Report on Time is of the Essence: Improving Recency Ranking Using Twitter Data

Luciano Del Corro Monika Mitrevska

December 20, 2010

1 Introduction

Recency content has become a critical issue in information retrieval. Efficient retrieval of fresh and relevant documents has not been fully overcome yet and it is an increasing topic of interest in academic and commercial research. So far, web search engines manage to deal reasonable well with the classical Navigational, Transactional and Informational queries. However with a more granular category like Recency Sensitive Queries, those referred to queries where the user expects documents that are both relevant and fresh, there is still a lack of a consistent and efficient method of retrieval.

Given the particular characteristics of the different types of queries, it seems sensible the idea that each one of them may require a different ranking system. Specifically, failures for time sensitive queries can be more severe than for the others. First, they are more likely to suffer from the zero recall problem which is associated to the failure of the search engine to index any relevant document for a given query. This is a terminal problem since no reformulation or scanning can expose the user to the relevant document. There is no amount of effort that can lead the user to a successful outcome.

The second issue related to failures in time sensitive queries is associated with the fact that the need for relevant content is usually immediate. The relevant documents could be ranked in a very low position so that the probability of finding them may be close to zero.

To sum up we face a dual problem. First the need of a method to quickly crawl fresh and relevant documents and second, the design of a ranking system so they can be efficiently retrieved.

In this context it seems reasonable to believe that different types of queries should be addressed with a specific ranking system according to its particular characteristics. This paper attempts to develop a method exploiting information from micro-blogs to address those two problems inherent to the so called time sensitive queries.

2 Current approaches to Recency Sensitive Queries

The authors sustain that current approaches do not successfully achieve an efficient ranking for recency content. The main drawbacks are related to the tendency to identify time sensitive documents just with news. The main search engines tend to integrate content from a specialized news index in which each news site is given an authority which is in turn used to determine the authority of a news article.

However, this approach according to the authors, suffers from several drawbacks. First, relevant documents for sensitive queries are web pages, not just news. In addition, a relevant document could be published in a low priority site and therefore its relevance would be underestimated. This leads to think that addressing recency ranking should focus on web results directly and not in a specific type of document.

Nevertheless, fresh documents should also not be treated as normal web pages. On one hand, standard crawling may be too slow to deal with freshness. On the other, query-independent signals may be useless since some features may not be accurately represented.

Regarding the first point and supporting the authors arguments it should be said that the whole point of time sensitive queries is that the need for fresh content is immediate. That is why a normal crawler can be too late to index the webpage and this may in turn not be fresh anymore. The relevance of recency content is intrinsically attached to the fact that it is fresh.

The second point is even more evident since those features non-related to the content of the documents will be missing or undervalued such as aggregate clicks or in-links. There is not enough time to accumulate such information about the document. It should be clear that any attempt to deal with recency content cannot be based on information that gets significant time to gather.

Given this facts it is reasonable to think that the focus should be not only in a different ranking strategy but also in a different type of data to address freshness. The paper argues that information from micro-blogging can be exploited to improve search engine performance for recency sensitive queries.

3 Micro-blogging data

According to the paper micro-blogging data has a couple of desirable characteristics that can be efficiently exploited to address time sensitive queries.

Given its constrained size (usually no more than 140 characters) and the possibility to rapidly post from a variety of interfaces, post in micro blogs tend to be generated in a high quantity. Even more, those real time updates tend to be related to developing topics like on-going events which is reinforced by the fact that mobile devices can be used to perform the updates which potentially converts micro-bloggers in non-professional reporters.

