters in length. With more than 200 million users (March 2011), Twitter has become one of the biggest mass media to broadcast and digest information for users. It has exhibited advantages over traditional news agencies in the success of reporting news more timely, for instance, in reporting the Chilean earthquake of 2010 (Mendoza et al., 2010). A comparative study (Teevan et al., 2011) shows that queries issued to Twitter tend to seek more temporally relevant information than those to general web search engines.

Due to the massive information broadcasted on Twitter, there are a huge amount of searches every day and Twitter has become an important source for seeking information. However, according to the Pear Analytics (2009) report on 2000 sample tweets, 40.5% of the tweets are pointless babbles, 37.5% are conversational tweets, and only 3.6% are news (which are most valuable for users who seek news information). Therefore, when a user issues a query, recommending tweets of good quality has become extremely important to satisfy the user's information need: how can we retrieve trustworthy and informative posts to users?

However, we must note that Twitter is a social networking service that *encourages* various content such as news reports, personal updates, babbles, conversations, etc. In this sense, we can not say which content has better quality without considering the value to the writer or reader. For instance, for a reader, the tweets from his friends or who he follows may be more desirable than those from others, whatever the quality is. In this paper, we have a special focus on finding tweets on news topics when we construct the evaluation datasets.

We propose a method of incorporating various features for quality-biased tweet recommendation in response to a query. The major contributions of this paper are as follows:

- We propose an approach for quality-biased ranking of short documents. Quality-biased is referred to the fact that we explore various features that may indicate quality. We also present a complete feature analysis to show which features are most important for this problem.
- We propose a content conformity hypothesis, and then formulate it into a regularization factor on top of a regression model. The performance of the system with such a factor is boosted.
- It is feasible to plug unlabeled data into our approach and leveraging unlabeled data can enhance the performance. This characteristics is appealing for information retrieval tasks since only a few labeled data are available in such tasks.

The rest of the paper is organized as follows: in Section 2 we survey related work. We then formulate our problem in Section 3 and present the hypothesis in Section 4. Various features are presented in Section 5. The dataset and experiment results are presented in Section 6 and Section 7, respectively. We summarize this work in Section 8.

2 Related Work

2.1 Quality Prediction

Quality prediction has been a very important problem in many tasks. In review mining, quality prediction has two lines of research: one line is to detect spam reviews (Jindal and Liu, 2008) or spam reviewers (Lim et al., 2010), which is helpful to exclude misleading information; the other is to identify high-quality reviews, on which we will focus in this survey. Various factors and contexts have been studied to produce reliable and consistent quality prediction. Danescu-Niculescu-Mizil et al. (2009) studied several factors on helpfulness voting of Amazon product reviews. Ghose and Ipeirotis (2010) studied several factors on assessing review helpfulness including reviewer characteristics, reviewer history, and review readability and subjectivity. Lu et al. (2010) proposed a linear regression model with various social contexts for review quality prediction. The authors employed author consistency, trust consistency and co-citation consistency hypothesis to predict more consistently. Liu et al. (2008) studied three factors, i.e., reviewer expertise, writing style, and timeliness, and proposed a non-linear regression model with radial basis functions to predict the helpfulness of movie reviews. Kim et al. (2006) used SVM regression with various features to predict review helpfulness.

Finding high-quality content and reliable users is also very important for question answering. Agichtein et al. (2008) proposed a classification framework of estimating answer quality. They studied content-based features (e.g. the answer length) and usage-based features derived from question answering communities. Jeon et al. (2006) used non-textual features extracted from the Naver Q&A service to predict the quality of answers. Bian et al. (2009) proposed a mutual reinforcement learning framework to simultaneously predict content quality and user reputation. Shah and Pomerantz (2010) proposed 13 quality criteria for answer quality annotation and then found that contextual information such as a user's profile, can be critical in predicting the quality of answers.

However, the task we address in this paper is quite different from previous problems. First, the document to deal with is very short. Each tweet has up to 140 characters. Thus, we are going to investigate those factors that influence the quality of such short texts. Second, as mentioned, high-quality information on Twitter (e.g., news) is only a very small proportion. Thus, how to distill high quality content from majority proportions of low-quality content may be more challenging.

2.2 Novel Applications on Twitter

Twitter is of high value for both personal and commercial use. Users can post personal updates, keep tight contact with friends, and obtain timely information. Companies can broadcast latest news to and interact with customers, and collect business intelligence via opinion mining. Under this background, there has been a large body of novel applications on Twitter, including social networking mining (Kwark et al., 2010), real time search[1] , sentiment analysis[2], detecting influenza epidemics (Culotta, 2010), and even predicting politics elections (Tumasjan et al., 2010).

As Twitter has shown to report news more timely than traditional news agencies, detecting tweets of news topic has received much attention. Sakaki et al. (2010) proposed a real-time earthquake detection framework by treating each Twitter user as a sensor. Petrovic et al. (2010) addressed the problem of detecting new events from a stream of Twitter posts and adopted a method based on locality-sensitive hashing to make event detection feasible on web-scale corpora. To facilitate fine-grained information extraction on news tweets, Liu et al. (2010) presented a work on semantic role labeling for such texts. Corvey et al. (2010) proposed a work for entity detection and entity class annotation on tweets that were posted during times of mass emergency. Ritter et al. (2010) proposed a topic model to detect conversational threads among tweets.

