Structuring Tweets for Improving Twitter Search

Zhunchen Luo and Yang Yu
China Defense Science and Technology Information Center, No. 26, Fucheng Road, 100142, Beijing, China.
E-mail: {zhunchenluo, 7yu6yang}@gmail.com

Miles Osborne
School of Informatics, University of Edinburgh, EH8 9AB, Edinburgh, UK. E-mail: miles@inf.ed.ac.uk

Ting Wang
College of Computer, National University of Defense Technology, 410073, Changsha, Hunan, China.
E-mail: tingwang@nudt.edu.cn

Spam and wildly varying documents make searching in Twitter challenging. Most Twitter search systems generally treat a Tweet as a plain text when modeling relevance. However, a series of conventions allows users to Tweet in structural ways using a combination of different blocks of texts. These blocks include plain texts, hashtags, links, mentions, etc. Each block encodes a variety of communicative intent and the sequence of these blocks captures changing discourse. Previous work shows that exploiting the structural information can improve the structured documents (e.g., web pages) retrieval. In this study we utilize the structure of Tweets, induced by these blocks, for Twitter retrieval and Twitter opinion retrieval. For Twitter retrieval, a set of features, derived from the blocks of text and their combinations, is used into a learning-to-rank scenario. We show that structuring Tweets can achieve state-of-the-art performance. Our approach does not rely on social media features, but when we do add this additional information, performance improves significantly. For Twitter opinion retrieval, we explore the question of whether structural information derived from the body of Tweets and opinionatedness ratings of Tweets can improve performance. Experimental results show that retrieval using a novel unsupervised opinionatedness feature based on structuring Tweets achieves comparable performance with a supervised method using manually tagged Tweets. Topic-related specific structured Tweet sets are shown to help with query-dependent opinion retrieval.

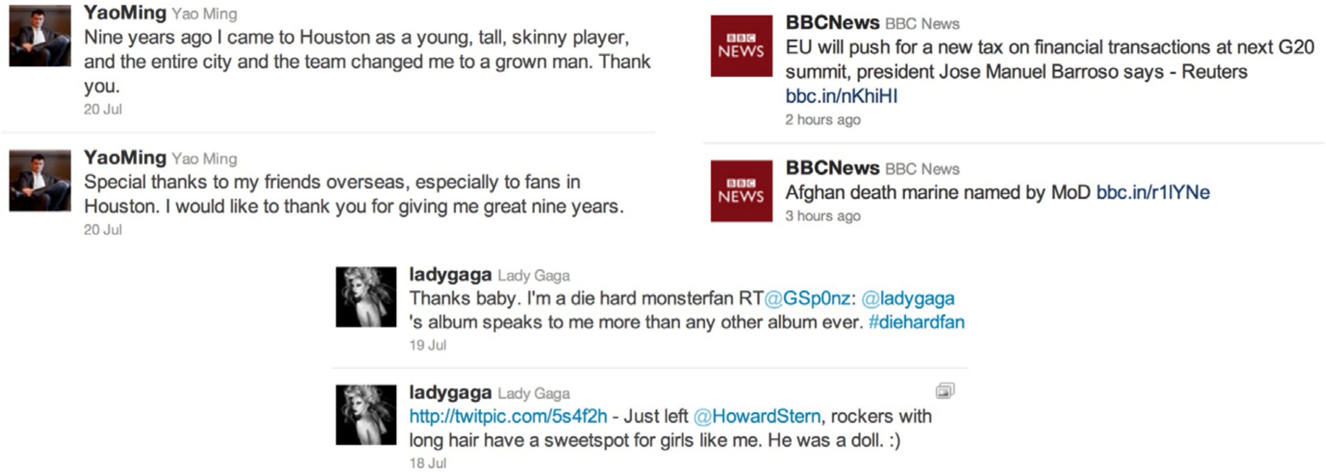
Introduction

With the widespread adoption of Twitter there is a need to search through that information and find relevant documents. Performing a search in Twitter is very different from the traditional setting (Teevan, Ramage, & Morris, 2011). The two main challenges include a much larger volume of data and increased noise, while the benefits come in the form of social metadata that normally not present in traditional media. However, existing Twitter search systems simply treat the text of a Tweet as a unit of plain text when modeling relevance (Duan et al., 2010; Efron, 2010; Gottron, Kunegis, & Alhadi, 2011; Massoudi, Tsagkias, de Rijke, & Weerkamp, 2011; Naveed, Metzler & Cai, 2011). Previous work shows that web pages and normal text documents can be subdivided into nonoverlapping structural blocks based on their contents or functions. These blocks and their combinations can be used to improve the representation of documents in an information retrieval task (Ahnizeret et al., 2004; Callan, 1994; Fernandes et al., 2007). Although a Tweet is a short text, it can be seen as a structural document constructed from blocks.

Figure 1 shows some Tweets by Yao Ming, BBC News, and Lady Gaga 1. We can see a lot of variation of style between the three—Yao Ming uses only a plain text, BBC News often ends their Tweets with a link to the story, and Lady Gaga uses a mixture of hashtags, links, and mentions. Regardless of the length of plain texts, hashtags, links, mentions, and so on, these can be seen as blocks for Tweets. In this paper we use these blocks to induce structure in Tweets for the purpose of improving ad hoc retrieval performance. This is based on the idea that the occurrence of a term in different blocks imposes different importance factors in the ranking process, as each block has its own specific information about the topic, function, length, position, textual quality, and context in a Tweet. Moreover, the sequence of these blocks for every Tweet also encodes changing discourse and even reflects the quality of the document.

Received August 30, 2013; revised April 30, 2014; accepted May 1, 2014

© 2015 ASIS&T • Published online in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/asi.23332



YaoMing Yao Ming
Nine years ago I came to Houston as a young, tall, skinny player, and the entire city and the team changed me to a grown man. Thank you.
20 Jul

YaoMing Yao Ming
Special thanks to my friends overseas, especially to fans in Houston. I would like to thank you for giving me great nine years.
20 Jul

BBCNews BBC News
EU will push for a new tax on financial transactions at next G20 summit, president Jose Manuel Barroso says - Reuters bbc.in/nKhiHI
2 hours ago

BBCNews BBC News
Afghan death marine named by MoD bbc.in/r1lYNe
3 hours ago

ladygaga Lady Gaga
Thanks baby. I'm a die hard monsterfan RT@GSp0nz: @ladygaga 's album speaks to me more than any other album ever. #diehardfan
19 Jul

ladygaga Lady Gaga
http://twitpic.com/5s4f2h - Just left @HowardStern, rockers with long hair have a sweetspot for girls like me. He was a doll. :)
18 Jul

FIG. 1. Examples of Tweets by Yao Ming, BBC News, and Lady Gaga. Yao Ming is a retired Chinese professional basketball player in the NBA. BBC News is a web news gathering and broadcast of news and current affairs. Lady Gaga is an American pop singer. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

We call the individual blocks *Twitter Building Blocks* (TBBs). Combinations of block sequence (TBB structures) capture the structural information of Tweets. These structures can be used to cluster Tweets and each cluster has its own informational characteristic. For example, Tweets with the same structure as BBC News Tweets in Figure 1 are likely to be broadcast news. Moreover, the structures are related to the textual quality of Tweets. Here we present the study of using structural information to improve Twitter retrieval and Twitter opinion retrieval.

In the Twitter retrieval task, we use structural information into a learning-to-rank scenario. We develop a set of features that are derived from TBBs and structures. These features are expected to improve the performance of Tweets retrieval. The advantages of these new features are that they are not only related to structural information of the Tweets but also can be derived from the Tweet text itself directly without relying on other social media features. We compare the performance of the learning-to-rank model using these new features to a state-of-the-art method (Duan et al., 2010). The results show that these features can achieve comparable performance when used alone, and higher performance when used jointly with other social media features.

Unlike Twitter retrieval, the task of opinion retrieval in Twitter is finding Tweets related to a query and also contain opinions about it (query). Estimating the opinionatedness score of a Tweet is essential for this task. Here we use corpus-derived method to estimate the opinionatedness value of the Tweet as a feature for ranking. However, collecting manually tagged Tweets is time-consuming. Estimating the opinionatedness of a Tweet is also a topic-dependent problem (Jiang et al., 2011). It is impossible to collect topic-related manually tagged Tweets for every topic. Therefore, we propose a novel approach, using the structural information and social information of the Tweets, to automatically generate a large number of accurate "pseudo" subjective Tweets (PSTs) and "pseudo" objective Tweets (POTs). These two Tweet sets can be used as a corpus to derive lexicons for estimating the opinionatedness of a new Tweet. We show that our approach can achieve comparable performance with a method that uses a manually tagged Tweets corpus. Our approach can also generate topic-related PSTs and POTs to a given query, which can help query-dependent opinion retrieval.

The contributions of this paper can be summarized as follows:

1. We propose TBBs, which capture a sequence of tokens that encode a variety of communicative intent, and sequence of these TBBs (TBB structures) captures changing discourse.
2. We show that our structuring of Tweets yields results Twitter retrieval that are very similar to a state-of-the-art system that uses social media features. Especially, we do not need to use those social media features.
3. We also show that the ranking model with an opinionatedness feature, using our automatic generation of PSTs and POTs based on specific constructed Tweet sets, can achieve comparable performance with a method using manually tagged Tweets. Using this method in a query-dependent scenario yields further gain.

Related Work

Our tasks are improving Twitter retrieval and opinion retrieval by exploiting structural information of Tweets. In this section we first examine prior literature on ad-hoc information retrieval in Twitter, then we introduce some opinion retrieval work in web search, since there were no opinion retrieval in Twitter work before. Finally, we review previous structured documents retrieval work that could help motivate our structuring Tweets idea.

Twitter Retrieval

Twitter search introduces new problems for information retrieval systems to tackle. First, the text of a Tweet is very short, which may make document weighting models such as BM25 less effective. Because the term frequency is a good indicator for the importance of documents in a web search setting, but each term is likely to only appear once in a Tweet. Furthermore, the shorter Tweet length may make vocabulary mismatch between the query and the relevant Tweets more acute, reducing the recall of the Tweet rankings produced with standard document weighting models (McCreadie & Macdonald, 2013). Massoudi et al. (2011) studied a new retrieval model for Twitter search by considering the model with textual quality and Twitter-specific quality indicators. They found that this model had a significant positive impact on Tweets retrieval. Naveed et al. (2011) combined document length normalization in a retrieval model to resolve the sparsity of short texts for Tweets. Ferguson et al. (2012) examined the applicability of term frequency statistics and document length normalization to microblog retrieval. They found document length normalization always harmed performance, and the benefit from incorporating term frequency statistics was minor.