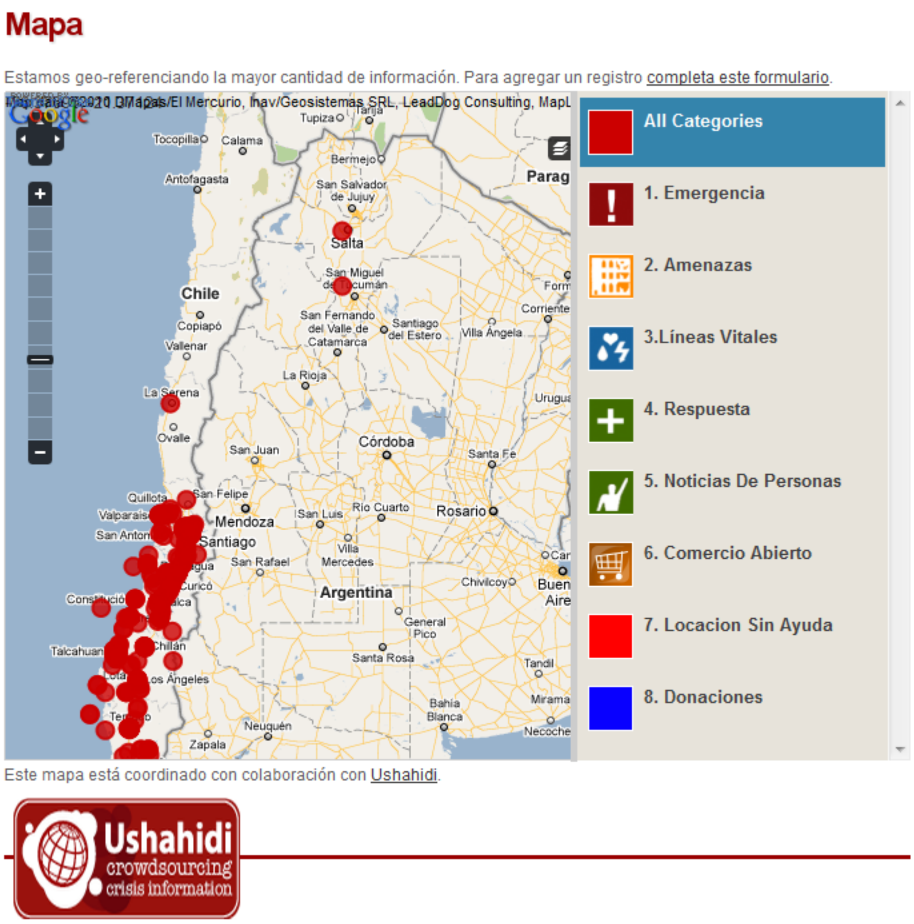
This argument from the paper is totally justified by the additional evidence that we provide. It is interesting to observe the data provided by Usahidi (Figure 1) which shows the SMS activity during the earthquake in Chile this year. The activity is absolutely correlated with the most affected areas by the earthquake as we can see in the map.

This also leads to the fact that as the posts are posted according to diverse and dynamic browsing priorities, the priority between different posts is transparent and it is revealed by the action of posting itself. The authors sensibly remark that using the links posted in micro blogs avoid the need of predicting priority.

At the same time micro-blogging platforms provide a rich linking structure for the post encouraging the inclusion of links which can contain any type of documents (news or not) which as we will see later are usually fresh.

One final interesting property of micro-blogging data is that the different platforms usually provide a transparent topography of the social network structure. Social networks, which are easy to modelize (typically through a directed graph), can be very useful in terms of the identification of the more popular or trusted individuals which in turn can lead us to derive some interesting properties of the links posted by those individuals. This last argument is a point of debate that will be discussed later.

Figure 1: SMS activity during earthquake in Chile



4 Why Twitter?

The reasons that led the authors to choose twitter are clear and reasonable: Twitter is the rising star in Internet, its popularity has been growing at an amazing path as we can see in the table 1 and the graph (Figure 2) below which we collected.

In addition, in the past few years, the amount of research papers about Twitter has been growing steadily. There is enough evidence to support the connection between twitter and recency content (Java et al., 2007; Hughes and Palen, 2009) so that Twitter seems to be a prudent choice.