Since a large amount of tweets are posted every day, ranking strategies is extremely important for users to find information quickly. Current ranking strategy on Twitter considers relevance to an input query, information recency (the latest tweets are preferred), and popularity (the retweet times by other users). The recency information, which is useful for real-time web search, has also been explored by Dong et al. (2010) who used fresh URLs present in tweets to rank documents in response to recency sensitive queries. Duan et al. (2010) proposed a ranking SVM approach to rank tweets with various features.

3 Problem Formulation and Methodology

Given a set of queries $Q = \{q_1, q_2, \cdots, q_n\}$, for each query q_k, we have a set of short documents $D_k = \{d_k^1, d_k^2, \cdots\}$ which are retrieved by our built-in search engine. The document set D_k is partially labeled, i.e., a small portion of documents in D_k were annotated with a category set C=$\{$1, 2, 3, 4, 5$\}$ where 5 means the highest quality and 1 lowest. Therefore, we denote $D_k = D_k^U \cup D_k^L$, where D_k^U indicates the unlabeled documents, and D_k^L the labeled documents. Each document in D_k is represented as a feature vector, $d_i = (x_1, x_2, \cdots, x_m)$ where m is the total number of features.

The learning task is to train a mapping function $f(\mathcal{D}) : \mathcal{D} \rightarrow \mathcal{C}$, to predict the quality label of a document given a query q. We use a linear function $f(\mathbf{d}) = \mathbf{w^T d}$ for learning and where $\mathbf{w}$ is the weight vector. Formally, we define an objective function as follows to guide the learning process:

$$\Theta(\mathbf{w}) = \frac{1}{n}\sum_{k=1}^{n}\frac{1}{\mid D_k^L \mid}\sum_{\mathbf{d_i}\in D_k^L}\ell(\mathbf{w}^T\mathbf{d_i}, \hat{y}_i) + \alpha\mathbf{w}^T\mathbf{w} \tag{1}$$

where $\ell(.,.)$ is the loss function that measures the difference between a predicted quality $f(\mathbf{d_i}) = \mathbf{w}^T\mathbf{d_i}$ and the labelled quality $\hat{y}_i$, D_k^L is the labeled documents for query q_k, $\hat{y}_i$ is the quality label for document $\mathbf{d_i}$, n is the total number of queries, and α is a regularization parameter for $\mathbf{w}$. The loss function used in this work is the square error loss, as follows:

$$\ell(\mathbf{w}^T\mathbf{d_i}, y_i) = (\mathbf{w}^T\mathbf{d_i} - \hat{y}_i)^2 \tag{2}$$

It's easy to see that this problem has a closed-form solution, as follows:

$$\widehat{\mathbf{w}} = \arg\min_{\mathbf{w}}\Theta(\mathbf{w}) = (\sum_{i=1}^{N_l}\mathbf{d_i}\mathbf{d_i}^T + \alpha N_l\mathbf{I})^{-1}\sum_{i=1}^{N_l}\hat{y}_i\mathbf{d_i} \tag{3}$$

where $\mathbf{I}$ is an identity matrix of size m (the dimension of feature vector), and N_l is the total number of labeled documents in all the queries. As mentioned, there are a large number of documents retrieved for each query while we only sample a small number of documents for manual annotation. Thus there are much more unlabeled documents yet to be utilized.

4 Content Conformity Hypothesis

To make quality prediction more consistent and to utilize the unlabeled data, we propose the content conformity hypothesis which assumes that the quality of documents similar in content should be close to each other. This hypothesis can be formulated as a regularization factor in the objective, as follows:

$$\Theta_1(\mathbf{w}) = \Theta(\mathbf{w}) + \beta\sum_{k=1}^{n}\sum_{\substack{d_i, d_j \in D_k \\ \wedge IsSim(d_i, d_j)}}(\mathbf{w^T d_i} - \mathbf{w^T d_j})^2 \tag{4}$$

[1] http://twittertroll.com/

[2] http://twittersentiment.appspot.com/

where $IsSim(d_i, d_j)$ is a predicate asserting that two documents are similar, and β is an empirical parameter. Note that D_k is usually all labeled data but it may also include *unlabeled* documents for query q_k. In this way, we can utilize the unlabeled documents as well as the labeled ones. There are various ways to determine whether two documents of the same query are similar. One way is to use TF*IDF cosine similarity to find similar documents with a threshold, and another way is to use clustering where two documents in the same cluster are similar. We use the first means in this paper and leave the second for future work.