Recently, several Twitter search methods have been proposed, which were based on the meta information. Choi, Croft, and Kim (2012) suggested a low-cost quality model using surrogate judgments based on user behavior (i.e., reTweets) that can be collected automatically. They believe that the reTweet probability of a Tweet may co-relate to the relevance of the Tweet. This is because reTweet probability of a Tweet determines if the Tweet is needed to be broadcast to the user's followers, while relevance determines if the Tweet is informative to the users. These are orthogonal issues (Ravikumar, Talamadupula, Balakrishnan, & Kambhampati, 2013). There were also multiple approaches (Cha, Haddadi, Benevenuto, & Gummadi, 2010; Luo, Osborne, Petrovic, & Wang, 2012; Luo, Osborne, Tang, & Wang, 2013; Luo, Tang, & Wang, 2013) that try to rank Tweets based on specific features of the user who Tweeted the Tweet. Researchers and social observers both believe that hashtags, as a new type of organizational objects of information, played a dual role in online microblogging communities (Yang, Sun, Zhang, & Mei, 2012). Efron (2010) proposed a language modeling approach for hashtag retrieval. He used the retrieved hashtags on a topic of interest for query expansion to improve the performance of Twitter search. Berendsen, Tsagkias, Weerkamp, and de Rijke (2013) gave a starting point: Tweets with a hashtag are relevant to the topic covered by the hashtag and hence to a suitable query derived from the hashtag. From this they described a method for creating pseudo test collections for microblog search in an unsupervised way. Duan et al. (2010) indicated links shared in Tweets, which provide more detailed information beyond the Tweet's 140 characters, may be relevant to the query at a high probability. They took the feature whether a Tweet contains a link into a ranking model. Luo, Osborne, Petrovic, et al. (2012) investigated the Twitter search performance about different positions of links in a Tweet. McCreadie and Macdonald (2013) proposed three approaches for incorporating the content of hyperlinked documents when ranking Tweets. All of them found containing links was the most effective feature for Twitter retrieval. Mentions, that is, user, names prefixed with the "@" symbol are used to indicate replies or direct messages to the user in question. Luo, Osborne, and Wang (2012) found the text of a Tweet with mentions was more likely to be "personal content." The quality of these Tweets might be low, which affects the Tweets ranking performance. Duan et al. (2010) and Han et al. (2013) investigated effectiveness of the feature whether a Tweet contains mentions. Besides, the unique characteristics of Twitter, such as reTweets, links, hashtags, and mentions. There are other aspects that have potential influence on the relevance of Tweets. For example, the temporal factor, indicating whether the Tweet is timely to a given query topic, is important in real-time Twitter search (Efron, 2011). However, using traditional Twitter search approaches, it is rather difficult to properly integrate the variety of features into the retrieval model.

The relatedness of a Tweet to a query depends on many factors. By combining various sources of evidence of relevance, learning-to-rank has been widely applied in Twitter search. Learning-to-rank, a family of learning approaches, aims at learning the ranking model automatically from a training data set and it has been broadly used in information retrieval and machine learning (Cheng et al., 2013). Duan et al. (2010) considered learning-to-rank for Tweets. They proposed a new ranking strategy which used not only the content relevance of a Tweet, but also the account authority and Tweet-specific features. We take their approach as our baseline for comparison. Miyanishi et al. (2011) applied an unspecified learning-to-rank method by clustering the Tweets retrieved for given topics. There were other studies that resolve Twitter retrieval problem based on machine learning technology (Berendsen et al., 2013; Zhang, He, & Luo, 2012; Zhang et al., 2012). The above approaches do not consider any information about the structure of Tweets.

TREC 2011 and 2012 introduced the Microblog Track that addressed a single pilot task, entitled *real-time search task*, where the user wished to see the most recent but relevant information to the query (Ounis, Lin, & Soboroff, 2011). Many groups participated in the track from across the world. The experimental results indicated the large gap between the best and median evaluation score (e.g., MAP value) per-topic. It shows that Tweets retrieval is far from being a solved problem. Amati et al. (2011) proposed a new DFRee-KLIM retrieval model from the divergence from randomness framework that accounts for the very short nature of Tweets. Li et al. (2011) applied the Word Activation Force algorithm and Term Similarity Metric algorithms to mine the connection between the expansion terms and the given topic. The most effective approaches submitted to the TREC 2011 Microblog Track focused purely on relevance (Ounis et al. 2011). Metzler and Cai (2011) used learning-to-rank with pseudo-relevance feedback to find the 30 most relevant Tweets. Within the context of the TREC 2011 and 2012 Microblog Tracks, McCreadie and Macdonald (2013) thoroughly evaluated to what extent hyperlinked documents can aid Tweets retrieval effectiveness. We use the TREC Microblog Track 2011 and 2012 test collections (including Tweets2011) to evaluate our proposed approaches for integrating structural information into the Tweets ranking process.

Opinion Retrieval

Retrieval for blogs was first introduced in TREC 2006 (Ounis et al., 2006) and continued in TREC 2007 and 2008 (Macdonald, Ounis, & Soboroff, 2007; Ounis, Macdonald, & Soboroff, 2008). Most groups participate in TREC adopted a two-stage approach, where an initial set of relevant but not necessarily opinionated documents are re-ranked by taking into account various document opinion features.

There was other work related to opinion retrieval. Eguchi and Lavrenko (2006) first introduced an opinion ranking formula that combines sentiment relevance models and topic relevance models into a generation model. This formula was shown to be effective on the MPQA corpus, but it does not perform very well in the following TREC opinion retrieval experiment. Zhang and Ye (2008) and Huang and Croft (2009) also put forward their own way to unify sentiment relevance models and topic relevance models for ranking. Gerani, Carman, and Crestani (2009) first investigated learning-to-rank for blog posts. All this work is in the context of blogs or web documents, Twitter, however, is a novel domain and specific structural information of Tweets and rich social environment should be considered when modeling relevance.

In opinion retrieval, estimating the opinionatedness of a document is essential. He, Macdonald, He, and Ounis (2008) proposed an approach to calculate the opinionatedness of a document based on subjective terms. These terms are automatically derived from manually tagged data. They used all opinionated relevant documents to queries as a subjective document set and other topic-relevant documents as an objective document set. The opinionated score of each term can measured by the divergence of the distribution in these two sets. Amati et al. (2008) adopted a similar approach for the automatic construction of an opinion-term vocabulary for ad-hoc retrieval. Seki and Uehara (2009) used a statistical language model, incorporating distant word dependencies, to model the opinionated document for ranking. Jijkoun, de Rijke, and Weerkamp (2010) presented a method for automatically generating topic-specific subjective lexicons based on extracting syntactic clues of manually tagged data. Li, Zhou, Feng, and Wong (2010) proposed a novel notion of topic-sentiment word pairs as a new representation for opinion retrieval task. While all of the above approaches are effective for opinion retrieval, they need human tagged subjective and objective documents.

Unlike the work introduced above, Zhang et al. (2007) used the reviews of RateitAll.com and other webpages as a source of "pseudo" subjective sentences (PSSs), and Wikipedia documents as an external source of "pseudo" objective sentences (POSs). They assume that the subjective portion should be dominant in the reviews so that the effect of the objective portion can be neglected. The situation is opposite when using Wikipedia documents. They then used these PSSs and POSs to build an SVM sentence classifier. This classifier can give the sentence an opinionated score that is combined with the topic relevance score for ranking. We propose an approach based on a similar idea in the context of Twitter. We use the structural information and social information of Tweets to automatically generate "pseudo" subjective Tweets (PSTs) and "pseudo" objective Tweets (POTs), for opinion retrieval in Twitter.

Structured Documents Retrieval

Previous information retrieval work found that web pages and normal text documents could be subdivided into nonoverlapping structural blocks based on their contents or functions. These blocks and their combinations can be used to improve the representation of documents (structured documents). Structured documents retrieval attempts to exploit this structural information to an information retrieval task. This is based on the idea that the same term in different blocks has its own importance factor for ranking and that certain structural combinations of blocks have specific informational characteristics. Although a Tweet is a short text, it also can be seen as a structured document.

Before, Ahnizeret et al. (2004) used manual assignment of block weights to improve the quality of search results, which can be used to derive effective block-based term weighting methods. They also showed that such a structure was useful for data-intensive web sites, which are subjected to frequent content updates, for example, digital libraries, web forums, news web sites, and so on. Fernandes et al. (2007) and de Moura et al. (2010) used the block structure of a web page to improve ranking results. They proposed approaches based on automatically computed block-weight factors. Cai et al. (2004) proposed a method for taking advantage of the segmentation of web pages into blocks for search task. All of these studies assume a same term can behave differently in particular blocks.

Our approach is improving twitter retrieval by exploiting structural information of Tweets. The prior literature on Tweet parsing motivate our work. Foster et al. (2011) examined the consequences of applying an off-the-shelf WSJ-trained POS-tagging and dependency-parsing model to the language of Twitter. They observed a drastic drop in performance moving from their in-domain WSJ test set to the new Twitter data set. The reason is the propagation of part-of-speech tagging errors. Therefore, we use Gimpel et al. (2011)'s part-of-speech Tweet tagger to reduce the tagging errors for Tweets.

Twitter Building Block

Previous work shows that some long text documents (e.g., web pages and articles) can be subdivided into non-overlapping structural blocks based on their contents or functions. These documents can be seen as structured documents using new representation by the blocks and their combinations. Exploiting the structural information of texts, many search systems improved their retrieval performance (Ahnizeret et al., 2004; Callan, 1994; Fernandes et al., 2007).

Most of the existing Twitter retrieval approaches simply treat the text of a Tweet as a unit of plain text when modeling relevance. Although a Tweet is a short text, it also can be viewed as a combination of text blocks with each block itself consisting of a sequence of tokens. We call each of these text blocks TBBs. Various combination of these TBBs give different Tweet structures (TBB structures). These structures can be used to cluster Tweets and each cluster has its own informational characteristic. This structural information might help Twitter search.

TBB Definition

In Twitter, three "special" actions have emerged that users regularly use in their Tweets: tagging (adding tags to a Tweet to indicate the topic of content), reTweeting (reposting someone else's Tweet), and mentioning (directly mentioning a user). We further divide the content of Tweets into three classes: the sharing of information through links, comments, and normal message. We therefore propose six types of building blocks:

TAG: Combination of hashtags (#) and keywords (e.g., #iphone) indicating the topic of a Tweet.
MET: To indicate another user(s) (e.g., @ladygaga) as the recipient(s) of a Tweet.
RWT: To indicate copying and rebroadcasting of the original Tweet (e.g., *RT* @ladygaga).
URL: Links to outside content (e.g., http://www.facebook.com).
COM: Comments, used to describe people's sentiment, appraisals, or feelings toward another TBB in the same Tweet.
MSG: Message content of a Tweet.

Figure 2 shows two Tweets that illustrate these TBBs. Every underlined sequence of tokens shows a TBB. In Figure 2 we can see that Tweet (a) has a sequence of TBBs of "COM RWT MET MSG" and Tweet (b) has a sequence of TBBs of "MSG URL TAG." The form and order of TBBs encode changing discourse of Tweets. Tweet (a) means the author reTweeted (RWT) @miiisha_x's message (MSG) which is mentioned to @XPerkins (MET) and at the same time gave his comment (COM) about the message (MSG). In Tweet (b) the author gave a message (MSG), a link (URL), and two hashtags (TAG). In the last two blocks, the author provided additional resource and labeled the topic of the Tweet that can help readers to better understand the original message (MSG).