Finally, it must be said that Twitter has all the interesting properties described above as public content and social network topology, a public timeline

Figure 2: Tweets per day evolution

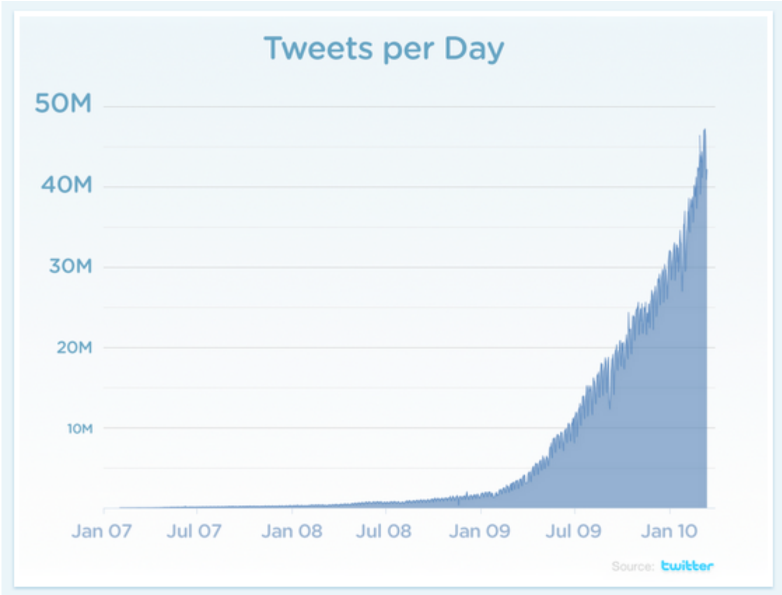


Table 1: Selected Twitter Statistics (Jan. 2010)

More than 106MM users
300,000 new users per day
180MM of unique visitors
75% of traffic from external sources
Search engine receives 600MM queries per day
37% use the phone to tweet

of updates and a wide range of possibilities to interact with the system (e.g: web, SMS, IM Agent).

5 Overview of the method: How to use microblogging data?

The paper performs a very simple and intuitive two phase method. It consists on first crawling and filtering URLs from Twitter and then incorporating those URLs into a general web search system.

5.1 Crawling Twitter URLs

The main and valid concern at this point is the tendency of twitter post to include a significant amount of links to spam, adult and self promotion pages which are usually undesirable. This of course requires some type of filtering heuristics in order to select only those interesting URLs and avoid damages to the quality of the data.

Given the importance of this task and the need to keep the paper focus on the main issue of the connection between Recency content and micro blogging data, the authors decided to take the simplest approach which at the same time provides a high level of confidence: over-filtering.

The filtering heuristics tries to get rid of a large amount of links attempting at the same time keep a reasonable amount to perform the experiment. It consists of two simple rules: First discharge those URLs referred to by the same user more than two times and the second, discharge those URLs only referred to by just one user.

The first one is saying that if a user is insisting with the same URL more than one time this is likely to be spam and the second is saying that URLs that are interesting for just one person are also discarded.

In a short period of time the authors could gather a little bit more than one million URLs. After the first rule 66.7% remained and after the second just 5.9% remained.

At this point it is interesting to say that it was obvious that the authors tried to keep the confidence level high even at the cost of losing efficiency. However, it is worth to mention that a more complex filtering system should easily allow a significant increment in the proportion of non-spam URLs so the crawling capacity over twitter URLs can be further improved.

Another interesting aspect is that compared to a crawler of a commercial web search engine 90% of the URLs crawled by the authors from Twitter had not been crawled by the search engine. This is important since it gives some evidence that crawling micro-blogging data may improve the collection of fresh URLs.

However, regarding this last issue it must be said that the paper mentions that the overhead of crawling Twitter URLs is quite significant. They specifically say that realtime crawling and indexing of all Twitter URLs may require considerable overhead. Before any application could be successfully implemented this question must be carefully addressed.

5.2 Ranking System

5.2.1 Learning to Rank

Learning to Rank or Machine Learning Ranking (MLR) is the task of constructing a function to rank a set of documents according to the interplay between a given query and a set of properties of the document based on the content or some other desired elements.