To obtain the closed-form solution of Eq. 4, we define an auxiliary matrix $A = (a_{ij})$ where each a_{ij} is 1 if document d_i is similar to document d_j for some query. Then, Eq. 4 can be re-written as follows:

$$\Theta_1(\mathbf{w}) = \Theta(\mathbf{w}) + \beta \sum_{i<j} a_{ij}(\mathbf{w^T d_i} - \mathbf{w^T d_j})^2 \quad (5)$$

Let $\mathbf{D} = [\mathbf{d_1}, \mathbf{d_2}, \ldots, \mathbf{d_N}]$ be an $m \times N$ matrix in which each $\mathbf{d_i}$ is a column feature vector for a document. Note that this matrix includes both labeled and unlabeled documents, and N is the total number of documents. Then the last term in Eq. 5 can be re-written as

$$\sum_{i<j} a_{ij}(\mathbf{w^T d_i} - \mathbf{w^T d_j})^2 = \mathbf{w^T D \Lambda_A D^T w} \quad (6)$$

where $\Lambda_A = \Delta_A - A$ and Δ_A is a diagonal matrix with $(\Delta_A)_{ii} = \sum_j a_{ij}$. By some mathematical manipulations, the problem in Eq. 6 has the following closed-form solution (Zhu and Goldberg, 2009):

$$\widehat{\mathbf{w}} = (\sum_{i=1}^{N_l} \mathbf{d_i d_i}^T + \alpha N_l I + \beta N_l \mathbf{D \Lambda_A D^T})^{-1} \sum_{i=1}^{N_l} \hat{y}_i \mathbf{d_i} \quad (7)$$

5 Features

We design several groups of features to indicate the quality of tweets from different perspectives. These features include: content quality, user profile and authority, sentiment polarity, query relevance, and Twitter specific features.

5.1 Content Quality

Documents with higher quality in content will be more desirable for users in search. We thus exploit several features to respect quality:

Tweet's length: longer tweet may be more informative as each tweet has been limited to up to 140 characters.

Average term similarity: each tweet d has a score of $\frac{1}{|D_i|} \sum_{d_i \in D_i} sim(d, d_i)$ where D_i is the document set for query q_i, and $sim(.,.)$ is a cosine TF*IDF similarity measure for two documents.

Ratio of unique words: in some tweets, the same word is repeated many times while there are only few unique words. Tweets with more unique words may have more information. The number of unique words is normalized by the total number of words.

Ratio of POS tags: We compute the ratio of nouns, verbs, adverbs, adjectives, etc. in a tweet. Each POS tag corresponds to one dimension in the feature vector.

5.2 User Profile and User Authority

A user with a real and complete profile may post tweets responsibly and accountably. Authoritative users (particularly celebrity users) are more probably to post high-quality tweets.

Profile integrity: we have several features for measuring this property: whether the user of a tweet has a description field, whether the description field contains a url, whether the user verifies her account via the registered email, and whether the user provides the location information.

User Activeness: the average number of tweets that the user posted per day and how many days a user has registered.

User authority: In the spirit of (Duan et al., 2010), we utilize follower score (the number of followers) , mention score (the number of times a user is referred to in tweets), popularity score to measure the authority of a user. The popularity score is obtained with the PageRank algorithm based on retweet relationship (two users have an edge in the graph if a tweet posted by one user is retweeted by another user).

5.3 Sentiment

As mentioned, Twitter has become a popular site for expressing opinions and personal sentiment towards public persons or events. Thus we believe that a tweet with clear sentiment polarity will be more favorable for users. Therefore, we adopt a sentiment lexicon (SentiWordNet) and collect the top 200 frequent emoticons from our tweet corpus to identify positive and negative sentiment words.

Positive sentiment: the ratio of positive sentiment words or emoticons in a tweet.

Negative sentiment: the ratio of negative sentiment words or emoticons.

Sensitive words: the number of sensitive words. We manually collect 458 offending or pornographic words.

The emoticon lexicon and the sensitive word lexicon will be available to public.

5.4 Twitter-Specific Features

Tweet has its own characteristics which may be used as features, such as whether a tweet contains a common url (such as http://www.google.com), whether the url is a tinyurl (Twitter has a service which shortens urls to very short url), the number of hashtags (topical terms leading by a '#') in a tweet, how many users are mentioned in the tweet (a user is mentioned if the tweet contains a term like @user_name), and how many times the tweet has been re-posted (so-called 'retweeted').

5.5 Query-specific Features

As our task is to supply quality-biased ranking of tweets for an input query, query-specific features will favor those tweets relevant to the input query.

Query term frequency: the frequency of the query term (exact matching) in a tweet.

BM25 score: the BM25 score is used to quantify the overall relevance of a tweet to the query.

Recency: the time lag (in days) between the current tweet and the earliest tweet in the collection for the query. In this case, the more recent tweets may contain latest information, which will be more desirable for users.

6 Dataset

To investigate the factors that influence the quality of short texts, we use the Edinburgh Twitter Corpus (Petrovic et al., 2010)[3] in which each tweet has up to 140 characters. The corpus contains 97 million tweets, and takes up 14 GB of disk space uncompressed. The corpus was collected through the Twitter streaming API from a period spanning November 11th 2009 until February 1st 2010. Each tweet has some meta-data: the timestamp of the tweet, an anonymized user name, the textual content, and the posting source (via web, mobile, etc.).

