Improving Twitter Retrieval by Exploiting Structural Information

Zhunchen Luo[†*], Miles Osborne[‡], Saša Petrović[‡] and Ting Wang[†]
[†]College of Computer, National University of Defense Technology
410073 Changsha, Hunan, CHINA
[‡]School of Informatics, The University of Edinburgh
EH8 9AB, Edinburgh, UK
zhunchenluo@nudt.edu.cn, miles@inf.ed.ac.uk
sasa.petrovic@ed.ac.uk, tingwang@nudt.edu.cn

Abstract

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 combination of different blocks of texts. These blocks include plain texts, hashtags, links, mentions, etc. Each block encodes a variety of communicative intent and sequence of these blocks captures changing discourse. Previous work shows that exploiting the structural information can improve the structured document (e.g., web pages) retrieval. In this paper we utilize the structure of tweets, induced by these blocks, 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 upon social media features, but when we do add this additional information, performance improves significantly.

Introduction

The large volume of real-time tweets posted on Twitter per day are highly attractive for information retrieval purpose. However, existing Twitter search systems simply treat the text of a tweet as a unit of plain text when modeling relevance (Efron 2010; Duan et al. 2010; Massoudi et al. 2011; Naveed et al. 2011). Previous work shows that web pages and normal text documents can be sub-divided into non-overlapping 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 (Callan 1994; Ahnizeret et al. 2004; 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 plain text, BBC News often end 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, etc, 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 impose 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.

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. This structural information is used in a learning-to-rank approach for Twitter search. In this paper, a set of features are derived from TBBs and structures which is expected to improve the performance of tweet 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 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.

The contributions of this paper can be summarized as follows:

1. We propose *Twitter Building Blocks* (TBBs), which capture 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 of Twitter retrieval that are very similar to a state-of-the-art system which uses social media features. Specially, we do not need to use those social media features.

[*]This research was carried out while Zhunchen Luo was a visiting international student at the University of Edinburgh supported by a CSC scholarship.

[1]Yao Ming is a retired Chinese professional basketball player in NBA. BBC News is a web news gathering and broadcasting of news and current affairs. Lady Gaga is an American pop singer.

Figure 1: Examples of Tweets by Yao Ming, BBC News and Lady Gaga

3. We also demonstrate that social media features are additive to our approach yielding further gain.

Related Work

We review related works on two main areas: structured document retrieval and tweet retrieval.

Structured Document Retrieval

Structured document retrieval attempts to exploit structural information by retrieving documents based on combined structure and content information. This is based on the idea that the same term in different blocks has its own importance factor for ranking and that certain structural combination of blocks have specific informational characteristics. Ahnizer et al. (2004) use 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 show that such structure is useful for data-intensive web sites, which are subjected to frequent content updates, e.g., digital libraries, web forums, news web sites etc. Fernandes et al. (2007) and Moura et al. (2010) use the block structure of a web page to improve ranking results. They propose approaches based on automatically computed block-weight factors. Cai et al. (2004) propose a method for taking advantage of the segmentation of web pages into blocks for search task. All of studies assume a same term can behave differently in particular blocks.

Tweet Retrieval

O'Connor et al. (2010) present *TweetMotif*, an exploratory search application for Twitter. Their work mainly focuses on topic discovery and summary of results. Efron (2010) proposes a language modelling approach for hashtag retrieval. He uses the retrieved hashtags on a topic of interest for query expansion to improve the performance of Twitter search. Massoudi et al. (2011) study a new retrieval model for Twitter search by considering the model with textual quality and Twitter specific quality indicators. They find that this model has a significant positive impact on tweet retrieval. Naveed et al. (2011) combine document length normalization in a retrieval model to resolve the sparsity of short texts for tweets. The relatedness of a tweet to a query depends on many factors. All the above approaches do not consider any information about the structure of tweets. Duan et al. (2010) consider learning-to-rank for tweets. They propose a new ranking strategy which uses 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. They do not utilize any structural information for tweet retrieval.