MLR usually has two properties: It is feature based and performs discriminative training. When developing a ranking system there must be defined a set of features each of them representing a characteristic of the document that it is supposed to influence its relevance. The system will ultimately learn how, according to the underlying model, those features influence the relevance of the document.

In addition, learning to rank is based on a discriminative model where the input, output and hypothesis spaces need to be defined as well as a loss function. There is a wide range of algorithms available to perform the ranking which are shown in the figure below (Figure 3). There are three main categories of learning algorithms: those that use a Pointwise Approach, those under the Pairwise Approach and finally the ones on the Listwise Approach.

Figure 3: MLR Algorithm categories (Tie-Yan Liu, 2009)

Category	Algorithms
Pointwise Approach	**Regression**: Least Square Retrieval Function (TOIS 1989), Regression Tree for Ordinal Class Prediction (Fundamenta Informaticae, 2000), Subset Ranking using Regression (COLT 2006), ... **Classification**: Discriminative model for IR (SIGIR 2004), McRank (NIPS 2007), ... **Ordinal regression**: Pranking (NIPS 2002), OAP-BPM (EMCL 2003), Ranking with Large Margin Principles (NIPS 2002), Constraint Ordinal Regression (ICML 2005), ...
Pairwise Approach	Learning to Retrieve Information (SCC 1995), Learning to Order Things (NIPS 1998), Ranking SVM (ICANN 1999), RankBoost (JMLR 2003), LDM (SIGIR 2005), RankNet (ICML 2005), Frank (SIGIR 2007), MHR(SIGIR 2007), GBRank (SIGIR 2007), QBRank (NIPS 2007), MPRank (ICML 2007), IRSVM (SIGIR 2006), ...
Listwise Approach	**Listwise loss minimization**: RankCosine (IP&M 2008), ListNet (ICML 2007), ListMLE (ICML 2008), ... **Direct optimization of IR measure**: LambdaRank (NIPS 2006), AdaRank (SIGIR 2007), SVM-MAP (SIGIR 2007), SoftRank (LR4IR 2007), GPRank (LR4IR 2007), CCA (SIGIR 2007), ...

One important thing that has to be bear in mind when choosing a ranking model is that usually just the top positions matter. The probability that a user goes farther down in the retrieved ranking is almost zero.

Two usual questions that remain on the background of any ranking method are those referred to the updates that at some point will need to be performed and the fact that calibrating parameters is still a tedious task. So far these two issues are largely problem dependent.

To sum up the result of a learning to rank algorithm is a function that ranks a set of documents according to their relevance to a given query. The function is trained according to some editorially labeled data

5.2.2 Ranking Twitter URLs

As it was discussed a crucial issue to build a learning to rank system is to define the features representing some relevant aspect of a given document. When dealing with normal URLs features can be grouped into Content and Aggregate Features.

Content Features are those related to the content of the document. It is sensible to think that these features will be unaffected if the document is fresh or not. However, Aggregate Features, those trying to measure the long term popularity of a given document like in-links or aggregate clicks will be totally underrepresented for a fresh document.

This is the reason why the authors consider that these last set of features are not useful in dealing with Recency content and in consequence they developed a new set of specific features specially designed to rank specifically those links gathered from Twitter.

Twitter Features Twitter Features are based on the contextual information of a post containing a URL and in the social networks topology of twitter.

The paper defines three basic features based on the text surrounding the URL posted in Twitter.

The first one is a similarity measure which tries to capture the idea that if there is more overlaping words between the post and the query the URL should be better ranked. It is defined as

$$\phi^j_{\text{cosine}} = \frac{\mathbf{u}_j^\mathsf{T}\mathbf{q}}{\|\mathbf{u}_j\|_2\|\mathbf{q}\|_2}$$

where q is the query vector and

$$\mathbf{u}_j^\mathsf{T} = \sum_i M_{ij} D_{i\cdot}$$

is a vector containing the words associated to a given URL where $Mm*w$ is the matrix with m tweets and w URLs and $Dm*v$ is a matrix with m tweets and v words.