We collect a set of news queries using Google Trends. Intuitively, those hot queries in Google Trends will also have high possibility to be discussed on Twitter. The top 10 queries per day captured by Google Trends for the period 11th November, 2009 to 1st February, 2010 are collected. We then randomly sample 60 hot queries from these queries. And for each query, we use our own built-in search engine (based on BM25) to retrieve a set of tweets for manual annotation. To minimize the labeling cost, for each query, we sample 150-200 tweets for annotation as each query may return thousands of results, which makes the complete annotation impossible. These queries are grouped into four categories: thing (10 queries), person (15), event (30) and place (5). Table 1 shows some example queries of each type. For all these queries, there are about 9,000 unique tweets to be annotated.

Then, two computer science students were asked to annotate the tweets. The quality of a tweet was judged to a 5-star likert scale, according to the relevance, informativeness, readability, and politeness of the content. If the label difference of two tweets is larger than 1, the tweets were re-annotated until the quality difference is within 1.

Thing	**Person**
haiti relief effort	adam james
newegg	tiger woods mistress
flight 253	jennifer jones
groupon	john wall
Event	**Place**
obama ambulance	google headquarters
boeing 787 first flight	solomon islands
eureka earthquake	humanitarian bowl
shaq retires	college of charleston

Table 1: Sample queries for each query type.

6.1 Evaluation Metrics

We adopt information retrieval metrics to measure the performance since the task can be viewed as a ranking task (ranking document according to its quality). $nDCG$ (Järvelin and Kekäläinen., 2000) is used to evaluate the ranking performance, as follows:

$$nDCG(\Omega, k) = \frac{1}{|\Omega|} \sum_{q \in \Omega} \frac{1}{Z_q} \sum_{i=1}^{k} \frac{2^{r_i^q} - 1}{log(1+i)}$$

where Ω is the set of test queries, k indicates the top k positions in a ranked list, Z_q is a normalization factor obtained from a perfect ranking (based on the labels), and r_i^q is the relevance score (the annotated quality label) for the *i-th* document in the predicted ranking list for query q. We also evaluate the system in terms of MAP [4] where the document whose quality score is larger than 3 is viewed as relevant and otherwise irrelevant.

Note that the ranking task is approached as a regression problem, mean square error is thus adopted to measure the learning performance:

$$MSE(\mathcal{D}) = \frac{1}{|\mathcal{D}|} \sum_{d_j \in \mathcal{D}} (f(d_j) - \hat{y}_j)^2$$

where $\mathcal{D}$ is the test document collection, $f(d_j)$ is the predicted label, and $\hat{y}_j$ is the annotated label.

$nDCG$ and MSE have a significant difference in that $nDCG$ only considers the top k documents for each

[3] Though the corpus is not available now on the original website due to licensing problems, readers are encouraged to request a copy from us. We are downloading a new dateset for further evaluation.

[4] `http://en.wikipedia.org/wiki/Information_retrieval`

query while MSE takes into account all documents in the test collection.

7 Experiment and Evaluation

In this section, we will first assess whether the proposed hypothesis holds on our labeled data. We then evaluate whether the performance of the model with the regularization factor (as defined in Eq. 4) can be enhanced. We next compare the regression model with several baselines: BM25 model, Length Model (tweets containing more words may have better quality), ReTweet Model (tweets of higher quality may be re-posted by more users), and a Learning-to-Rank model (L2R) as used in (Duan et al., 2010)(a ranking SVM model). Finally, we investigate the influence of different feature groups on the performance. We conduct five-fold cross validation in the following experiments (3/5 partitions are for training, 1/5 are used as a validation set, and the left for test).

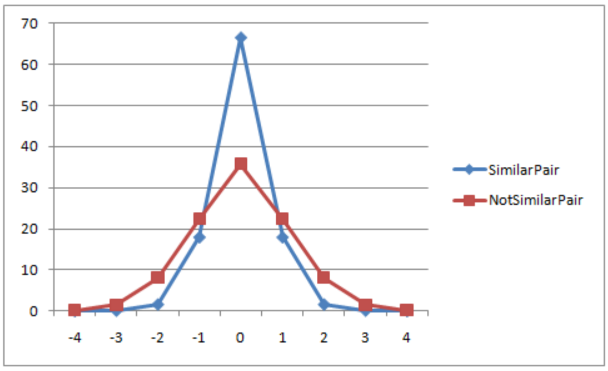

Figure 1: The hypothesis holds on the annotated dataset. y-axis is the percentage of pairs and x-axis is the quality difference between two documents in a pair.

7.1 Hypothesis Evaluation

We will evaluate whether the content conformity hypothesis holds on our manually annotated dataset. To define the similarity predicate ($IsSim$ in Eq. 4), we assume two documents are similar if their TF*IDF cosine similarity is no less than 0.6. We then compute the statistics of the quality difference of similar pairs and that of dissimilar pairs. We find that more than 53% similar pairs have exactly identical quality labels, out of all similar pairs. And more than 93% similar pairs have a quality difference within 1. For dissimilar pairs, only 35% pairs have identical quality labels. This shows that if two documents are similar, there is high probability that their quality labels are close to each other, and that if two documents are dissimilar, it's more likely that they have more divergent quality scores. These statistics are shown in Figure 1.

