A Learned Approach for Ranking News in Real-time using the Blogosphere

Richard McCreadie, Craig Macdonald, and Iadh Ounis

School of Computing Science
University of Glasgow, G12 8QQ, UK
{richardm,craigm,ounis}@dcs.gla.ac.uk

Abstract. Newspaper websites and news aggregators rank news stories by their newsworthiness in real-time for display to the user. Recent work has shown that news stories can be ranked automatically in a retrospective manner based upon related discussion within the blogosphere. However, it is as yet undetermined whether blogs are sufficiently fresh to rank stories in real-time. In this paper, we propose a novel learning to rank framework which leverages current blog posts to rank news stories in a real-time manner. We evaluate our proposed learning framework within the context of the TREC Blog track top stories identification task. Our results show that, indeed, the blogosphere can be leveraged for the real-time ranking of news, including for unpredictable events. Our approach improves upon state-of-the-art story ranking approaches, outperforming both the best TREC 2009/2010 systems and its single best performing feature.

1 Introduction

Large quantities of fresh news content from e-news providers are being continually published each day [1]. Meanwhile, millions of users consult e-newspapers and news aggregators to find out the most interesting events and stories occurring worldwide [1]. However, the volume and rate at which news content is currently created, highlights the need for automatic means to sort through this large volume of news in real-time, identifying the most currently newsworthy stories for display. This task can be seen as a ranking problem. For example, on the homepage of a news website, current news stories are ranked by their perceived newsworthiness at that time. Highly newsworthy stories receive prominent placement on the page, while lesser stories are displayed less prominently or not at all.

Recent work examining the automatic ranking of news stories has indicated that related blogging activity can be used as an indicator of story newsworthiness [4, 14]. Indeed, the blogosphere is well known as a medium for news reporting and discussion [13, 20, 21]. Relatedly, almost 20% of searches to a blog search engine were reported to be news-related [17]. This shows that the blogosphere is likely to be a good source of information regarding current news.

News stories can be roughly classified into those resulting from predictable and unpredictable events [2]. Of interest is that only predictable events, exhibit elevated levels of blog posting activity beforehand [14]. For this reason, the majority of previous models for news story ranking have focused on the retrospective ranking of news, i.e. at a later point in time [12]. It is not clear whether the blogosphere will remain an effective source of evidence for ranking news stories when moving to real-time setting. In particular, there may be as yet insufficient blog posts to accurately estimate newsworthiness for stories relating to unpredictable events.

In this paper, we investigate the extent to which it is possible to automatically rank news stories in real-time using the blogosphere. In particular, we propose a novel learning to rank (LTR) [8] approach for this task. LTR techniques are machine learning algorithms which take as input a set of *features* about each object to be ranked, i.e. a story in this case. They learn a weight for each feature, so that by combining weighted features a better overall ranking is produced than when ranking by any single feature alone. In this work, we aim to define suitable features which indicate the current newsworthiness of each story, allowing us to produce an accurate ranking of top news stories in real-time.

The advantages of an LTR approach to this problem in comparison to existing story ranking strategies are two-fold. Firstly, LTR provides a principled means for combining multiple sources of timely story ranking evidence as features. Secondly, LTR is extensible, hence should a new possible feature become available, e.g. the number of clicks on a specific story, then this can be easily integrated. Moreover, to our best knowledge, our framework is the first application of LTR for news story ranking using the blogosphere.

Existing story ranking strategies estimate the newsworthiness of a story based upon an aggregate of recent blog posts. Building upon recent work in the field of applying learning to rank techniques to aggregate problems [11], we propose a novel approach, which leverages existing story ranking approaches as features for use with learning to rank. In particular, we consider each feature to be comprised of a *story ranking component* that estimates a story's newsworthiness and a *temporal component* that specifies for which period of time newsworthiness should be estimated.

We evaluate the proposed learning to rank approach within the context of the TREC top news stories identification task. Our experiments examine the value that the blogosphere can bring to real-time news story ranking. The results show that our approach is effective at ranking news stories in real-time, including those relating to unpredictable events. Indeed, it markedly improves upon the best TREC 2009 and TREC 2010 system performances.