Then they defined

$$\begin{aligned} \omega_{iq} &= (\tilde{D}_{i\cdot})^\mathsf{T}\tilde{\mathbf{q}} && \text{overlapping terms} \\ \epsilon_{iq} &= \|\tilde{D}_{i\cdot}\|_1 - \omega_{iq} && \text{extra terms} \\ \mu_{iq} &= \|\mathbf{q}\|_1 - \omega_{iq} && \text{missing terms} \end{aligned}$$

which can be used to define the following similarity measure where unmatched terms are most severely penalized.

$$\phi^j_{\text{unit}} = \frac{1}{\|\mathbf{q}\|_1}\sum_{i=1}^{m} \epsilon^\alpha_{iq}\mu^\beta_{iq}\omega_{iq}M_{ij}$$

With *alpha* and *beta* controlling the relative importance of the extra and the missing terms.

Finally a feature related to the exact matches between continuous ordered terms in the query and in the tweet can be defined as

$$\phi^j_{\text{exact}} = \frac{1}{\|M_{\cdot j}\|_1}\sum_{i=1}^{m} \text{phraseMatch}(q,i)M_{ij}$$

In terms of the features based on the social network the authors designed some features that use the popularity of a user as a proxy of the authority of the page that the user posted. In this sense the popularity is equated to credibility. It is, a measure of how trustable a given person is. A user with more followers will be more trustable so the links posted by him will get higher rank.

To do this the authors adopted a Page Rank approach to calculate the eigenvector of the adjacency matrix which is a measure of the centrality of the users in the graph.

$$\pi_{t+1} = (\lambda \mathbf{W} + (1-\lambda)\mathbf{U})\pi_t$$

Then the value corresponding to the user is used as the authority of the web page he posted.

$$\phi^j_{\text{authority}} = \pi_i$$

This can lead us to a feature combining content and social network importance by weighting the unit feature by the authority of the person who posted the URL.

$$\phi^j_{\text{unit-}\pi} = \frac{1}{\|\mathbf{q}\|_1} \sum_{i=1}^{m} \epsilon^{\alpha}_{iq} \mu^{\beta}_{iq} \omega_{iq} M_{ij} \phi^i_{\text{authority}}$$

This approach may raise some concerns about the convenience of equating popularity to credibility. Regarding this aspect the authors say that even though it is true that the most followed users were celebrities a set of desirable sources of information as some scientific institutions, governmental agencies or news agencies were also reasonably popular.

In this direction it is sensible to assume that none of these institutions will be posting random documents non-related to their main activities. It follows than popularity may be in these cases a good proxy of the authority of the pages.

However, as it was previously said the users with more followers were mainly celebrities. These persons may perfectly post a random page which will be surely heavily replicated by the fans for no other reason that the popularity of the celebrity and in consequence that page will receive a high authority. If this last phenomenon is strong enough some results may be perfectly biased.

The following table displays the most popular users at the time when the paper was written and as we can see the first positions are mainly celebrities.

userID	User/Type
twitter	Twitter Official
kimkardashian	Kim Kardashian
aplusk	Ashton Kutcher
denise_richards	Denise Richards
ddlovato	Demetria Lovato
katyperry	Katy Perry
khloekardashian	Khloe Kardashian
johncmayer	John Mayer
astro_mike	Mike Massimino
robdyrdek	Rob Dyrdek
...	...
nasa	NASA Space Program
mcuban	Mark Cuban
wired	Wired Magazine
problogger	Darren Rowse
chrispirillo	Chris Pirillo
cbsnews	CBS News
jkottke	Jason Kottke

In addition the authors developed a set of features, described in Figure 4, that can be grouped into three main categories. From 1 to 6 we can see features related the users that issue the URL. From 7 to 12 those related with the person who first issued the URL and finally from 13 to 18 those related with the person with the highest score who issued the URL.