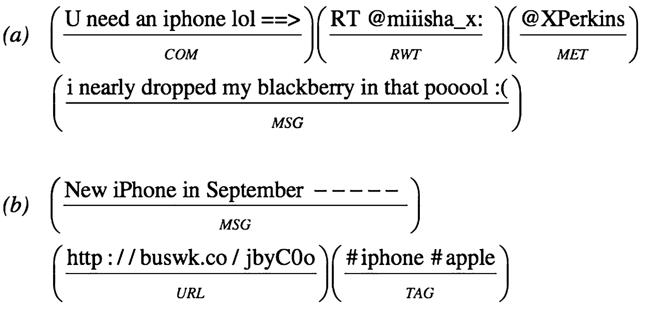


FIG. 2. Tweets with gold TBB annotation.

TABLE 1. Distribution of TBB structures.

TBB structure	Per.(%)	TBB structure	Per.(%)
MSG	30.25	TAG MSG	1.55
MET MSG	20.70	TAG MSG URL	1.20
MSG URL	18.40	RWT MSG URL	0.95
OTHERS	13.20	COM RWT MSG	0.85
COM URL	4.10	MET MSG URL	0.85
MSG TAG	2.65	MSG MET MSG	0.70
MSG URL TAG	2.10	RWT MSG TAG	0.70
RWT MSG	1.75		

To understand how people use these building blocks, we randomly collected 2,000 samples of English[1] Tweets, automatically tokenized them using a tokenizer from O'Connor et al. (2010),[2] and then manually tagged their TBBs and structures. Table 1 shows the distribution of different TBB structures in these Tweets. The 14 most frequently occurring TBB structures are listed. All other TBB structures are grouped into "OTHER." We can see that the "MSG," the simplest structure, has the highest percentage. Other high frequency structures are also the simple structures containing no more than three TBBs. The percentage of "OTHERS" structure is only 13.2%. All these suggest that people usually use some simple and fixed structures to Tweet.

Automatic TBB Tagger

Manual annotation of TBBs for every Tweet is clearly infeasible. We develop an automatic tagger for this task. The task can be seen as two subtasks: TBB type classification and TBB boundary detection, which makes the task very similar to Named Entity Recognition. We thus adopt a sequential labeling approach to jointly resolve these two subtasks and use an IOB-type labeling scheme.

Given a Tweet as input, the expected output is a sequence of blocks $B1B2...Bm$. Every Bi is a sequence of consecutive tokens $ti1ti2...tin$. Each token tij in a Tweet is assigned only one label *"X_Y" (X = TAG, MET, RWT, URL, COM, MSG; Y = B, I)* to indicate its type and boundary. Every token tij in block Bi has the same X value. *"Y = B"* only labels the tokens tij $(j = 1)$ and *"Y = I"* labels other tokens tij $(j > 1)$.

TABLE 2. Automatic TBB tagger result using CRF approach.

Label	Num.	Pre.(%)	Rec.(%)	F1 (%)
TAG_B	72	88.00	91.67	89.80
TAG_I	34	93.94	91.18	92.54
URL_B	164	95.62	93.29	94.44
URL_I	24	55.56	41.67	47.62
MET_B	145	91.45	95.86	93.60
MET_I	63	94.34	79.37	86.21
RWT_B	72	93.06	93.06	93.06
RWT_I	129	90.51	96.12	93.23
COM_B	70	67.27	52.86	59.20
COM_I	550	64.48	46.55	54.07
MSG_B	482	90.50	90.87	90.68
MSG_I	5708	94.27	97.06	95.64
AVG		84.92	80.79	82.80

For example, the labels of tokens “iPhone” and “#iphone” in the Tweet (b) of Figure 2 are “MSG_I” and “TAG_B.”

We use a *Conditional Random Field* for tagging (Lafferty, McCallum, & Pereira, 2001), enabling the incorporation of arbitrary local features in a log-linear model. Our features include:

Token Type: A text window of size 7 with the current token in the middle. We choose the window of size 7 based on other sequence labeling tasks in Twitter (Gimpel et al., 2011; Liu et al., 2011; Ritter et al., 2011; Owoputi et al., 2013).[3]
Pos: Part-of-speech for every token.[4]
Length: Number of characters in the token.
Pre_Suf_fix: Prefix features and suffix features of characters up to length 3.
Twitter orthography: Several regular rules can be used to detect tokens in different types of TBB:
Every token in TAG that begins with “#.”
Every token begins with “www.,” “http:” or ends with .“com” is a URL tag.
The sequence of tokens, which its pattern is “@usename:” or “@usename,” are MET tags.
The sequence of tokens, of the form “RT @usename:,” “RT @usename,” “RT” or “via @usename,” are RWT tags.
The preceding of “RT @username” and the succeeding of “via @username” or “<<” is a COM tag.

TABLE 3. The heuristics for tagging TBB in tweets.

TAG_B	the token begins with “#” and the preceding token does not begin with “#”
TAG_I	the token begins with “#” and the preceding token begins with “#”
URL_B	the token begins with “www.”, “http:” or ends with “.com” and the preceding token does not begin with “www.”, “http:” or end with “.com”
URL_I	the token begins with “www.”, “http:” or ends with “.com” and the preceding token begins with “www.”, “http:” or ends with “.com”
MET_B	the token begins with “@” and the preceding token does not begin with “@” or “RT”
MET_I	the token begins with “@” and the preceding token begins with “@”
RWT_B	the token begins with “RT” or “via”
RWT_I	the token begins with “@” and the preceding token begins with “RT” or “via”
COM_B	the preceding sequence of tokens begins with “via @”
MSG_B	the first token which does not begin with “#”, “RT”, “www.” or “http:”
MSG_I	the other tokens

TABLE 4. Automatic TBB tagger result using heuristic-based approach.

Label	Num.	Pre.(%)	Rec.(%)	F1 (%)
TAG_B	72	66.67	74.07	70.18
TAG_I	34	72.73	53.33	61.54
URL_B	164	100.00	93.94	96.88
URL_I	24	25.00	11.11	15.38
MET_B	145	63.16	17.39	27.27
MET_I	63	100.00	90.91	95.24
RWT_B	72	90.63	90.63	90.63
RWT_I	129	100.00	45.90	62.92
COM_B	70	0.00	0.00	0.00
COM_I	550	0.00	0.00	0.00
MSG_B	482	82.22	55.78	66.47
MSG_I	5708	86.91	99.50	92.78
AVG		65.61	52.71	56.61

We manually tagged 2,000 Tweets for training and testing. We randomly divided the data into a training set 1,000 Tweets, a development set of 500 Tweets, and a test set of 500 Tweets. The FlexCRFs toolkit[5] was used to train a linear model. Table 2 shows the performance of our automatic TBB tagger that achieves an average F1 score of 82.80%. The tags “COM_B” and “COM_I” have relative low F1 values. The reason is that the COM tag is infrequently labeled by human and opinion mining is always a challenging task in NLP. The tag “URL_I” also has low F1 value. The reason is that some of links has been wrongly tokenized by Twitter tokenizer (O’Connor et al., 2010). However, the effect is insignificant because the number of “URL_I” tag is small. From these labeled tokens, the boundaries of TBBs and the structure of a Tweet are identified. TBB structure identification can achieve an accuracy of 82.60%.

Additionally, we use a purely heuristic-based approach based on Twitter orthography features set to tag the Tweets. We wonder to what extent the trained model would outperform a purely heuristic-based approach. The details of heuristics are listed in Table 3.

Table 4 shows the result of TBB tagging using heuristic-based approach. We can see that just using some simple heuristics, it is easily to tag the “URL_B,” “MET_I,” “RWT_B,” and “MSG_I,” but the TBB tagger model based on a machine-learning approach significantly outperforms purely heuristic-based approach.

TBB Analysis

We take a look at the characteristics of different TBB structures and textual quality in each one.

TABLE 5. Statistics of TBB structures clusters.

ID	Characteristic	TBB structures	Proportion(%)
Cluster_1	Public Broadcast	"MSG URL" or "MSG URL TAG" or "TAG MSG URL"	85
Cluster_2	Private Broadcast	"COM URL" or "MET MSG URL"	75
Cluster_3	High Quality News	"RWT MSG URL"	78
Cluster_4	Messy	"OTHERS"	67

Characteristic of TBB. It is possible to usefully cluster Tweets by TBB structures and these clusters have similar informational characteristics:

- Public Broadcast: Tweets produced by BBC News (for example) conventionally have these forms: "MSG URL," "MSG URL TAG," and "TAG MSG URL." These Tweets usually contain an introductory text followed by a corresponding link.
- Private Broadcast: Tweets posted by ordinary users who have a small number of followers are typically of the form "COM URL" and "MET MSG URL." E.g, the structure of a Tweet "I like it and the soundtrack http://www.imdb.com/title/tt1414382/" is "COM URL." The number of the people who care about these kinds of Tweets is much smaller than public broadcast Tweets.
- High Quality News: In the case of Tweets containing high quality news, the most common form is "RWT MSG URL." E.g, a Tweet "RT @CBCNews Tony Curtis dies at 85 http://bit.ly/dlSUzP" is not only simple news, but also a hot topic.
- Messy: Tweets containing complex structures are of the form "OTHERS." An example is the Tweet "RT @preciousjwl8: Forreal doeee? (Wanda voic) #Icant cut it out #Newark http://twipic.com/2u15xa. . .lmao!!WOW . . . http://tmi.me/1UwsA." It is not easily readable, as the discourse changes frequently.

We ask a researcher (nonauthor) to spot-check the proportion of Tweets in clusters assigned particular TBBs that have certain informational characteristics. The number of Tweets in each cluster is 100. Table 5 gives the statistics analysis of TBB structures clusters. We can see that in Cluster_1, which the TBB structures of Tweets are "MSG URL" or "MSG URL TAG" or "TAG MSG URL," has 85% "Public Broadcast" Tweets. Cluster_2, Cluster_3 and Cluster_4 have 75%, 78%, 67% "Private Broadcast," "High Quality News," "Messy" Tweets, respectively. All these show it is possible to cluster Tweets by TBB structures and each cluster has certain informational characteristics. The proportion of Tweets that are "Messy" is not very large in Cluster_4. The reason is that most of "OTHERS" Tweets only have four or five blocks and the discourse changes infrequently. For example, the Tweet "A great refreshing #holiday with #beach #tour in #greece !!! http://bit.ly/crw9xn" does not have discourse changing.

Textual Quality of TBB. Twitter provides a large volume of data in real time. The textual quality of Tweets, however, varies significantly ranging from news-like text to meaningless strings (Han & Baldwin, 2011). Previous work shows that considering textual quality of Tweets in a retrieval model can help Twitter search (Duan et al., 2010; Massoudi et al., 2011; Naveed et al., 2011). For this reason, we consider the relation between the structure of a Tweet and its textual quality.

TABLE 6. OOV values about TBB structures.