The remainder of the paper is structured as follows. In Section 2, we discuss prior work in the field of news article ranking. Section 3 describes our proposed learning to rank approach. Section 4 describes our experimental setup including corpora used and training/testing details. In Section 5, we report the performance of our LTR approach in comparison to the best TREC systems and discuss the most effective features. We provide concluding remarks in Section 6.

2 News Story Ranking

The Blog track at the Text REtrieval Conference (TREC) examined how news story ranking could be achieved in an automatic manner using evidence from the blogosphere [12]. In particular, the *top news stories identification task* examined whether the blogosphere could be used to identify the most newsworthy stories for a given day [12]. Participants were provided with a large number of news stories from the period of 2008 and had to rank those stories for a fixed set of topic days using only evidence extracted from the Blogs08 corpus - a 28.5 million blog post sample of the blogosphere [9]. Notably, the top news stories identification task was run during both TREC 2009 and TREC 2010. The 2009 task focused on a retrospective setting, i.e. participants were ranking the news stories at a later point in time, while the 2010 task simulated a real-time setting.

Various strategies to retrospectively measure the newsworthiness of a news story using the blogosphere have been proposed. For example, Mejova *et. al.* [15] use the number of 'citations', i.e. the number of blog posts linking to the news story, for ranking. However, this provided limited effectiveness due to the sparsity of links to each news story within the blogosphere. Lee *et. al.* [4] proposed a language modelling approach, whereby the likelihood of each story generating recent blog posts indicates the story's newsworthiness. This approach is more effective, as it avoids the sparsity problem by exploiting the textual similarity between a story and recent blog posts. Similarly, McCreadie *et. al.* [14] also exploited textual similarity between blog posts, proposing to model a story's newsworthiness as a voting process [10]. In particular, they retrieved a fixed number of blog posts related to the news story. Each blog post acts as a 'vote' for the story being newsworthy on the day that the blog post was published. The final score for a news story is the number of votes received for the day of the story.

For the real-time setting introduced in TREC 2010, similar strategies were proposed, however only blog posts published on the topic day or before can be used. For example, Xu *et. al.* [22] estimated the current newsworthiness of a story by summing the BM25 scores for each blog post that was published on the topic day for that story. Hence, only blog posts published on the same day as the story were considered. Similarly, Lin *et. al.* [6] built a vector-space story-to-blog-post representation, using only those blog posts from the story day. They estimated a story's news worthiness based upon the number of blog posts with a high cosine similarity to it.

Recall that for our proposed LTR approach, we define a set of story ranking features, each of which estimates the newsworthiness of a news story. We propose to use existing story ranking strategies, like those described above, as the basis for our story ranking features. In particular, these story ranking strategies act as the *story ranking component* of each of each feature.

However, it is of note that all of the above strategies, excepting that by Mejova *et. al.* [15], use a textual representation of the story. For instance, this could be the headline of an associated news article, or even the full content of such an article. Furthermore, prior work by McCreadie *et. al.* [14] indicated

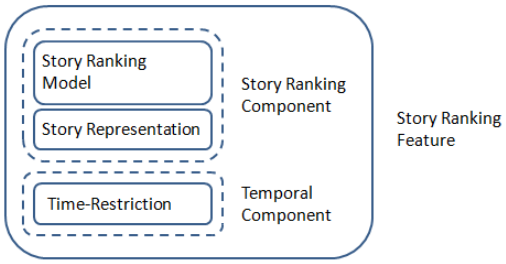

Fig. 1. Illustration of the components of a story ranking feature.

that by enhancing this representation, e.g. by enriching an article headline using query expansion, overall ranking performance could be improved. As such, we consider the story ranking component to be comprised of two sub-components: *the ranking model* and a *story representation*. Indeed, we experiment with eight different story representations in our subsequent experiments. An illustration of our feature components is shown in Figure 1. In the next section, we detail how these two sub-components are combined with a temporal component under our learning to rank approach to rank news stories in real-time.

3 Learning to Rank News Stories

We propose a new learning to rank approach to rank news stories in real-time. Learning to rank techniques are machine learning algorithms which take as input a set of document features and learn weights for each of those features within an information retrieval (IR) system [8]. The aim is to find the weighted linear combination of these features that results in the most effective document ranking.