Twitter Building Block

A tweet 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 *Twitter Building Blocks* (TBBs). Various combination of these TBBs give different tweet structures (TBB structures).

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 though 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 the tweet.

MET: To indicate another user(s) (e.g., *@ladygaga*) as the recipient(s) of tweet.

RWT: To indicate copying and rebroadcasting of the original tweet (e.g., *RT @ladygaga*).

URL: Links to outside contents (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 the tweet.

Figure 2 shows two tweets which 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 labelled the topic of the tweet, that can help readers to better understand the original message (MSG).

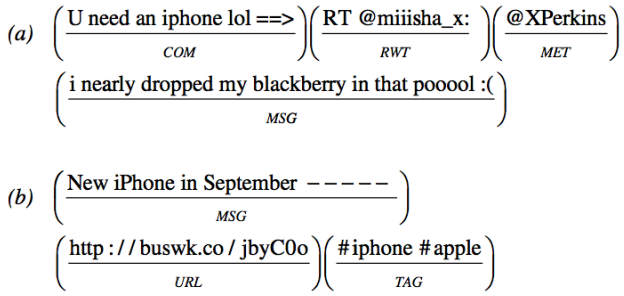

Figure 2: Tweets with Gold TBB Annotation

In order to understand how people use these building blocks, we randomly collected 2,000 samples of English[2] tweets, automatically tokenized them using a tokenizer from O'Connor et al. (2010)[3] and then manually tagged their TBBs and structures. Table 1 shows the distribution of different TBB structures in these tweets. The fourteen 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 sub-tasks: 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 sub-tasks and use an IOB-type labeling scheme.

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

[3]`http://github.com.brendano/tweetmitif`

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		

Table 1: Distribution of TBB Structures

Given a tweet as input, the expected output is a sequence of blocks $B_1B_2...B_m$. Every B_i is a sequence of consecutive tokens $t_{i1}t_{i2}...t_{in}$. Each token t_{ij} 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 t_{ij} in block B_i has the same *X* value. *"Y = B"* only labels the tokens t_{ij} *(j=1)* and *"Y = I"* labels other tokens t_{ij} *(j>1)*. 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, and 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.

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 a "#".
- Every token begin with "www.", "http:" or end with ".com" is a URL tag.
- The sequence of tokens, which its pattern is "@username :" or "@username", are MET tags.
- The sequence of tokens, of the form "RT @username :", "RT @username", "RT" or "via @username", are RWT tags.
- The preceding of "RT @username" and the succeeding of "via @username" or "«" is a COM tag.

We manually tagged 2000 tweets for training and testing. We randomly divided the data into a training set of 1000 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 which 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 labelled by human and opinion mining is always a challenging task in NLP. The tag "URL_I" also has low F1 value. The reason

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

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

is that some of links has been wrongly tokenized by Twitter tokenizer (O'Connor et al. 2010). However the effect is insignificant since the number of "URL_I" tag is small. From these labelled tokens, the boundaries of TBBs and the structure of a tweet are identified. TBB structure identification can achieve an accuracy of 82.60%.

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

Table 2: Automatic TBB Tagger Result

TBB Analysis

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

It's 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.

Twitter provides a large volumes of data in real time. The textual quality of the tweets, however, varies significantly ranging from news-like text to meaningless strings (Han and 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.

We randomly collected 10,000 English tweets for each TBB structure through Automatic TBB Tagger labelling and calculated their *Out of Vocabulary* (OOV) value. The OOV value is the number of words out of vocabulary, which only appear in "MSG" and "COM" blocks, divided by the total number words. This is used to roughly approximate the language quality of text. 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 words out of vocabulary in tweets are misspelt words or abbreviations. Table 3 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 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 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		

Table 3: OOV Values about TBB Structures

TBB for Learning to Rank Tweets

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

Learning to Rank Framework

Learning to rank is a data driven approach which effectively incorporates a bag of features in a model for ranking task. Every tweet is manually tagged whether it is a relevant tweet in training data. A bag of features related to the relevance of tweet is extracted. RankSVM (Thorsten Joachims, 1999) is used to train a ranking model. The ranking performance of model using a particular in testing data reflects the effect of that features on tweet retrieval.