TBB structure	O.(%)	TBB structure	O.(%)
OTHERS	4.30	MET MSG URL	1.42
TAG MSG URL	3.42	MSG	1.32
MSG URL	1.93	MSG TAG	1.31
MSG URL TAG	1.91	RWT MSG URL	1.30
COM URL	1.80	MET MSG	1.15
COM RWT MSG	1.78	RWT MSG	0.82
MSG MET MSG	1.64	RWT MSG TAG	0.58
TAG MSG	1.63		

We randomly collected 10,000 English Tweets for each TBB structure through Automatic TBB Tagger labeling and calculated their *Out of Vocabulary* (OOV) value. The OOV value is the number of words out of vocabulary divided by the total number words.[6] This is used to roughly approximate the language quality of text (Agichtein et al., 2008). In order to adapt the characteristics of the language in Twitter, we collected 0.5 million most frequent words from 1 million English Tweets as the vocabulary. Most of the words out of vocabulary in Tweets are misspelt words or abbreviations. Table 6 shows that different TBB structures have very different OOV values. This suggests that textual quality associated with TBBs structures is different. The structures of "RWT MSG TAG" and "RWT MSG" have the lowest value of OOV. It suggests people usually reTweet other users' high-quality text. "OTHERS" has the highest OOV value. This is because each Tweet has to follow the 140-characters limitation, whereas most of the Tweets associated with "OTHERS" contain more blocks about "TAG," "MET," "RWT," and "URL." As a result, the user quite often introduces abbreviations in order to compress the length of "MSG" and "COM" blocks.

TBB for Twitter Retrieval

The above-proposed TBB are evaluated when retrieving Tweets. This particular chosen application evaluates the existence of useful features and information in TBB.

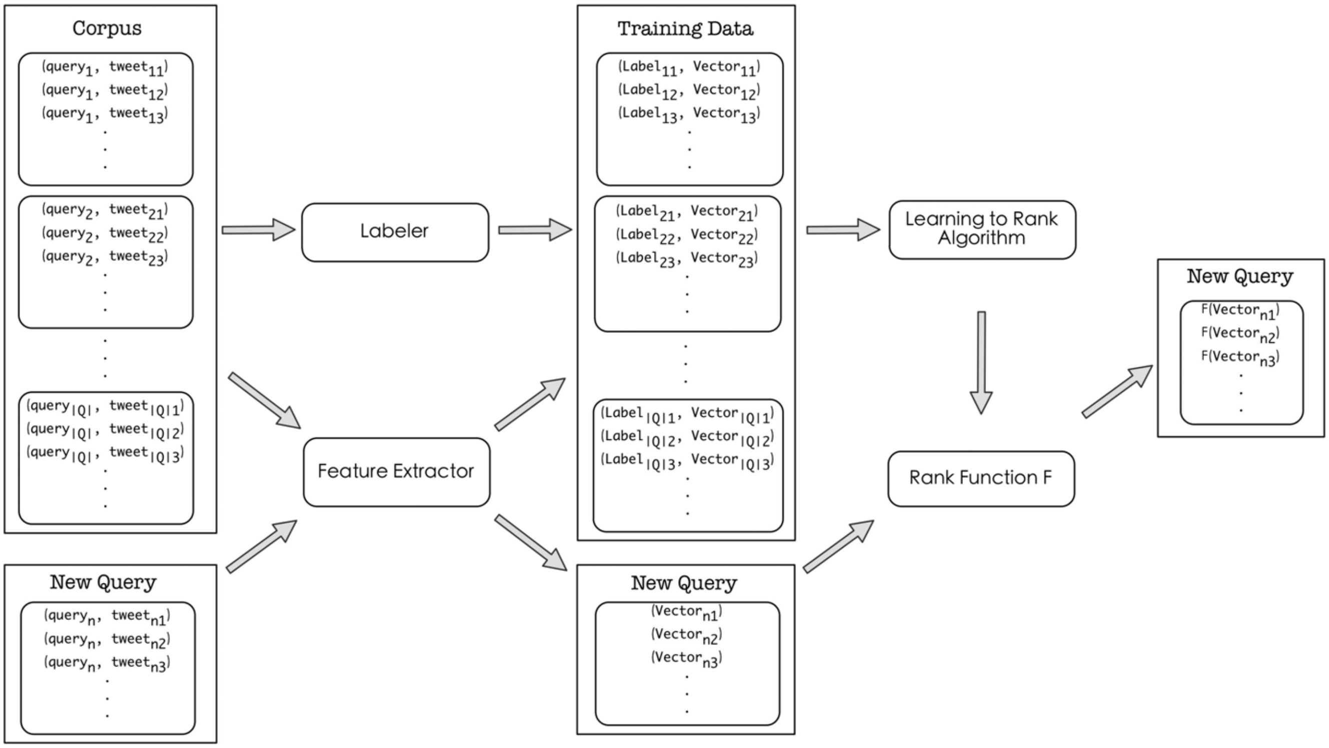


FIG. 3. Framework of learning-to-rank Tweets.

Learning-to-Rank Framework

Learning-to-rank is a data-driven approach that effectively incorporates a bag of features in a model for ranking task (Luo, Osborne, & Wang, 2013). Figure 3 shows the framework of learning-to-rank opinions in Twitter. First, a set of queries *Q* and related Tweets were used as training data. Every Tweet is labeled whether it is a relevant Tweet. A bag of features related to the relevance of a Tweet is extracted to form a feature *Vector*. Then a learning-to-rank algorithm is used to train a ranking model. For a new query, their related Tweets, which extract the same features to form feature *Vectors*, can be ranked by the rank function based on this model. The ranking performance of the model using a particular of feature sets in testing data can reflect the effect of these features for finding relevant Tweets.

Here SVM^{Rank} that implements the ranking algorithm is used.[7] We use a linear kernel for training and report results for the best setting of parameters (Joachims, 1999).[8] The ranking performance of model using a particular in testing data reflects the effect of those features on Tweets retrieval.

TBB Features

For every Tweet, a set of features is derived from its TBBs, which are called TBB features. These features only use the text of Tweets without relying on external social media attributions of the Tweets. We group these features according to several categories as follows.

Query : Walkman

Tweet :

$$\left(\underbrace{\text{HuffingtonPostNews: Sony Stops Production Of Cassette }\textbf{Walkman}}_{MSG}\right)$$
$$\left(\underbrace{\text{http://huff.to/aqxAMP}}_{URL}\right)\left(\underbrace{\text{\#TFB \#TAF}}_{TAG}\right)$$

FIG. 4. A query and a result Tweet.

TBB Structure Type: Each Tweet has a unique TBB structure. We represent a Tweet as a 15-dimension feature vector, where each dimension represents a frequently observed TBB structure. Fourteen dimensions of this vector are the TBB structures extracted from the 2,000 human tagged Tweets (see Table 1) and one represents all other TBB structures. If the Tweet's structure is a certain TBB structure, the corresponding element of the feature vector is assigned 1, otherwise 0; for example, the Tweet's TBB structure is "MSG URL TAG" in Figure 4, the element of the feature vector corresponding this structure is 1, the other elements are 0.

TBB Query Position: We use six binary features to indicate the positional information of the query in corresponding TBB. A phrase or a hashtag is usually used as a query to search in Twitter. So the features are whether the query is at the beginning or inside of "MSG," "COM," or "TAG" block; for example, in Figure 4, the query "Walkman" is inside of "MSG" block.

Neighbor TBB Type: The contextual information of the TBB containing query is also used. The features are whether the preceding or succeeding of TBB containing query is “TAG,” “MET,” “RWT,” “URL,” “COM,” or “MSG” block; for example, the succeeding of TBB containing query “Walkman” for the Tweet in Figure 4 is “URL” block.

TBBs Count: Intuitively, the more blocks in a Tweet that contain the query, the more this Tweet is related to the user’s requirements. Therefore, we use this feature to estimate the number of TBBs containing the query; for example, only one TBB contains the query “Walkman” for the Tweet in Figure 4.

TBB Length: The number of tokens in the longest TBB containing the query. Intuitively, a long TBB is apt to contain more information than a short one. We expect this feature as the content richness of the TBB.

TBBs OOV: This feature is calculated from the proportion of words in the TBBs containing the query that are out of vocabulary. It can measure the text quality of the block.

TBB Language: This is a binary feature indicating if the language of the longest TBB containing the query is English. People are more likely to choose native language Tweets as relevant results.

Data Set for Twitter Retrieval

In our experiments, we use the Twitter streaming API[9] as a source of documents to search over. The streaming API provides us with a sample (≈1%) of the full stream of all Tweets. The sampling algorithm used by Twitter ensures that this sample is indeed random and does not have any biases.[10] Tweets coming from the streaming API were continually indexed and a web interface for searching over this collection was provided to our annotators. We index roughly 800,000 Tweets each day. For our first experiment, three annotators were asked to use our search engine for a longer period of time (October 4, 2010, to October 28, 2010) in the same way they would use Twitter’s search engine. In addition to this, they were asked to sometimes pose Twitter’s trending topics as queries. We do this for two reasons: (a) to simulate the high-volume queries that Twitter gets and (b) to reduce the effect of data sparsity in our sample stream (we know that there are more data in the sample that relates to trending topics, by definition). After posing a query, the annotators were presented a list of n results (they could specify *n* to be what ever they like, and the default value was 10) with the highest BM25 score for the given query (based on Okapi BM25; Robertson et al., 1995). We use Lucene BM25[11] to calculate the BM25 score of a Tweet to a query. We use default setting as the specific BM25 parameters ($k1 = 2$; $b = 0.75$).

For each of the Tweets in the list, the annotators could mark it as being relevant to their query, producing binary relevance judgments for each Tweet. Note that when making their relevance judgments, annotators were also presented the information about the authors of Tweets, timestamps, and could also click on the links in the text. The latter is especially important because spammers on Twitter often write about popular topics and then insert malicious/spam links in their Tweets. For example, query *simpsons banksy* yields as one of the results Who would have thought banksy simpsons would be so popular! http://bit.ly/au21HT, where the link in the Tweet is an ad. Without following the link, one might think it is a link to the popular Simpsons intro and mark the Tweet as relevant. Some details about our data are shown in Table 7. Note that we use 100 queries, which should be sufficient for meaningful comparison.

TABLE 7. Annotated Twitter retrieval data statistics.

Number of queries	100
Trending topic queries	31
Average query length	1.48
Average number of results per query	9.36
Total relevant Tweets	184
Total nonrelevant Tweets	752

We considered the reliability of these relevance judgments and asked two researchers to decide whether the Tweets were relevant to the query in question. The two annotators had a kappa score of 0.69, which is generally considered to indicate “good” reliability.

Twitter Retrieval Experiment

To avoid overfitting the data we perform 10-fold cross-validation in our data set. We use *Mean Average Precision* (MAP) as the evaluation metric, since it is most standard among the TREC community that has been shown to have especially good discrimination and stability (Manning, Raghavan, & Schtze, 2008). Additionally, Twitter retrieval can be seen as a more precision-orientated task, hence it would be useful to report high-precision measures as well. Therefore, we use precision at rank 5 (P@5) as the other evaluation metric.

TBB feature evaluation. We take the approach of Duan et al. (2010) as the baseline. In their approach they use feature selection to choose the features and found five features listed in Table 8 can give the best published results on the Twitter search task.[12] We also develop some social media features for Tweets ranking, called SM_Rank. The features of SM_Rank are also listed in Table 8. We use our Automatic TBB Tagger to tag the Tweets and then extract TBB features for ranking, called TBB_Rank. We use various combinations of three sets of features to get different ranking methods for the test that called Baseline+SM_Rank, Baseline+TBB_Rank, SM+TBB_Rank, and Baseline+SM+TBB_Rank, respectively.