Various learning to rank techniques have been proposed within the literature. These techniques fall into one of three categories. Point-wise techniques learn on a per-document basis, i.e. each document is considered independently. Pair-wise techniques optimise the number of pairs of documents correctly ranked. List-wise techniques optimise an information retrieval evaluation measure, like mean average precision, that considers the entire ranking list at one time [8]. Prior work has indicated that list-wise techniques learn more effective models [8]. As such, we use a list-wise learning to rank technique in this work. In particular, we use Metzler's Automatic Feature Selection algorithm (AFS) [16]. This is a greedy feature selection algorithm, which iteratively selects the feature that most improves retrieval performance. Notably, features that do not aid retrieval are not selected, i.e. they receive a weight of 0.

Traditional LTR techniques define features on the object that is to be ranked, i.e. the news story in this case. However, news story ranking is an aggregate ranking task, i.e. newsworthiness is defined in terms of a collection of related objects, i.e. blog posts relating to the story, rather than the news story itself. Inspired by prior work in the field of applying learning to rank techniques to aggregate problems [11], we propose an novel approach that generates LTR features by combining different components from multiple story ranking strategies.

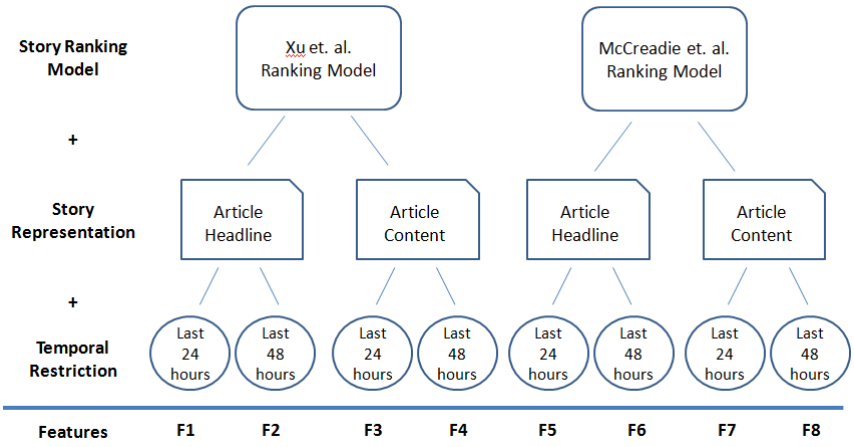

Fig. 2. Eight story ranking features generated from two story ranking models, two story representations (headline and content) and two time-restrictions (previous 24 and 48 hours).

In particular, under our LTR approach, a single feature is comprised of three components, a *ranking model*, *story representation* and *temporal restriction*, as illustrated previously in Figure 1. These components represent a real-time story ranking strategy in a generic manner, i.e. to rank a story, a story ranking model (ranking model) takes as input a textual representation of a story (story representation) to be ranked, and a collection of recent blog posts covering a fixed period of time (temporal restriction). For example, Xu *et. al.* [22]'s real-time approach estimates a story's newsworthiness by summing the BM25 scores for each blog post published during the last 24 hours using the headline from a related article as a query. Therefore, one possible example of a story ranking feature would be to combine the ranking model is that proposed by Xu *et. al.*, an article headline representation and blog posts from only the previous 24 hours. By varying the ranking model amongst those described in Section 2, using methods to enhance article headline or article content representations, and by considering more or less recent blog postings, we generate a large number of different features. Figure 2 illustrates this process.

In the next section, we describe our experimental setup for evaluating our learning to rank approach to real-time news story ranking and its individual features.

4 Experimental Setup

We evaluate our learning to rank approach within the context of the TREC 2009/2010 Blog track top news stories identification task. In particular, we rank news stories from the period of January 2008 to February 2009 using evidence from the TREC Blogs08 corpus [9] which spans the same period. Notably, we

rank news stories published by two different news providers, namely: the New York Times and Reuters. The New York Times corpus, denoted $NYT08$, was used during the TREC 2009 task [12], while the Reuters corpus, denoted $TRC2$, was used during TREC 2010 [5]. For each of these news corpora, for a set of 'topic days', stories published on those days were assessed in terms of their newsworthiness. We evaluate story rankings produced by our learning to rank approach, for both the 55 topic days from $NYT08$ and the 50 topic days from $TRC2$. Table 1 summarises the corpora used during our subsequent experiments.