As shown in the figure, we can see that the hypothesis holds. Therefore, we can safely formulate the hypothesis into a regularization factor in the subsequent experiments.

7.2 Parameter Tuning

We explain here how the parameters (α, β) are chosen. In Table 2, we can see clearly that the best performance is obtained when $\alpha = 1e-8$. In Table 3, the model that utilizes only labeled data obtains most of the best nDCG scores when $\beta = 0.001$. For the MAP metric, the scores when $\beta = 0.001$ and $\beta = 0.0001$ are very close. Unlike MSE that considers all documents in the test collection, nDCG only considers the top ranked documents, which are more desirable for parameter choosing since most users are only interested in top ranked items. In Table 4, the model that utilizes unlabeled data obtains best performance when $\beta = 0.0001$. These optimal parameters will be used in our subsequent experiments.

7.3 Influence of the Regularization Factor

In this section, we address two issues: (1) whether the regularization factor inspired by content conformity hypothesis (Eq. 4) can improve the performance; and (2) whether the performance can be improved if using unlabeled data (see D_k in Eq. 4).

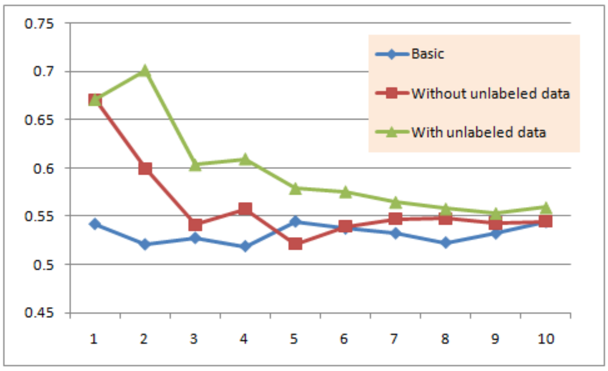

Figure 2: nDCG performance (y-axis) for top k ranks. The similarity predicate ($IsSim(a, b)$ in Eq. 4) is implemented with TF*IDF cosine similarity.

As shown in Figure 2, under the hypothesis the ranking performance is boosted compared to the basic model. As shown in Eq. 4, unlabeled data can be included in the regularization factor, thus we add the same number of unlabeled documents[5] for each query. We conduct experiments with and without such unlabeled data respectively. Adding unlabeled data can improve the ranking performance. This is appealing for most IR applications since

[5] Arbitrary number of documents may be added but we will evaluate this as future work.

α	1e-10	1e-9	1e-8	1e-7	1e-6	1e-5	0.0001	0.001	0.01
nDCG@1	0.131	0.325	**0.542**	**0.542**	**0.542**	**0.542**	**0.542**	**0.542**	**0.542**
nDCG@2	0.180	0.390	**0.521**	0.521	**0.521**	**0.521**	**0.521**	**0.521**	0.464
nDCG@3	0.209	0.377	**0.527**	**0.527**	**0.527**	**0.527**	**0.527**	0.512	0.424
nDCG@4	0.240	0.342	**0.518**	**0.518**	**0.518**	**0.518**	**0.518**	**0.518**	0.425
nDCG@5	0.235	0.350	**0.545**	0.516	0.516	0.516	0.516	0.518	0.444
nDCG@6	0.244	0.389	**0.537**	0.504	0.504	0.504	0.528	0.527	0.440
nDCG@7	0.250	0.399	**0.532**	0.528	0.528	0.528	0.530	0.523	0.432
nDCG@8	0.281	0.403	**0.522**	0.519	0.519	0.519	0.516	0.514	0.435
nDCG@9	0.291	0.411	**0.532**	0.517	0.517	0.517	0.518	0.511	0.450
nDCG@10	0.300	0.422	**0.544**	0.535	0.535	0.535	0.522	0.517	0.466
MAP	0.177	0.253	**0.390**	0.378	0.378	0.377	0.372	0.360	0.293
MSE	37.983	3.914	**1.861**	1.874	1.875	1.880	1.914	1.920	1.970

Table 2: The performance of different α parameters. The bolded cells show the optimal performance.

β	1e-10	1e-9	1e-8	1e-7	1e-6	1e-5	0.0001	0.001	0.01
nDCG@1	0.542	0.542	0.542	0.542	0.542	0.542	0.542	**0.671**	0.480
nDCG@2	0.521	0.521	0.521	0.521	0.521	0.570	0.570	**0.600**	0.472
nDCG@3	0.527	0.527	0.480	0.480	0.527	**0.596**	0.549	0.541	0.501
nDCG@4	0.520	0.518	0.481	0.481	0.518	0.543	0.515	**0.558**	0.468
nDCG@5	0.503	**0.528**	0.512	0.512	0.516	0.521	0.505	0.521	0.448
nDCG@6	0.514	0.515	0.521	0.521	0.511	0.522	0.533	**0.539**	0.468
nDCG@7	0.524	0.523	0.524	0.524	0.529	0.517	0.517	**0.547**	0.461
nDCG@8	0.515	0.525	0.514	0.514	0.527	0.521	0.519	**0.548**	0.473
nDCG@9	0.518	0.519	0.513	0.513	0.524	0.536	0.521	**0.543**	0.470
nDCG@10	0.528	0.529	0.517	0.517	0.535	0.537	**0.545**	**0.545**	0.472
MAP	0.386	0.385	0.367	0.367	0.369	0.375	**0.388**	0.387	0.264
MSE	1.863	1.862	1.893	1.893	1.859	1.845	1.763	1.315	**0.908**

Table 3: The performance of different β parameters with only labeled data (α=1e-8 according to Table 2). The bolded cells show the optimal performance.