Table 9 shows the performance of these methods. We can see that just using the TBB features, which are derived from the Tweet text itself, can achieve comparable performance as the Baseline and SM_Rank methods, which utilize the social

TABLE 8. Baseline and social media features.

Baseline features	Description
Link	Whether the Tweet contains a link
Length	The number of words in the Tweet
Important_follower	The highest follower score[1] of the user who published or retweeted the Tweet
Sum_mention	Sum of mention scores[2] of users who published or retweeted the Tweet
First_list	List score[3] of the user who published the Tweet
Social media features	**Description**
Followers count	The number of followers the author has
Friends count	The number of friends the author has
Listed count	List score
Author mentions	Whether the Tweet has mentions
Hashtags count	The number of hashtags in the Tweet
Reply	Is the current Tweet a reply
retweeted	Whether the current Tweet was retweeted
Source Web	Whether the source of the Tweet is web
Statuses count	The number of statuses of the Tweet's author
retweet count	How many times has this Tweet been retweeted
Author retweet count	The number of times the author has been retweeted
Overlap words	Overlap (Jaccard score) between query and the Tweet
Tweet timestamps	How long (in seconds) did the user publish the Tweet before the query submitted

[1] Follower score: number of followers a user has.
[2] Mention score: number of times a user is referred to in Tweets.
[3] List score: number of lists a user appears in.

TABLE 9. Performance of ranking methods.

	MAP	P@5
Baseline	0.4197	0.2400
SM_Rank	0.4338	0.2667
TBB_Rank	0.4235	0.2578
Baseline+SM_Rank	0.4546	0.2800*
Baseline+TBB_Rank	0.4326	0.2622*
SM+TBB_Rank	0.4710*†	0.2800*
Baseline+SM+TBB_Rank	0.4712*†	0.2756*

A star (*) and dagger (†) indicate statistical improvement over the Baseline and SM_Rank, respectively.

media information. We conducted a paired *t*-test between the results of these three methods and found no statistically significant difference (at $p = .05$) by MAP and P@5. By adding social media features to TBB_Rank we get a significant improvement in MAP (at $p = .05$). Moreover, the ranking of Baseline+TBB_Rank and SM+TBB_Rank by P@5 are higher than Baseline. Lastly, combinations of all three sets of features provide the highest MAP value. All the results suggest that structural information of Tweets can improve Twitter retrieval.

Feature selection. There is a large set of features in the Baseline+SM+TBB_Rank that get the best results in Tweets ranking. But sometimes subsets of features may get comparable performance even better. We use a novel feature selection method to find the best features conjunction to improve the performance.

We apply feature selector based on max-relevance and min-redundancy criterion (Peng, Long, & Ding, 2005) as follows:

$$\max_{s} D - R \tag{1}$$

where D and R are average feature relevance and redundancy measure for feature subset S with m individual features $\{X_i\}_{i=1}^{m}$, respectively. The D and R are given as:

$$D = \frac{1}{|S|}\sum_{x \in S} I(x_i, c)$$

and

$$R = \frac{1}{|S|^2}\sum_{x_i, x_j \in S} I(x_i, x_j)$$

I(xi, c) and *I(xi, xj)* denote the feature-class and feature-feature correlation. In this study, *xi* refer to our features while c indicates the binary class of retrieved Tweets with $c = 1$ indicate relevant Tweet to search query and $c = 0$ for otherwise. The correlation I is obtained by averaging over individual correlation *Iq* corresponding for each query *q*. *Iq* itself is computed from the sample points of Tweets pre-retrieved by BM25 algorithm according to query *q*. To avoid exhaustive search of S in (1), suboptimal incremental forward search is used where a best feature is added at each step sequentially to the selected subset. Let us denote S_{m-1} as a selected feature set with $m - 1$ features and F as a full M-feature set. Then the task is to select m^{th} feature from the remaining set $F - S_{m-1}$ according to the following condition (Peng et al., 2005)

$$x_m = \arg\max_{x_j \in F - S_{m-1}} \left[I(x_j, c) - \frac{1}{m-1} \sum_{x_i \in S_{m-1}} I(x_i, x_j) \right]$$

and append xm to the selected feature subset S_{m-1}. Then it sets $m = m + 1$ and repeats the selection until last feature $m = M - 1$. This leads to ranked M sequential feature sets $S_1 \subset S_2 \subset \ldots \subset S_{M-1} \subset S_M$. The best feature set with highest learning-to-rank cross-validation performance (by MAP) on validation data is selected. This selected set will be evaluated again on testing data and compared with use of the full subset.[13]

Table 10 lists the subset of 12 features that can achieve best performance of Tweets ranking in our validation data by our feature selection approach. We use the ranking approach based on these features. Table 11 shows the result. Best_Rank is the performance of method using features selection. It shows this method can achieve a better result than Baseline+SM+TBB_Rank. We conduct a paired *t*-test between Best_Rank and two baselines. The results

TABLE 10. The selected features subset.

TBB structure type (MSG URL)
TBB structure type (MSG URL TAG)
TBB query position (head of MSG)
TBB query position (inside of MSG)
Neighbor TBB type (MET is preceding TBB)
TBB length
retweet count
Source web
Reply
Tweet timestamps
Overlap words statuses count

TABLE 11. Performance of feature selection.

	MAP	P@5
Baseline	0.4197	0.2400
SM_Rank	0.4338	0.2667
Baseline+SM+TBB _Rank	0.4712*†	0.2756*
Best_Rank	0.4950*†	0.2842*†

A star (*) and dagger (†) indicate statistical improvement over the Baseline and SM_Rank, respectively.

TABLE 12 Features in TBB structure type related to the "URL" block.

URL block features	Description
MSG URL	Whether the Tweet's TBB structure is "MSG URL" RWT
MSG URL	Whether the Tweet's TBB structure is "RWT MSG URL"
COM URL	Whether the Tweet's TBB structure is "COM URL"
TAG MSG URL	Whether the Tweet's TBB structure is "TAG MSG URL"
RWT MSG URL	Whether the Tweet's TBB structure is "RWT MSG URL"
MSG URL TAG	Whether the Tweet's TBB structure is "MSG URL TAG"
OTHER URL	Whether the Tweet's TBB structure is the other infrequent structures containing "URL"

TABLE 13. Performance of each feature related TBB structure containing URL.

	MAP	P@5
Baseline	0.4197	0.2400
MSG URL	0.4019	0.2313
MSG URL TAG	0.3327	0.1903
RWT MSG URL	0.3289	0.1884
TAG MSG URL	0.3245	0.1896
COM URL	0.3191	0.1825
OTHER URL	0.1984	0.1124
MET MSG URL	0.1932	0.1105

demonstrate that Best_Rank both outperforms two baselines with a significance level of 0.01.

We present our TBB features from Table 10. The feature whether the Tweet contains "URL" block is effective for Tweets ranking. This argument is the same as Duan et al. (2010). Moreover, we also found "MSG URL" is an important TBB structure for Twitter retrieval. TBB Query Position about "MSG" suggests that the appearance of query in the Tweet is also important. "MET is preceding TBB" shows that most of replying Tweets are unmeaningful information. The last "TBB Length" shows the informative level of TBB that can help Tweets ranking.

URL block features evaluation. Duan et al. (2010) found that the existence of links in a Tweet is the most effective feature for Tweet retrieval. We are interested in which TBB structures containing the "URL" block are highly valued for ranking. We test the features of TBB Structure Type related to the "URL" block (called URL Block Features), which are listed in Table 12. We evaluate the effect of each feature by replacing the Link feature in the Baseline method for Tweets ranking. Table 13 gives the performance of each feature related TBBs structure containing the "URL" block.

We can see from Table 13 that only the "MSG URL" ranking method gives comparable performance as the Baseline by MAP and P@5 (there is no significant difference between them at $p = 0.05$). The performance of other ranking methods declines seriously. This shows that the feature indicating whether the Tweet's TBB structure is "MSG URL" can replace the Link feature in the Baseline model. The reason may be that most TBB structures for Tweets containing links are "MSG URL" (see Table 1) and the Tweets with this structure are more likely to be relevant Tweets than the other structures containing the "URL" block. For example, the query *wikileaks* yields two Tweets in our data:

(a) Obama administration braces for WikiLeaks release of thousands of secret documents on Iraq war (Star Tribune) http://bit.ly/9lnBGB
(b) BBCWorld: Wikileaks files "threaten troops" http://bbc.in/c4Sznk: BBCWorld: Wikileaks files "threaten troops"... http://dlvr.it/7P7zM

Annotators tag Tweet (a) as the relevant Tweet and Tweet (b) as the nonrelevant one. TBB structure for Tweet (a) and (b) are "MSG URL" and "MSG URL MSG URL," respectively. The reason the annotators tag Tweet (b) as the nonrelevant Tweet is that this Tweet has two "URL" blocks, which makes the Tweet messy. In our experiment both of Baseline and SM_Rank rank Tweet (b) higher than Tweet (a), but our TBB_Rank ranks Tweet (a) higher. It shows that our TBB can capture more information about the Tweets containing links that can improve Tweets ranking.

TABLE 14. Performance of TREC microblog track testing.

	MAP	P@30
TREC_BM25	0.4120	0.3015
TREC_Best_Rank	0.4417*	0.3321*

A star (*) indicates statistical improvement over the Baseline and TREC_BM25.

TREC Microblog Track Testing

Judgment level of about 10 results per query and only using one run BM25 to collect judgments may cause the collection to be non-reusable for evaluating other algorithms, and a bias toward BM25-like algorithms may exhibit. Basically, when only using top results from BM25 to provide judgments, all other retrieval algorithms evaluated will show a lower bound score. That is because all unjudged documents will be seen as irrelevant. That's why TREC uses many more than two different retrieval algorithms to pool results and evaluates to the depth of 1,000 or more through sampling. For these reasons, we use the TREC Microblog Track data sets to estimate the effectiveness of our approach for ad-hoc search task in Twitter.[14]

In TREC 2011 and 2012, the Microblog track used the Tweets2011 collection specifically created for ad-hoc Twitter retrieval evaluations. The Tweets2011 collection is comprised of 16M Tweets spread over 2 weeks in 2011, sampled courtesy of Twitter (Ounis et al., 2011). Besides the Tweets, 109 topics were given as the targets of retrieval. We utilized an existing language detection library to identify English Tweets and found that 4,766,901 Tweets were classified as English. Then we indexed these English Tweets and implemented a search engine. We posed the Microblog track 109 topics as queries and got a list of 100 Tweets ranked based on the BM25 score for each query. We also use Lucene-BM25[15] to calculate the BM25 score of a Tweet to a query. Finally, we got 10,521 ranked Tweets and 2,046 Tweets were topic-relevant Tweets. We are interested to know how effective our TBB approach is when it is applied to re-rank these Tweets.

Here, we used the baseline based on BM25 score called TREC_BM25. Then we add the features listed in Table 10 into a ranking model (TREC_Best_Rank). We also perform 10-fold cross-validation in this data set. The primary measures for retrieval effectiveness were precision at rank 30 (P@30) and mean average precision (MAP) (Ounis et al., 2011). We also use these evaluation metrics. Table 14 shows the results. We can see that the performance of TREC_Best_Rank is significantly better than TREC_BM25. It means that our approach is still effective when using the TREC Tweets2011 corpus.