Relevance Model As a Ranking function the paper uses the Gradient Boosted Decision Tree (GBDT). It consists of a stagewise additive expansions plus a steepest descent minimization. It consists of a function represented as a combination of decision trees.

It is not clear why they choose to use this method. Probably the authors thought this part of the paper as a black box, not playing any particular role in the approach they selected. It is, the method should be regarded as not having any influence in the results.

Given this fact it is really difficult to have a clear idea of how the ranking function was implemented. Moreover, there may be an error in the func-

Figure 4: Additional Twitter Features

$\phi_{\text{other-1}}$	average number of followers for the users who issued the tiny URL
$\phi_{\text{other-2}}$	average post number for the users who issued the tiny URL
$\phi_{\text{other-3}}$	average number of users who retweeted the tweets containing the tiny URL
$\phi_{\text{other-4}}$	average number of users who replied those users that issued the tiny URL
$\phi_{\text{other-5}}$	average number of followings for the users who issued the tiny URL
$\phi_{\text{other-6}}$	average Twitter score of all the users who issued the tiny URL
$\phi_{\text{other-7}}$	number of followers for the user who first issued the tiny URL
$\phi_{\text{other-8}}$	number of posts by the user who first issued the tiny URL
$\phi_{\text{other-9}}$	number of users who retweeted the user who first issued the tiny URL
$\phi_{\text{other-10}}$	number of users who replied the user who first issued the tiny URL
$\phi_{\text{other-11}}$	number of followings for the user who first issued the tiny URL
$\phi_{\text{other-12}}$	Twitter score of the users who first issued the tiny URL
$\phi_{\text{other-13}}$	number of followers for the user who issued the tiny URL with the highest Twitter score
$\phi_{\text{other-14}}$	number of posts by the user who issued the tiny URL with the highest Twitter score
$\phi_{\text{other-15}}$	number of users who retweeted the user who issued the tiny URL and has the highest Twitter score
$\phi_{\text{other-16}}$	number of users who replied the user who issued the tiny URL and has the highest Twitter score
$\phi_{\text{other-17}}$	number of followings for the user who has the highest Twitter score among the users that issued the tiny URL
$\phi_{\text{other-18}}$	Twitter score of the users who issued the tiny URL and who is the highest Twitter score
$\phi_{\text{other-19}}$	number of different users who sent the tiny URL.

tion below displayed in the paper which not fully matches with the original paper on GBDT of Friedman (2001). Specifically the first term T_t in the function should be T_0, but again this is unclear since they do not give much specifications about the relevance model.

$$f_t(x) = T_t(x;\Theta) + \lambda \sum_{t=1}^{T} \beta_t T_t(x;\Theta_t)$$

The objective of the model is to train two different sets of URLs (Regular and Twitter) that need to be blended later. Thats why it is important that the output of the function is a classification and not a real value. The used classifications are Perfect, Excellent, Good, Fair or Bad. In this way the sets of regular and twitter URLs can be trained and classified independently from each other with different set of features.

The function was trained using sample query-pairs, including both regular and Twitter URLs, each with a label. Then the function was applied to test queries.

6 Experiments

The paper compares the following 5 models (Figure 5). The first column corresponds to the set of features used in regular URLs and the second to the set of features used in the Twitter URLs. Regular features refer to the set of content and regular features, content refers to content alone, twitter to the content and the new developed twitter features and finally composite uses a content model score as a feature and also twitter features.

Figure 5: Runs used in experimentations

$(\mathcal{M}_{\text{regular}}, \mathcal{M}_{\text{regular}})$	Use $\mathcal{M}_{\text{regular}}$ on regular and Twitter URLs.
$(\mathcal{M}_{\text{content}}, \mathcal{M}_{\text{content}})$	Use $\mathcal{M}_{\text{content}}$ on regular and Twitter URLs.
$(\mathcal{M}_{\text{regular}}, \mathcal{M}_{\text{content}})$	Use $\mathcal{M}_{\text{regular}}$ on regular URLs and $\mathcal{M}_{\text{content}}$ on Twitter URLs.
$(\mathcal{M}_{\text{regular}}, \mathcal{M}_{\text{twitter}})$	Use $\mathcal{M}_{\text{regular}}$ on regular URLs and $\mathcal{M}_{\text{twitter}}$ on Twitter URLs.
$(\mathcal{M}_{\text{regular}}, \mathcal{M}_{\text{composite}})$	Use $\mathcal{M}_{\text{regular}}$ on regular URLs and $\mathcal{M}_{\text{composite}}$ on Twitter URLs.