β	1e-10	1e-9	1e-8	1e-7	1e-6	1e-5	0.0001	0.001	0.01
nDCG@1	0.542	0.542	0.542	0.542	0.542	0.542	**0.671**	0.277	0.147
nDCG@2	0.521	0.521	0.521	0.521	0.521	0.570	**0.701**	0.429	0.146
nDCG@3	0.527	0.527	0.527	0.527	0.558	0.565	**0.603**	0.438	0.168
nDCG@4	0.518	0.518	0.486	0.518	0.522	0.550	**0.610**	0.437	0.208
nDCG@5	0.519	0.516	0.488	0.545	0.548	0.527	**0.579**	0.429	0.218
nDCG@6	0.514	0.518	0.499	0.537	0.547	0.528	**0.576**	0.431	0.235
nDCG@7	0.529	0.535	0.503	0.538	0.541	0.516	**0.565**	0.453	0.256
nDCG@8	0.520	0.525	0.503	0.528	0.530	0.532	**0.558**	0.454	0.286
nDCG@9	0.513	0.518	0.520	0.521	0.535	0.534	**0.553**	0.478	0.300
nDCG@10	0.523	0.536	0.524	0.533	0.536	0.540	**0.559**	0.481	0.313
MAP	0.385	0.382	0.374	0.389	0.394	0.381	**0.427**	0.296	0.164
MSE	1.845	1.868	1.896	1.842	1.819	1.613	0.803	**0.499**	0.697

Table 4: The performance of different β parameters with unlabeled data (α=1e-8 according to Table 2). The bolded cells show the optimal performance.

most IR problems only have a small number of labeled data available.

7.4 Comparison to Baselines

To demonstrate the performance of our approach, we compare our system to three unsupervised models and one supervised model. The unsupervised models are: the BM25 model, the Length model which ranks tweets by the document length in tokens, and the RTNum model which ranks tweets by the frequency of being re-posted. The supervised model is a ranking SVM model (L2R) that was used in (Duan et al., 2010). In this experiment, the model (as indicated by "Full" in Fig. 3) is the best model presented in the preceding section.

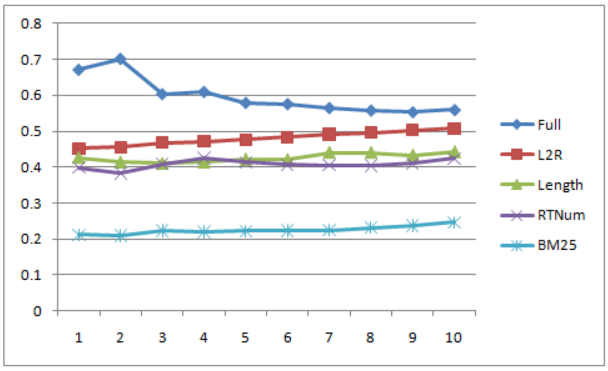

Figure 3: nDCG performance (y-axis) with different approaches. x-axis is the top k ranks. The Full model used unlabeled data with the regularization factor.

We can see that the proposed approach outperforms those unsupervised models remarkably, and it also performs better than the L2R model (Ranking SVM). Noticeably, the Length model is strong in performance, which shows the document length is a good indicator of quality. The RTNum model takes advantage of a Twitter specific property - a document of higher quality may be posted repeatedly by other users with higher probability. This is a special property for Twitter documents. Not surprisingly, the supervised methods outperform all unsupervised methods.

To further demonstrate that our approach outperforms the baselines, we list the results (in terms of $nDCG@k = 1, 5, 10$, and MAP) in Table 5 which clearly shows the advantages of our proposed approach. Note that our performance shown in Table 5 is significantly better than all the baselines (p-value<0.001 by t-test). We choose the significance level of 0.01 through the paper.

nDCG @k	Full	L2R	Length	RTNum	BM25
k=1	**0.671**	0.442	0.466	0.398	0.212
k=5	**0.579**	0.477	0.422	0.413	0.222
k=10	**0.559**	0.508	0.442	0.424	0.247
MAP	**0.427**	0.315	0.253	0.225	0.126

Table 5: Performance comparison between systems. Our results are significantly better than the baselines.

7.5 Feature Study

To investigate the influence of different features on performance, we perform a feature ablation study. As shown in Section 5, we classify features into different groups. In this experiment, we first train the basic model (as defined in Eq. 1) with all the features, and then remove one group of features each time. We also experiment with only content features to justify the effectiveness of these features.