TBB for Opinion Retrieval in Twitter

Opinion Retrieval in Twitter

Opinion retrieval deals with finding relevant documents that express either a negative or positive opinion about some topics. Social Networks such as Twitter, where people routinely post opinions about almost any topic, are rich environments for opinions.

We also use a standard machine-learning approach to learn a ranking function for Tweets that uses the opinionated feature in addition to traditional topic-relevant features such as BM25. We use corpus-derived method to estimate the opinionatedness value of the Tweet as a feature for ranking.

An Opinionatedness Feature

The opinionatedness score of a Tweet is essential for an opinion retrieval task. Previous approaches for this estimation are divided into two categories: (a) classification approach and (b) lexicon-based approach (Na, Lee, Nam, & Lee, 2009). We adopt the lexicon-based approach, since it is simple and not dependent on machine-learning techniques. However, a lexicon such as MPQA Subjectivity Lexicon[16] that is widely used might not be effective in Twitter, since the textual content of a Tweet is often very short and highly informal. Therefore, we use a corpus-derived lexicon to construct an opinion score for each Tweet. We estimate the opinionatedness score of each Tweet by calculating the average opinion score over certain terms. We use the chi-square value, based on manually tagged subjective Tweets set and objective Tweets set, to estimate the opinion score of a term. The score measures how dependent a term is with respect to the subjective Tweets set or objective Tweets set. For all terms in a Tweet, we only keep the terms with a chi-square value no less than m. The estimated formula is as follows:

$$Opinion_{avg}(d) = \sum_{t \in d.\chi^2 \geq m} p(t|d) \times Opinion(t)$$

where $p(t|d) = c(t, d)/|d|$ is the relative frequency of a term t in the Tweet d. $c(t, d)$ is the frequency of term t in the Tweet d. $|d|$ is the number of terms in the Tweet d.

$$Opinion(t) = \mathrm{sgn}\left(\frac{o_{11}}{o_{1*}} - \frac{o_{11}}{o_{1*}}\right) * \chi^2(t)$$

where $sgn(*)$ is a sign function. $\chi^2(t)$ calculates chi-square value of a term.

$$\chi^2(t) = \frac{(o_{11}o_{12} - o_{12}o_{21}) * o}{o_{1*}o_{2*}o_{1*}o_{2*}}$$

O_{ij} in Table 15 is counted as the number of Tweets having term t in the subjective/objective Tweets set, respectively. For example O_{12} is the number of Tweets not having term t in the subjective Tweets set.

Manually labeling the Tweets necessary for constructing opinionated scoring is time-consuming and also topic-dependent. For example, Tweets about "android" might contain opinionated terms "open," "fast," and

TABLE 15. Pearson's chi-square.

	t	$\neg t$	Row total
Sub. set	$O11$	$O12$	$O1*$
Obj. set	$O21$	$O22$	$O2*$
Col. total	$O*1$	$O*2$	O

$O1* = O11 + O12$, $O2* = O21 + O22$, $O*1 = O11 + O21$, $O*2 = O12 + O22$, $O = O11 + O12 + O21 + O22$.

TABLE 16. Annotated Twitter opinion retrieval data statistics.

Number of queries	50
Average query length	1.94
Average number of results per query	100
Total relevant Tweets	831
Total nonrelevant Tweets	4169

"excellent," but these terms are unlikely to be the subjective clues of Tweets related to some news events (e.g., "UK strike"). It is clearly impossible to tag a large number of Tweets for every given topic. Therefore, we develop an approach to collect "pseudo" subjective Tweets (PSTs) and "pseudo" objective Tweets (POTs) automatically.

In Twitter, some simple structural information of Tweets and users' information can be used to generate PSTs and POTs. For example, people usually reTweet another user's Tweet and give a comment before Tweet. The TBB structures of these Tweets usually begin with "COM RWT." Tweets with this structure are more likely to be subjective. Many Tweets posted by news agencies, who usually post many Tweets and have many followers, are likely to be objective Tweets and these Tweets usually contain links. The TBB structures of these Tweets contain "URL" block. We define these two types of Tweets as follows:

1. "Pseudo" Subjective Tweet (PST): the TBB structure of a Tweet begins with "COM RWT." For example, a Tweet "I thought we were isolated and no one would want to invest here! RT @BBCNews: Honda announces 500 new jobs in Swindon bbc. in/ vT12YY" is a pseudo subjective Tweet and its TBB structure is "COM RWT MSG URL."
2. "Pseudo" Objective Tweet (POT): If a Tweet satisfies two criteria: (a) it contains "URL" block and (b) the user of this Tweet posted many Tweets before and has many followers. This Tweet is likely to be an objective Tweet; for example "#NorthKorea:#KimJongil died after suffering massive heart attack on train on Saturday, official news agency reports bbc. in/ vzPGY5."

Using the definition introduced above, it is easy for us to construct patterns and collect a large number of PSTs and POTs from Twitter. We assume that the Tweets in the PST set are all subjective Tweets and the Tweets in the POT set are all objective Tweets. Although this is not 100% true, the subjective Tweets portion should be dominant in the PST set so that the effect of the objective Tweets portion can be neglected. It is opposite in the POT set. Since the TBB information and authors' information are independent of the topic of a Tweet, if there are a lot of Tweets related to a given topic, it is easy to collect topic-dependent PSTs and POTs.

Data Set for Twitter Opinion Retrieval

There exists no ground truth data set for evaluating opinion retrieval in Twitter. To identify which of the Tweets is topic related to a query and also contains opinions about it (query), we therefore create a new data set by ourselves.

We crawled and indexed about 30 million Tweets using the Twitter API in November 2011. All Tweets are in English. Using these Tweets we implemented a search engine using Lucene. Seven people (a woman and six men) were asked to use our search engine. All of them are not native speakers but are good at it. They were allowed to post any query. Given a query the search engine would present a list of 100 Tweets ranked based on the BM25 score. We use default setting as the specific BM25 parameters ($k1 = 2$; $b = 0.75$). All the queries were issued on December 1, 2011.

Based on the principle about the Tweet whether it is topic related and expresses opinions about it (query), the user who issued the query assigned a binary label to every Tweet. If a Tweet is topic related to a query and also contains opinions judged by the people who issued the query, the score of Tweet class is 1, otherwise the score is 0. We emphasize that if a Tweet is only topic related or just contains opinions, it is not a relevant Tweet in our task. Table 16 shows some details about our opinion retrieval in Twitter data.

We also considered the reliability of these opinionated relevance judgments. For each query, we randomly sampled 10 relevant Tweets (as labeled by our original judges) and asked two annotators to judge the relevance. The two annotators had a kappa score of 0.54, which is generally considered to indicate "fair" reliability.

Twitter Opinion Retrieval Experiment

For learning-to-rank, SVM^{Rank} which implements the ranking algorithm is used. We use a linear kernel for training and report results for the best setting of parameters. We perform 10-fold cross-validation in our data set and use MAP and precision at rank 5 (P@5) as the evaluation metrics.

We investigate the opinionatedness feature for Tweets ranking. As a baseline, we use the ranking approach that uses the Okapi BM25 score of each Tweet as a feature for modeling. Because Tweets are short texts and BM25 is well known for evening out high variation in the length of documents without fully normalizing for length, BM25 might not be the most natural baseline. We use the classic VSM as the other baseline proposed by Salton, Wong, and Yang (1975). In our experiments, we take advantage of Lucene to implement VSM.

DOI: 10.1002/asi

To automatically generate “pseudo” subjective Tweets (PSTs) and “pseudo” objective Tweets (POTs), we design some simple patterns: we first use our TBB tagger to label the TBB structures of 1-month Tweets in November 2011. For PSTs generation, we choose the Tweets that begin with “COM RWT” blocks. Additionally, we find the length of “COM” block should be no less than 10 characters. For POTs generation, we choose the Tweets that contain “URL” block, the author for each Tweet has no less than 1,000 followers, and has posted at least 10,000 Tweets. In our 1-month Tweets data set, 4.64% Tweets are PSTs and 1.35% Tweets are POTs.

We asked one researcher (nonauthors) to spot-check the quality of our automatically harvested Tweets. We randomly selected 100 PSTs and 100 POTs and manually inspected them, judging the extent to which there were subjective or objective. In these Tweets, 95% PSTs were subjective Tweets and 85% POTs were objective Tweets. This supports the idea that our approach can generate a large number of accurate PSTs and POTs. Hence, we randomly choose 4,500 English PSTs and POTs to form a topic-independent data set.

Opinionatedness feature evaluation. In our corpus-derived approach, we use the Porter English stemmer and stop words[17] to preprocess the text of Tweets. Using these Tweet data sets we can calculate the value of opinionatedness score for a new Tweet. To achieve the best performance of Tweets ranking, we set the threshold of *m* to 5.02 corresponding to the significance level of 0.025 for each term in a data set. This setting is the same as Zhang et al.’s (2007) work. We call the feature using topic-independent data set to estimate the opinionatedness score Q_I. Previous work uses manually tagged blogs to estimate the opinionatedness score of a new blog (Gerani et al., 2009; He et al., 2008). In our experiment we use the manually tagged Tweets training data in each fold to estimate the opinionatedness score of a new Tweet. We compare the method, using Gold feature based on these manually tagged Tweets, with the method using our Q_I feature for Tweets ranking. We also develop another two features for comparison. MPQA_Lexicon feature: if a Tweet contains a word or a phrase in MPQA Subjectivity Lexicon, the opinionatedness score of the Tweet is 1, otherwise the score is 0. TwitterSenti feature: If the Tweet is a positive Tweet or a negative Tweet judged by public Twitter Sentiment API[18] (Go, Bhayani, & Huang, 2009), the Tweet’s opinionatedness score is 1. This API judges the Tweet as a neutral Tweet, the Tweet’s opinionatedness score is 0.[19]

TABLE 17. Performance of ranking method using different opinionatedness features).

	MAP	P@5
BM25	0.2509	0.1960
BM25+MPQA Lexicon	0.2553$^{\cdot}$	0.1960
BM25+TwitterSenti	0.2958$^{\cdot}$	0.3000$^{\cdot}$
BM25+Gold	0.3615$^{\cdot}$	0.4120$^{\cdot}$
BM25+Q I	0.3602$^{\cdot}$	0.3880$^{\cdot}$

A significant improvement over the BM25 ranking method with $^{\Delta}$ and $^{\cdot}$ (for $p < 0.05$ and $p < 0.01$).

TABLE 18. Performance of ranking method using different opinionatedness features.

	MAP	P@5
VSM	0.2812	0.2495
VSM +MPQA_Lexicon	0.2876	0.2520
VSM +TwitterSenti	0.3244$^{\cdot}$	0.3000$^{\Delta}$
VSM +Gold	0.3485$^{\cdot}$	0.4080$^{\cdot}$
VSM +Q_I	0.3566$^{\cdot}$	0.3880$^{\cdot}$

A significant improvement over the VSM ranking method with $^{\Delta}$ and $^{\cdot}$ (for $p < 0.05$ and $p < 0.01$).