The first two models use just regular and content features in both sets respectively. The third uses regular features on the first and content features in the second. The fourth one uses regular features in regular URLs and twitter and regular features on the twitter URLs. Finally they try with regular features for regular URLs and a composite of features in the URLs crawled from twitter.

7 Data

As it was said before data collected corresponds to two sets: one consisting of regular URLs crawled by a commercial search engine and another set crawled by the authors from Twitter. A very interesting aspect about the URLs crawled from Twitter that can be seen in the following table is the fact that most of the URLs crawled from Twitter are fresh compared with just 19,4% crawled by the normal crawler (Figure 6).

To highlight the fact that a fresh document is always better in a context of time sensitive queries the authors use a demotion system which reduces the classification of the URLs if they are not fresh. If a web page was somehow outdated they decreased the classification by one grade and if it was totally outdated it was degraded by two grades. This basically tries to reflect the fact that recency content involve both elements: relevance and freshness.

Figure 6: Data distribution in sense of relevance grade and recency label

(a) relevance grade (demoted)

	Perfect	Excellent	Good	Fair	Bad
Regular	0.7%	17.0%	44.9%	26.6%	10.9%
Twitter	13.0%	33.4%	41.0%	20.7%	3.6%

(b) relevance grade (non-demoted)

	Perfect	Excellent	Good	Fair	Bad
Regular	0.9%	23.0%	61.0%	36.1%	14.8%
Twitter	13.0%	33.4%	41.0%	20.7%	3.6%

(c) recency label

	Fresh	Non-fresh
Regular	19.4%	80.6%
Twitter	53.8%	46.2%

8 Evaluation

In addition to the demotion that tries to capture freshness the evaluation method used gives more weight to what happens in the first position of the retrieved ranking.

The following is the demotion version which combines freshness and relevance.

$$\text{NDCG}_n = Z_n \sum_{i=1}^{n} \frac{G_i}{\log_2(i+1)}$$

With G_i being the score and Z_n a normalization factor.

With a function increasing at a decreasing rate in the denominator, a mistake in the first positions is more penalized.

To focus just in freshness independently of relevance the authors presented a second evaluation function

$$\text{DCF}_n = \sum_{i=1}^{n} \frac{F_i}{\log_2(i+1)}$$

where F_i says if the URL is fresh or not.

9 Results

The results support the method developed by the authors. The approach which blends Twitter content into a standard ranked list significantly improves ranking in sense of both relevance and recency. The following table displays the results. We can observe how the approach which uses the twitter features outperforms the rest.

	Top 1						Top 5					
	$\text{NDCG}_{\text{demote},1}$		$\text{NDCG}_{\text{nodemote},1}$		NDCF_1		$\text{NDCG}_{\text{demote},5}$		$\text{NDCG}_{\text{nodemote},5}$		NDCF_5	
$(\mathcal{M}_{\text{regular}}, \mathcal{M}_{\text{regular}})$	0.588		0.611		0.474		0.666		0.681		0.518	
$(\mathcal{M}_{\text{content}}, \mathcal{M}_{\text{content}})$	0.570	-3.2%	0.610	-0.2%	0.513	$+7.5\%$	0.652	-2.1%	0.682	$+0.3\%$	0.587	$+11.7\%$
$(\mathcal{M}_{\text{regular}}, \mathcal{M}_{\text{content}})$	0.600	$+1.8\%$	0.618	$+1.2\%$	0.520	$+8.8\%$	0.680	$+2.1\%$	0.690	$+1.3\%$	0.569	$+8.9\%$
$(\mathcal{M}_{\text{regular}}, \mathcal{M}_{\text{twitter}})$	**0.720**	$+18.4\%$	**0.708**	$+13.7\%$	**0.717**	$+33.8\%$	**0.739**	$+9.9\%$	**0.729**	$+6.5\%$	**0.736**	$+29.6\%$
$(\mathcal{M}_{\text{regular}}, \mathcal{M}_{\text{composite}})$	0.715	$+17.9\%$	0.702	$+13.0\%$	0.747	$+36.5\%$	0.735	$+9.4\%$	0.723	$+5.8\%$	0.756	$+31.4\%$