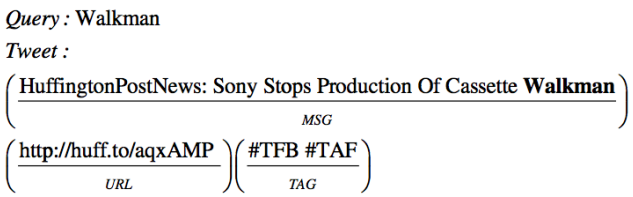

Figure 3: A Query and A Result Tweet

TBB features for Learning to Rank

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.

TBB Structure Type: Each tweet has a unique TBB structure. We represent a tweet as a fifteen-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. E.g., the tweet's TBB structure is "MSG URL TAG" in Figure 3, 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. E.g., in Figure 3 the query "Walkman" is inside of "MSG" block.

Neighbour 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. E.g., the succeeding of TBB containing query "Walkman" for tweet in Figure 3 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. E.g., only one TBB contains the query "Walkman" for tweet in Figure 3.

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 proportion of words in the TBBs containing the query which 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.

	Number
Trending Topic	34
Science & Technology	27
News	17
Location	8
Entertainment	8
Others	6

Table 4: Different Classes of Queries Statistics

Average query length (words)	1.48
Average number of results per query	9.36
Total relevant tweets	184
Total non-relevant tweets	752

Table 5: Annotated Data Statistics

Dataset for Ranking

We crawled and indexed about 800,000 tweets using the Twitter streaming API every day and implemented a web search interface. Three annotators, who are all computer science researchers, were asked to use our search engine from October 4th 2010 to October 28th 2010. They were allowed to post any queries. Given a query, the search engine presented a list of n tweets ranked by BM25 score. The value n was chosen by the annotators with the default being10. In addition to the text of the tweet, annotators were also shown the tweet's timestamp and author name. The annotators assigned a binary label (relevant to the query or not) to every presented tweet. We collected 100 queries and their relevance judgments this way. We analysed these 100 queries which can be categorized into 6 classes as shown in Table 4. One important class is *Trending Topic*, e.g., "Chilean miner". There are many queries related to science and technology e.g., "java flaw". *News* are queries related news and hot topics, e.g., "wikileaks". The annotators post *Location* queries to retrieve news happening in these locations. *Entertainment* are queries related to films and celebrities. Other queries are grouped into *Others*. Summary statistics of the data are listed in Table 5.

Experiment

Ten-fold cross-validation was used in our experiments and we use Mean Average Precision (MAP) as the evaluation metric. We take the approach of Duan et al. (2010) as the baseline which gives the best published results on the Twitter search task[6]. We also develop some social media features for tweets ranking, called SM_Rank. The features of the baseline and SM_Rank are listed in Table 6. 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 which called Base-

[6]We do not take the method based on BM25 score as a baseline, since 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.

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 6: Baseline and Social Media Features

	MAP
Baseline	0.4197
SM_Rank	0.4338
TBB_Rank	0.4235
Baseline+SM_Rank	0.4546
Baseline+TBB_Rank	0.4326
SM+TBB_Rank	$0.4710^{*\dagger}$
Baseline+SM+TBB_Rank	$0.4712^{*\dagger}$

Table 7: Performance of Ranking Methods. A star(*) and dagger(†) indicate statistically improvement over the Baseline and SM_Rank respectively.

line+SM_Rank, Baseline+TBB_Rank, SM+TBB_Rank and Baseline+SM+TBB_Rank respectively.

Table 7 shows the performance of these methods. We can see that just using the TBB features, which derived from the tweet text itself, can achieve comparable performance as the Baseline and SM_Rank methods, which utilize the social media information. We conducted a paired t-test between the results of these three methods and found no statistically significant difference (at $p = 0.05$). By adding social media features to TBB_Rank we get a significant improvement in MAP (at $p = 0.05$). Combination of all three sets of features provide the highest MAP value. All the results suggest that structural information of tweets can improve search.