TABLE 19. Performance of ranking method using Q_D feature.

	MAP	P@5
BM25+Q_I	0.3602	0.3880
BM25+Q_D	0.3667$^{\Delta}$	0.4000

A significant improvement over the BM25+Q_I ranking method with $^{\Delta}$ and $^{\cdot}$ (for $p < 0.05$ and $p < 0.01$).

Tables 17 and 18 show the results of ranking using the opinionatedness features. We can see that all the methods using opinionatedness features improve the opinion retrieval performance over the BM25 and VSM. It shows estimating the opinionatedness score of a Tweet is essential for opinion retrieval task. Although the BM25+MPQA_Lexicon and VSM+MPQA_Lexicon method, using the MPQA_Lexicon feature, can improve Tweet ranking by MAP and P@5, the results achieved are worse than the other ranking methods. The reason is that the text of Tweets is different from other documents (e.g., reviews and blogs) and the MPQA Subjectivity Lexicon is not effective enough for Twitter analysis. The ranking method using Q_I feature can achieve comparable performance with the BM25+Gold method (there are no significant difference at $p = .05$). It suggests that using structural information and social information of Tweets to generate accurate PSTs and POTs automatically is useful for opinion retrieval in Twitter. Importantly, this method does not need any manually tagged Tweets.

Feature based on topic-dependent PSTs and POTs evaluation. Another advantage of our approach is that it is easy to gather topic-dependent PSTs and POTs. We use all PSTs and POTs introduced above to implement a search engine. Given a query, the search engine can give any number of query-dependent PSTs and POTs ranked by topic relevance. We generate 4,500 query-dependent PSTs and POTs for each query. Using each query-dependent Tweets we calculate the opinionatedness feature (called Q_D feature) for a new Tweet. Tables 19 and 20 show the results of ranking methods using the Q_D feature. It improves the

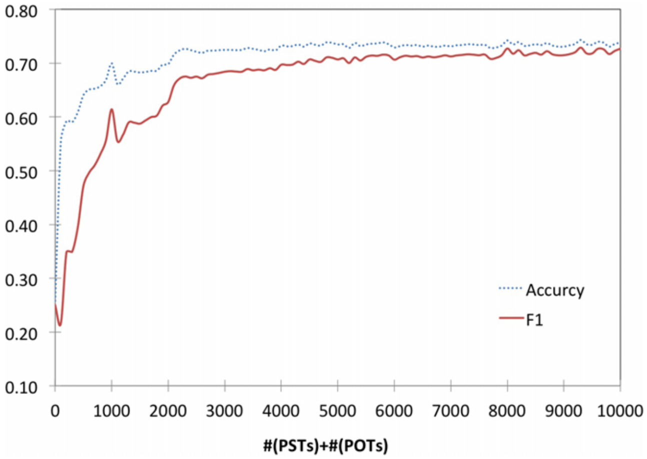


FIG. 5. Performance of Tweet classifier based on Opinion_avg(d). [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

TABLE 20. Performance of ranking method using Q D feature.

	MAP	P@5
VSM+Q_I	0.3566	0.3880
VSM+Q_D	0.3588	0.3880

A significant improvement over the VSM+Q_I ranking method with $^{\Delta}$ and $^{\cdot}$ (for $p < 0.05$ and $p < 0.01$).

opinion retrieval in Twitter over the BM25+Q_I ranking method that do not consider query-dependent situations. It means our approach based on TBB structures of Tweets can help resolving query-dependent problem for opinion retrieval in Twitter.

Table 21 shows a list of the highest score of $\chi^2(t)$ opinion terms derived from different query-dependent PSTs and POTs and query-independent PSTs and POTs (total 1,000 Tweets for each query). We can see that our approach can assign high scores to terms such as personal pronoun (e.g., “*i*,” *u*,” and “*my*”) and emoticons (e.g., “*:)*,” “*:(*“ and “*:d*”). The reason is that personal content Tweets are more likely to be subjective Tweets. For query-dependent PSTs and POTs, our approach successfully extracts the opinionated feature “*excit” (Opinion(t)* > 0) which can express attitude about the movie “Breaking Dawn,” and this term is unlikely to be used in the opinionated Tweets related to “UK strike” topic. In PSTs and POTs related to the “UK strike” topic, we discover (unsurprisingly) that the term “*bbc*” (*Opinion(t)* < 0) is more likely to appear in the objective Tweets posted by BBC news.

Subjective tweets classification. We are also interested in how the Opinion_avg(d) formula (this section) would perform if used as a Tweet classifier. We use 1,000 manually tagged subjective Tweets and 1,000 manually tagged objective Tweets as a gold testing data. We take a Tweet *d* which the value of *Opinion avg(d)* is more than 0 as a subjective Tweet and a Tweet which the value is no more than 0 as an

TABLE 21. The highest score of $\chi 2(t)$ opinion terms derived from different query-dependent PSTs and POTs and query-independent PSTs and POTs.

Rank	Breaking Dawn	HTC	Obama	UK strike	Q_I
1	i +	i +	i +	. . . -	i +
2	video –	lol +	you +	i +	lol +
3	go +	.. +	#obama –	followfridai –	:) +
4	.. +	u +	my +	rank –	.. +
5	me +	my +	lol +	you +	u +
6	lol +	new –	u +	my +	* +
7	new –	:) +	!! +	lol +	new –
8	via –	me +	me +	week –	my +
9	!!! +	* +	barack –	last –	morn +
10	wait +	rezound –	#tcot –	:) +	me +
11	pattinson –	you +	. . . –	u +	!!! +
12	robert –	phone –	cont +	me +	good +
13	. . . –	like +	.. +	thi +	:d +
14	so +	:d +	presid –	so +	via –
15	too +	!!! +	* +	!! +	!! +
16	:) +	morn +	i’m +	#ows –	cont +
17	see +	i’m +	:) +	#jobs –	haha +
18	can’t +	good +	we +	x +	ya +
19	:d +	!! +	do +	come +	too +
20	premier –	. . . –	he +	3 +	. . . –
21	kristen –	too +	obama’ –	gener –	I’m +
22	excit +	cream –	!!! +	onli +	:(+
23	again +	cont +	#news –	good +	thank +
24	i’m +	so +	know +	bbc –	,) +
25	im +	thank +	lmao +	here +	@damnitstrue +

“+” is the score of *Opinion*(*t*) no less than 5.02. “–” is the score of *Opinion*(*t*) no more than -5.02.

objective Tweet. Both accuracy and F1 are our classification evaluation metric. Figure 5 shows the performance of Opinion_avg(d) formula as a Tweet classifier with different numbers of PSTs and POTs. We can see that the performance of classifier using more than ~2,200 PSTs and POTs can achieve the value of accuracy 0.72 and F1 0.67. There is no significant improvement when using more PSTs and POTs. All these show that using the Opinion_avg(d) formula as a classifier can judge the subjective Tweets effectively. It also verifies that our TBB structuring Tweets approach can help opinion analysis in Twitter.

Conclusion

In this paper we introduced Twitter Building Blocks (TBBs) and their structural combinations (TBB structures), to capture structural information of Tweets. We showed that the TBB structures have very different properties, for example, their out-of-vocabulary (OOV) values are very different. We used this structural information for Twitter retrieval. The experimental results showed that the ranking approach using the TBB features alone achieved comparable performance to the state-of-the-art method. Additionally, we proposed a novel automatic approach which uses the structural information and social information of the Tweets to generate accurate "pseudo" subjective Tweets (PSTs) and "pseudo" subjective Tweets (POTs) automatically. Opinionated retrieval results using this information comparable to results using manually labeled data. All these show that although the texts of Tweets are very short, their structural information can improve Twitter search.

For future work we plan to use the TBB in other Twitter applications, for example, user clustering, spam filtering, and so on.

Endnotes

[1] We filtered English Tweets using a language-detection toolkit from http://code.google.com/p/language-detection/

[2] http://github.com.brendano/Tweetmitif

[3] We also tested the labeling tagger model with the window size 3, 5, 9 and the tagger with window 7 gives the best result.

[4] We used a part-of-speech Tweet tagger http://www.ark.cs.cmu.edu/TweetNLP

[5] http://flexcrfs.sourceforge.net/

[6] Here we only calculate the words appearing in "MSG" and "COM" blocks.

[7] http://www.cs.cornell.edu/people/tj/svm_light/svm_rank.html

[8] Since there are many binary features in our ranking model, we also tested a tree-based learner for ranking. But there is no significant difference from the linear learner in raking performance.

[9] http://stream.twitter.com/

[10] https://dev.twitter.com/docs/streaming-apis

[11] http://nlp.uned.es/~jperezi/Lucene-BM25/

[12] We do not take the method based on BM25 score as a baseline, since the Duan et al. (2010) method performs significantly better than BM25 and our method also performs better than it. The MAP value of BM25 method is 0.344 in our experiment.

[13] Here we choose Tweets of 90 queries as training data. The remaining Tweets of 10 queries are divided into validation data and testing data equally.

[14] http://trec.nist.gov/data/Tweets/

[15] http://nlp.uned.es/~jperezi/Lucene-BM25/

[16] http://www.cs.pitt.edu/mpqa/

[17] It contains standard stop words, commonly used punctuation and the Twitter convention "RT."

[18] http://www.sentiment140.com

[19] It does not mean the neutral Tweet is an objective Tweet. Actually some subjective Tweets have no polarity, but it is out of consideration in this paper.

References

Agichtein, E., Castillo, C., Donato, D., Gionis, A., & Mishne, G. (2008). Finding high-quality content in social media. In Proceedings of the 2008 International Conference on Web Search and Data Mining (pp. 183–194). New York: ACM.

Ahnizeret, K., Fernandes, D., Cavalcanti, J.M.B., de Moura, E.S., & da Silva, A.S. (2004). Information retrieval aware web site modelling and generation. In Conceptual Modeling 2004 (pp. 402–419). Berlin: Springer.

Amati, G., Ambrosi, E., Bianchi, M., Gaibisso, C., & Gambosi, G. (2008). Automatic construction of an opinion-term vocabulary for ad hoc retrieval. In Proceedings of the IR Research, 30th European Conference on Advances in Information Retrieval. Ecir'08 (pp. 89–100). Berlin, Heidelberg: Springer.

Amati, G., Amodeo, G., Bianchi, M., Marcone, F., Ugo Bordoni, F., Gaibisso, C., . . . Flammini, M. (2011). FUB, IASI-CNR, UNIVAQ at TREC 2011 microblog track. In TREC.

Berendsen, R., Tsagkias, N., Weerkamp, W., & de Rijke, M. (2013). Pseudo test collections for training and tuning microblog rankers. In Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval. SIGIR '13 (pp. 53–62). New York: ACM.

Cai, D., Yu, S., Wen, J.I., & Ma, W.Y. (2004). Block-based web search. In Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (pp. 456–463). New York: ACM.

Callan, J.P. (1994). Passage-level evidence in document retrieval. In Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (pp. 302–310). New York: Springer.

Cha, M., Haddadi, H., Benevenuto, F., & Gummadi, P.K. (2010). Measuring user influence in twitter: The million follower fallacy. ICWSM, 10, 10–17.