It is also clear from the results that combining both content and aggregate features for regular URLs is also important; the worst performing model is the one that uses just content features.

Finally the following table shows the importance list of the twitter features. It is interesting to see that not only twitter features are important in general but also that the text corresponding to tweet posting a URL is crucial to determine the relevance of a document.

Figure 7: Most Relevant Twitter Features

Twitter feature	importance rank	importance score
ϕ_{unit}	9	31.1
$\phi_{\text{other-17}}$	10	27.1
$\phi_{\text{other-15}}$	11	26.6
$\phi_{\text{other-3}}$	13	22.8
$\phi_{\text{other-1}}$	18	16.7

This last question may seem quite reasonable since the user posting a URL will need to be extremely specific about the content of the document since the tweet is extremely constrained in size (140 characters).

The second interesting issue that can be observed in the last table is that the social network is also important and worth to be used.

10 Conclusion

The use of micro-blogging data to address time sensitive queries seems to be a reasonable idea. The paper provides enough theoretical fundaments and promising experiments to justify further research in this direction.

However, some points need to be further developed in order to reach a better understanding and a further enhancement of the method.

The first point of concern regarding the use of URLs gathered from twitter is that the social and demographic structure of the social corpus that use Twitter may again be biased if this is not representative enough of the general public using internet. For instance, if we find that just young people use internet the relevance of the pages popular among young will be totally over-represented. The same thing if there is a bias in educational level, economic status, etc. Applying this strategy requires a deeper understanding of the social characteristics of the people using the chosen micro-blogging platform.

Another question that can be mentioned is related to the filtering heuristics applied over the URLs crawled from Twitter. The authors recognized in the discussion that more sophisticated heuristics can probably increase the number of twitter URLs without losing precision. In the experiment just a bit more of 5% of the total URLs originally crawled could be used. This may also be important since it is not clear if micro-blogging data is large enough to provide the necessary redundancies to perform an accurate ranking.

The authors mentioned as well some problems that may arise given the implementation of the blended ranking. They say that it creates a problem for multiple intent queries or queries that need summarization.

Finally we can mention the fact that the relevance model may not be neutral. The authors presented this part as a black box without further discussion. Moreover, the question about the existence of a ranking method particularly suitable to deal with micro-blogging data seems to be a very valid one. Probably future research should be directed to detect a relevance model addressing the problem of ranking diversity.

References

[1] A. Dong, R. Zhang, P. Kolari, J. Bai, F. Diaz, Y. Chang, Z. Zheng, H. Zha. Time is of the Essence: Improving Recency Ranking Using Twitter Data. In Proceedings of the 19th international conference on World wide web, Raleigh, USA, 2010.

[2] J. H. Friedman. Greedy function approximation: A gradient boosting machine. Annals of Statistics, 29(5):11891232, 2001.

[3] A. L. Hughes and L. Palen. Twitter adoption and use in mass convergence and emergency events. In Proceedings of the 6th International Conference on Information Systems for Crisis Response and Management, 2009.

[4] A. Java, X. Song, T. Finin, and B. Tseng. Why we twitter: understanding microblogging usage and communities. In WebKDD/SNA-KDD 07: Proceedings of the 9th WebKDD and 1st SNA-KDD 2007 workshop on Web mining and social network analysis, pages 5665, New York, NY, USA, 2007. ACM.