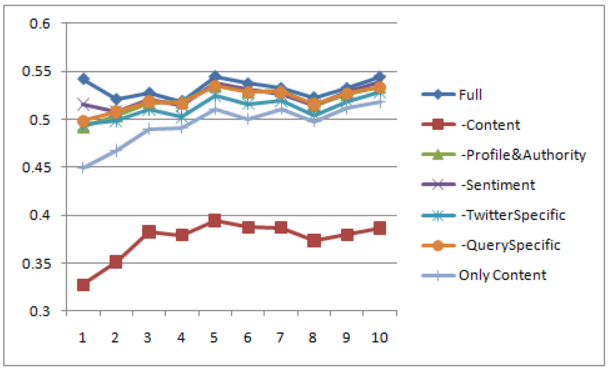

Figure 4: $nDCG@k$ performance with different feature groups. 'Full' means all the features. '-' means removing that feature group from the full feature set.

We can see that when removing content features, the performance drops substantially, which indicates that content is the most important indicator of quality. When using only content features, the performance is also fairly good (but significantly worse than the Full model, p-value$<$0.01 by t-test), showing that content is reliable features for this task. When removing Twitter specific features, there is a significant drop in performance (p-value$<$0.01). This indicates that such prior knowledge on tweets is helpful for ranking such documents. However, removing user profile and authority features does not affect the system much. Similar observation can be seen when removing sentiment features and query-specific features respectively. For query-specific features, it seems that such features play a light-weighted role. There may be two reasons for this: First, the documents are obtained from the BM25 model in our approach, thus all documents are more or less relevant to the query while our approach can be treated as a re-ranking process; Second, the document is very short, thus query-specific features may not be as important as in retrieving longer documents, more specifically, the query term frequency may not be as accurate as in longer documents.

To further investigate which specific content features are important, we conduct a further feature ablation study on content features. We find that the average term similarity (p-value$<$0.01), ratio of unique words (p-value=0.08), and ratio of POS tags (p-value$<$0.02) play more roles in performance. Not as expected, removing the length features does not lead to as a remarkable drop as removing other features (p-value=0.12). However, as shown in Figure 3, the Length model is strong in performance. This may infer that the length feature may be complemented by other content features.

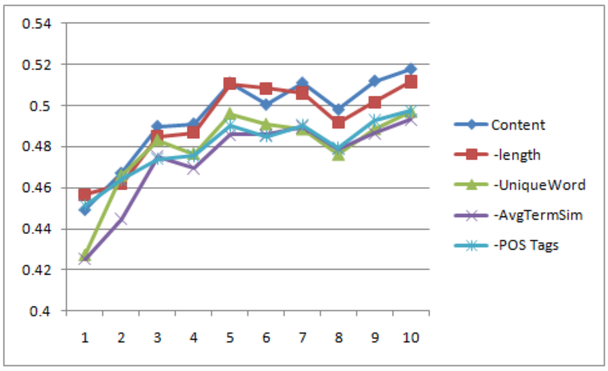

Figure 5: $nDCG@k$ performance with different content features. '-' means removing the feature group from the full content features.

Note that these experiments are performed with the basic model (Eq. 1). We also conduct similar feature studies with the regularization factor and similar observations are seen.

8 Conclusion

We presented a regression model which incorporates various features for suggesting quality-biased short-text documents. We proposed a content conformity hypothesis and formulated it into a regularization factor. The performance was boosted with such a factor. Moreover, unlabeled data can be used seamlessly in this approach, and leveraging such data leads to improvements in ranking performance. The comparative results demonstrate the effectiveness of finding high-quality tweets.

Short-text ranking is still in its infancy. There is still much work to do. For example, it is feasible to plug other hypotheses in this approach. As an instance, celebrity users may be more likely to post responsible tweets than common users. We also note that the quality of a tweet is not only determined by the text itself, but also by the external resources it points to (via a tiny URL) or it attaches (a picture or a video). Therefore, considering these factors would also be helpful in finding high-quality posts.

References

Marcelo Mendoza, Barbara Poblete, Carlos Castillo. 2010. *Twitter Under Crisis: Can we trust what we RT?*. 1st Workshop on Social Media Analytics (SOMA'10), July 25, 2010, Washington DC, USA.

Eugene Agichtein, Carlos Castillo, Debora Donato. 2008. *Finding High-Quality Content in Social Media.* WSDM'08, February 11-12, 2008, Palo Alto, California, USA. pp 183-193.

Jiang Bian, Yandong Liu, Ding Zhou, Eugene Agichtein, Hongyuan Zha. 2009. *Learning to Recognize Reliable Users and Content in Social Media with Coupled Mutual Reinforcement.* WWW 2009, April 20-24, 2009, Madrid, Spain.

Jiwoon Jeon, W. Bruce Croft, Joon Ho Lee, Soyeon Park. 2006. *A Framework to Predict the Quality of Answers with Non-Textual Features.* SIGIR'06, August 6-11, 2006, Seattle, Washington, USA

Chirag Shah, Jefferey Pomerantz. 2010. *Evaluating and Predicting Answer Quality in Community QA.* SIGIR'10, July 19-23, 2010, Geneva, Switzerland.