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 8. We evaluate the effect of each feature by replacing the Link feature in the Baseline method for tweets ranking. Table 9 gives the performance of each feature related TBBs structure containing the "URL" block.

We can see from Table 9 that only MSG URL ranking method gives comparable performance as the Baseline (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 non-relevant 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 non-relevant 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

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 8: Features in TBB Structure Type related the "URL" block

	MAP
Baseline	0.4197
MSG URL	0.4019
MSG URL TAG	0.3327
RWT MSG URL	0.3289
TAG MSG URL	0.3245
COM URL	0.3191
OTHER URL	0.1984
MET MSG URL	0.1932

Table 9: Performance of Each Feature Related TBB Structure Containing URL

can capture more information about the tweets containing links which can improve tweets ranking.

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, e.g., their out-of-vocabulary (OOV) values are very different. We used this structural information as features into a learning-to-rank scenario 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. Furthermore, using the TBB features together with other social media features can achieve higher performance. This shows that although the texts of tweets are very short, their structural information can improve Twitter retrieval.

For future work we plan to use the TBB in other Twitter applications, e.g., users clustering, spam filtering, etc.

Acknowledgements

We would like to thank Chee-Ming Ting, Desmond Elliott and Diego Frassinelli for their comments. This research is supported by the National Natural Science Foundation of China (Grant No. 61170156 and 60933005) and the National Grand Fundamental Research 973 Program of China (Grant No. 2011CB302603).

References

Ahnizeret, K.; Fernandes, D.; Cavalcanti, J.; deMoura, E.; and da Silva, A. 2004. Information Retrieval Aware Web Site Modelling and Generation. In *Proceedings of the 23th Internacional Conference on Conceptual Modeling*, pages 402-419.

Han, B., and 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*, pages 368-378.

Cai, D.; Yu, S.; Wen,J.; and Ma, W. 2004. Block-based Web Search. In *Proceedings of SIGIR' 04*, pages 456-463.

Callan, J. 1994. Passage-level evidence in document retrieval. In *Proceedings of SIGIR' 94*, pages 302-310.

Duan, Y.; Jiang, L.; Qin, T.; Zhou, M.; and Shum, H. 2010. An empirical study on learning-to-rank tweets. In *COLING '10*, August.

Efron, M. 2010. Hashtag retrieval in a microblogging environment. In *SIGIR' 10*, pages 787- 788.

Fernandes, D.; de Moura, E. S.; Ribeiro-Neto, B. A.; da Silva,A. S.; and Gonçalves, M. A. 2007. Computing block importance for searching on web sites. In *Proceedings of the 16th ACM Conference on Information and Knowledge Management*, pages 165-1740.

Joachims, T. 1999. Making Large-Scale SVM Learning Practical. In *Advances in Kernel Methods: Support Vector Learning*, pages: 169-184.

Lafferty, J.; McCallum, A.; and Pereira, F. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proc. of ICML*.

Massoudi, K.; Tsagkias, M.; de Rijke, M.; and Weerkamp, W. 2011. Incorporating Query Expansion and Quality Indicators in Searching Microblog Posts. In *Proc. of ECIR*.

de Moura, E. S.; Fernandes, D.; Ribeiro-Neto, B. A.; da Silva,A. S.; and Gonçalves, M. A. 2010. Using structural information to improve search in web collections. In *JASIST*, 61:2503–2513, December.

Naveed, N.; Gottron, N.; Kunegis, J.; and Alhadi, A.C. 2011. Searching Microblogs: Coping with Sparsity and Document Quality. In *Proceedings of the 20th ACM Conference on Information and Knowledge Management*.

O'Connor, B.; Krieger, M.; and Ahn, D. 2010. TweetMotif: Exploratory Search and Topic Summarization for Twitter. In *Proceedings of the International AAAI Conference on Weblogs and Social Media*, Washington, DC, May.