Cheng, F., Zhang, X., He, B., Luo, T., & Wang, W. (2013). A survey of learning to rank for real-time twitter search. In Pervasive computing and the networked world (pp. 150–164). Berlin: Springer.

Choi, J., Croft, W.B., & Kim, J.Y. (2012). Quality models for microblog retrieval. In Proceedings of the 21st ACM International Conference on Information and Knowledge Management (pp. 1834–1838). New York: ACM.

de Moura, E.S., Fernandes, D., Ribeiro-Neto, B., da Silva, A.S., & Goncalves, M.A. (2010). Using structural information to improve search in web collections. Journal of the American Society for Information Science and Technology, 61(12), 2503–2513.

Duan, Y., Jiang, L., Qin, T., Zhou, M., & Shum, H.Y. (2010). An empirical study on learning to rank of Tweets. In Proceedings of the 23rd International Conference on Computational Linguistics (pp. 295–303). Stroudsburg, PA: Association for Computational Linguistics.

Efron, M. (2010). Hashtag retrieval in a microblogging environment. In Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval (pp. 787–788). New York: ACM.

Efron, M. (2011). Information search and retrieval in microblogs. Journal of the American Society for Information Science and Technology, 62(6), 996–1008.

Eguchi, K., & Lavrenko, V. (2006). Sentiment retrieval using generative models. In Proceedings of the 2006 Conference on Empirical Methods in

Natural Language Processing. EMNLP '06 (pp. 345–354). Stroudsburg, PA: Association for Computational Linguistics.

Ferguson, P., O'Hare, N., Lanagan, J., Phelan, O., & McCarthy, K. (2012). An investigation of term weighting approaches for microblog retrieval. In Advances in information retrieval (pp. 552–555). Berlin: Springer.

Fernandes, D., de Moura, E.S., Ribeiro-Neto, B., da Silva, A.S., & Goncalves, MA. (2007). Computing block importance for searching on web sites. In Proceedings of the 16th ACM Conference on Information and Knowledge Management (pp. 165–174). New York: ACM.

Foster, J., Oetinoglu, O., Wagner, H., Le Roux, J., Hogan, S., Nivre, J., . . . Van Genabith, J. (2011). # hardtoparse: Pos tagging and parsing the twitterverse. In Proceedings of the Workshop on Analyzing Microtext (AAAI 2011; pp. 20–25).

Gerani, S., Carman, M.J., & Crestani, F. (2009). Investigating learning approaches for blog post opinion retrieval. In Proceedings of the 31th European Conference on IR Research on Advances in Information Retrieval. ECIR '09 (pp. 313–324). Berlin, Heidelberg: Springer.

Gimpel, K., Schneider, N., O'Connor, B., Das, D., Mills, D., Eisenstein, J., . . . Smith, N.A. (2011). Part-of-speech tagging for twitter: Annotation, features, and experiments. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-volume 2 (pp. 42–47). Stroudsburg, PA: Association for Computational Linguistics.

Go, A., Bhayani, R., & Huang, L. (2009). Twitter sentiment classification using distant supervision. Processing.

Han, B., & Baldwin, T. (2011). Lexical normalisation of short text messages: makn sens a# twitter. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies Volume 1 (pp. 368–378). Stroudsburg, PA: Association for Computational Linguistics.

Han, Z., Li, X., Yang, M., Qi, H., & Li, S. (2013). Feature analysis in microblog retrieval based on learning to rank. In Natural Language Processing and Chinese Computing (pp. 410–416). Berlin: Springer.

He, B., Macdonald, C., He, J., & Ounis, I. (2008). An effective statistical approach to blog post opinion retrieval. In Proceedings of the 17th ACM Conference on Information and Knowledge Management. CIKM '08 (pp. 1063–1072). New York: ACM.

Huang, X., & Croft, W.B. (2009). A unified relevance model for opinion retrieval. In Proceedings of the 18th ACM Conference on Information and Knowledge Management. CIKM '09 (pp. 947–956). New York: ACM.

Jiang, L., Yu, M., Zhou, M., Liu, S., & Zhao, T. (2011). Target-dependent twitter sentiment classification. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Vol. 1. HLT '11 (pp. 151–160). Stroudsburg, PA: Association for Computational Linguistics.

Jijkoun, V., de Rijke, M., & Weerkamp, W. (2010). Generating focused topic-specific sentiment lexicons. In Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics. ACL '10 (pp. 585–594). Stroudsburg, PA: Association for Computational Linguistics.

Joachims, T. (1999). Making large-scale support vector machine learning practical. In B. Scholkopf, C.J.C. Burges, & A.J. Smola (Eds.), Advances in kernel methods (pp. 169–184). Cambridge, MA: MIT Press.

Lafferty, J.D., McCallum, A., & Pereira, F.C.N. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In Proceedings of the 18th International Conference on Machine Learning. ICML '01 (pp. 282–289). San Francisco: Morgan Kaufmann Publishers.

Li, B., Zhou, L., Feng, S., & Wong, K-F. (2010). A unified graph model for sentence-based opinion retrieval. In Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics. ACL '10 (pp. 1367–1375). Stroudsburg, PA: Association for Computational Linguistics.

Li, Y., Zhang, Z., Lv, W., Xie, Q., Lin, Y., Xu, R., . . . Guo, J. (2011). Pris at trec 2011 microblog track. In TREC.

Liu, X., Zhang, S., Wei, R., & Zhou, M. (2011). Recognizing named entities in Tweets. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies Volume 1 (pp. 359–367). Stroudsburg, PA: Association for Computational Linguistics.

Luo, Z., Osborne, F., Petrovic, S., & Wang, T. (2012). Improving twitter retrieval by exploiting structural information. In AAAI '12: Proceedings of the 26th AAAI.

Luo, Z., Osborne, M., Tang, J., & Wang, T. (2013). Who will reTweet me?: Finding reTweeters in twitter. In SIGIR (pp. 869–872).

Luo, Z., Osborne, M., & Wang, T. (2012). Opinion retrieval in twitter. In 6th International AAAI Conference on Weblogs and Social Media.

Luo, Z., Osborne, M., & Wang, T. (2013). An effective approach to Tweets opinion retrieval. World Wide Web.

Luo, Z., Tang, J., & Wang, T. (2013). Propagated opinion retrieval in twitter. In Web information systems engineering-wise 2013 (pp. 16–28). Berlin: Springer.

Macdonald, C., Ounis, I., & Soboroff, I. (2007). Overview of the TREC 2007 blog track. In TREC.

Manning, C.D., Raghavan, P., & Schtze, H. (2008). Introduction to information retrieval. New York: Cambridge University Press.

Massoudi, K., Tsagkias, M., de Rijke, M., & Weerkamp, W. (2011). Incorporating query expansion and quality indicators in searching microblog posts. In Advances in information retrieval (pp. 362–367). Berlin: Springer.

McCreadie, R., & Macdonald, C. (2013). Relevance in microblogs: enhancing Tweet retrieval using hyperlinked documents. In Proceedings of the 10th Conference on Open Research Areas in Information Retrieval (pp. 189–196). Le Centre de Hautes Etudes Internationales D'informatique Documentaire.

Metzler, D., & Cai, C. (2011). USC/ISI at TREC 2011: Microblog track. In TREC.

Miyanishi, T., Okamura, N., Liu, s., Seki, K., & Uehara, K. (2011). TREC 2011 microblog track experiments at Kobe University. In TREC.

Na, S.H., Lee, Y., Nam, S.H., & Lee, J.H. (2009). Improving opinion retrieval based on query-specific sentiment lexicon. In Proceedings of the 31st European Conference on IR Research on Advances in Information Retrieval. ECIR '09 (pp. 734–738). Berlin, Heidelberg: Springer.

Naveed, N., Gottron, T., Kunegis, J., & Alhadi, A.C. (2011). Searching microblogs: Coping with sparsity and document quality. In Proceedings of the 20th ACM International Conference on Information and Knowledge Management (pp. 183–188). New York: ACM.

O'Connor, B., Krieger, M., & Ahn, D. (2010). Tweetmotif: Exploratory search and topic summarization for twitter. Proceedings of ICWSM.

Ounis, I., Lin, J., & Soboroff, I. (2011). Overview of the trec-2011 microblog track. In TREC.

Ounis, I., Macdonald, C., & Soboroff, I. (2008). Overview of the trec 2008 blog track. In TREC.

Ounis, I., Macdonald, C., de Rijke, M., Mishne, G., & Soboroff, I. (2006). Overview of the trec 2006 blog track. In TREC.

Owoputi, O., O'Connor, B., Dyer, C., Gimpel, K., Schneider, N., & Smith, N.A. (2013). Improved part-of-speech tagging for online conversational text with word clusters. In Proceedings of NAACL-HLT (pp. 380–390).

Peng, H., Long, F., & Ding, C. (2005). Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. IEEE Transactions on Pattern Analysis and Machine Intelligence, 27(8), 1226–1238.

Ravikumar, S., Talamadupula, K., Balakrishnan, R., & Kambhampati, S. (2013). Raprop: Ranking Tweets by exploiting the Tweet/user/web ecosystem and inter-Tweet agreement. In Proceedings of the 22nd ACM International Conference on Information & Knowledge Management (pp. 2345–2350). New York: ACM.

Ritter, A., Clark, S., Mausam, Etzioni, O. (2011). Named entity recognition in Tweets: an experimental study. In Proceedings of the conference on empirical methods in natural language processing (pp. 1524–1534). Stroudsburg, PA: Association for Computational Linguistics.

Robertson, S.E., Walker, S., Jones, S., Hancock-Beaulieu, M.M., Gatford, M. (1995). Okapi at trec-3. NIST Special Publication.

Salton, G., Wong, A., & Yang, C.S. (1975). A vector space model for automatic indexing. Communications of the ACM, 18(11), 613–620.

Seki, K., & Uehara, K. (2009). Adaptive subjective triggers for opinionated document retrieval. In Proceedings of the 2nd ACM International Conference on Web Search and Data Mining. WSDM '09 (pp. 25–33). New York: ACM.

Teevan, J., Ramage, D., & Morris, M.R. (2011). # twittersearch: A comparison of microblog search and web search. In Proceedings of the 4th ACM International Conference on Web Search and Data Mining (pp. 35–44). New York: ACM.

Yang, L., Sun, T., Zhang, M., & Mei, Q. (2012). We know what@ you# tag: does the dual role affect hashtag adoption? In Proceedings of the 21st International Conference on World Wide Web (pp. 261–270). New York: ACM.

Zhang, M., & Ye, X. (2008). A generation model to unify topic relevance and lexicon-based sentiment for opinion retrieval. In Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. SIGIR '08 (pp. 411–418). New York: ACM.

Zhang, W., Yu, C., & Meng, W. (2007). Opinion retrieval from blogs. In Proceedings of the 16th ACM Conference on Information and Knowledge Management. CIKM '07 (pp. 831–840). New York: ACM.

Zhang, X., He, B., & Luo, T. (2012). Transductive learning for real-time twitter search. In ICWSM.

Zhang, X., He, B., Luo, T., & Li, B. (2012). Query-biased learning to rank for real-time twitter search. In Proceedings of the 21st ACM International Conference on Information and Knowledge Management (pp. 1915–1919). New York: ACM.

DOI: 10.1002/asi