Takeshi Sakaki, Makoto Okazaki, Yutaka Matsuo. 2010. *Earthquake Shakes Twitter Users: Real-time Event Detection by Social Sensors.* WWW2010, April 26-30, 2010, Raleigh, North Carolina.

Haewoon Kwak, Changhyun Lee, Hosung Park, and Sue Moon. 2010. *What is Twitter, a Social Network or a News Media?* WWW 2010, April 26-30, 2010, Raleigh, North Carolina, USA.

Alan Ritter, Colin Cherry, Bill Dolan. *Unsupervised Modeling of Twitter Conversations.* 2010. Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the ACL, pages 172-180.

Sasa Petrovic, Miles Osborne, Victor Lavrenko. 2010. *Streaming First Story Detection with application to Twitter.* The 2010 Annual Conference of the North American Chapter of the ACL, pages 181-189, Los Angeles, California, June 2010.

Anlei Dong, Ruiqiang Zhang, Pranam Kolari, Jing Bai, Fernando Diaz, Yi Chang, Zhaohui Zheng, Hon-gyuan Zha. 2010. *Time is of the Essence: Improving Recency Ranking Using Twitter Data.* WWW 2010, April 26-30, 2010, Raleigh, North Carolina, USA.

Aron Culotta. 2010. *Towards detecting influenza epidemics by analyzing Twitter messages.* 1st Workshop on Social Media Analytics (SOMA'10), July 25, 2010, Washington, DC, USA.

Andranik Tumasjan, Timm O. Sprenger, Philipp G. Sandner, Isabell M. Welpe. 2010. *Predicting Elections with Twitter: What 140 Characters Reveal about Political Sentiment.* Association for the Advancement of Artificial Intelligence (www.aaai.org).

Yajuan Duan, Long Jiang, Tao Qin, Ming Zhou and Heung-Yeung Shum. 2010. *An Empirical Study on Learning to Rank of Tweets.* Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010), pages 295-303, Beijing, August 2010.

Xiaohua Liu, Kuan Li, Bo Han, Ming Zhou, Long Jiang, Zhongyang Xiong and Changning Huang. 2010. *Semantic Role Labeling for News Tweets.* Proceedings of 23rd International Conference on Computational Linguistics (Coling 2010)

Pear Analytics. 2009. *Twitter Study-August 2009.*

Anindya Ghose, Panagiotis G. Ipeirotis. 2010. *Estimating the Helpfulness and Economic Impact of Product Reviews: Mining Text and Reviewer Characteristics.* (January 24, 2010), Available at SSRN: http://ssrn.com/abstract=1261751.

Yue Lu, Panayiotis Tsaparas, Alexandros Ntoulas, Livia Polanyi. 2010. *Exploiting social context for review quality prediction.* WWW 2010, April 26-30, 2010, Raleigh, North Carolina, USA.

Yang Liu, Xiangji Huang, Aijun An, Xiaohui Yu. 2008. *Modeling and Predicting the Helpfulness of Online Reviews.* 2008 Eighth IEEE International Conference on Data Mining. pp. 443-452.

Soo-Min Kim, Patrick Pantel, Tim Chklovski, Marco Pennacchiotti. 2006. *Automatically Assessing Review Helpfulness.* Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing (EMNLP 2006), pp. 423-430, Sydney, July 2006.

Nitin Jindal, Bing Liu. 2008. *Opinion Spam and Analysis.* WSDM'08, February 11-12, 2008, Palo Alto, California, USA.

Ee-Peng Lim, Viet-An Nguyen, Nitin Jindal, Bing Liu, Hady W. Lauw. 2010. *Detecting Product Review Spammers using Rating Behaviors.* CIKM'10, October 26-30, 2010, Toronto, Ontario, Canada.

William J. Corvey, Sarah Vieweg, Travis Rood, Martha Palmer. 2010. *Twitter in Mass Emergency: What NLP Techniques Can Contribute.* Proceedings of the NAACL HLT 2010 Workshop on Computational Linguistics in a World of Social Media, pages 23-24, Los Angeles, California, June 2010.

Sasa Petrovic, Miles Osborne, Victor Lavrenko. 2010. *The Edinburgh Twitter Corpus.* Proceedings of the NAACL HLT 2010 Workshop on Computational Linguistics in a World of Social Media, pages 25-26, Los Angeles, California, June 2010

Kalervo Järvelin and Jaana Kekäläinen. 1998. *Ir evaluation methods for retrieving highly relevant documents.* In SIGIR 2000: Proceedings of the 23th annual international ACM SIGIR conference on Research and development in information retrieval, pages 41-48, 2000.

Xiaojin Zhu, Andrew B. Goldberg, Ronald Brachman, Thomas Dietterich. 2009. *Introduction to Semi-Supervised Learning.* Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool Publishers, 2009.

Jaime Teevan, Daniel Ramage, Meredith Ringel Morris. 2009. *#TwitterSearch: A Comparison of Microblog Search and Web Search.* WSDM11, February 9C12, 2011, Hong Kong, China.