Corpus	Quantity	Value
*Blogs*08	Time Range	14/01/08 → 10/02/09
	Number of blog posts	28,488,766
$NYT08$	Time Range	01/01/08 → 28/02/09
	# Stories	102,853
	Avg. Stories Per Day	264
	# Topic Days	55
	Avg. Headline Length	7
	Avg. Content Length	418
$TRC2$	Time Range	01/01/08 → 28/02/09
	# Stories	1,800,370
	Avg. Stories Per Day	4,628
	# Topic Days	50
	Avg. Headline Length	8
	Avg. Content Length	225

Table 1. Statistics for the TREC corpora used during evaluation.

Of note is that the $TRC2$ news corpus provides both an article headline for each story, as well as the full article content, whilst the $NYT08$ corpus provides only the article headline. To make these corpora comparable, we independently crawled the missing article content for the $NYT08$ corpus, cleaning the resulting text with the BoilerPipe [3] article extractor. Furthermore, research has shown that using simple heuristics to prune away clearly unimportant stories from news corpora can have a positive impact on story ranking performance [14]. As such, we implement the following simple corpus pruning techniques. On both corpora, we reproduce the pattern, date and uppercase pruning heuristics suggested by McCreadie *et. al.* [14]. However, the editorial patterns that indicate non-newsworthy stories change from corpus to corpus. As such, for the new TRC2 corpus, we analysed three days' worth of news stories and propose an alternative pattern set[1]. We remove all news stories with article headlines starting with the named patterns.

Using the aforementioned corpora, for all stories published on each of the 105 topic days spanning the two news corpora, we generate 160 story ranking features. These features are generated by combining a story ranking model, with

[1] Patterns: "ADVISORY" "ANALYSIS" "BSE" "CBOT" "CHRONOLOGY" "CORRECTED" "CREDIT" "DIARY" "EUROPEAN" "Europe Daily Earnings" "FACTBOX" "FEATURE" "India call money" "INDICATORS" "INSTANT" "Japan Hot Stocks" "NASDAQ" "NSEI" "NYSE" "PRESS" "REFILE" "RESEARCH" "RPT" "SEALED" "SERVICE" "STOCKS" "TABLE" "TAKE" "TECHNICALS" "TEXT" "*TOP" "TRADING" "TREASURIES" "US STOCKS" "WORLD"

a story representation and blog posts from a restricted period of time, as illustrated previously in Figure 2. In particular, we use two of the ranking models described earlier in Section 2, specifically the relevance-based model proposed by Xu *et. al.* [22], denoted *Relevance*, and the voting model proposed by McCreadie *et. al.* [14], denoted *Voting*. Any model-specific parameters are set as specified in the aforementioned papers. Furthermore, we use eight different story representations, four generated from the associated article headline for each story and four from the article content. Moreover, we vary the number of previous days of evidence that we make available using the temporal component. Specifically, we use up to the previous 10 days of published blog posts to rank news stories. Hence, the 2 ranking models, 8 story representations and 10 temporal restrictions multiply together to total 160 individual features. Table 2 lists each of the components which comprise these features and provides a short description.

Component	Name	Description
Model	$Relevance$	Aggregated relevance-based story ranking model [22].
	$Voting$	Voting-based story ranking model [14].
Story Representation	$Headline$	The story headline.
	$QE\_Blogs08$	The headline expanded using the Blogs06 blog post corpus [9].
	QE_NYT06	The headline expanded using 2000 news articles from the New York Times during May 2006.
	QE_TRC2	The headline expanded using 13 days of news stories from the TRC2 corpus but before the start of Blogs08 [5].
	$Content$	The article content.
	$Entities$	Named entities from the article content identified by a Wikipedia-based dictionary [18].
	$Noun - Phrases$	Noun Phrases extracted from the article content [19].
	$Summary$	Story summary generated using part-of-speech tagged article content [7].
Time Restriction	t_{Ndays}	Blog posts are available from the last N days, where $1 \leq N \leq 10$.

Table 2. Feature components and sets for news story ranking.

To train the weights for each of these features, we experiment with two different training regimes, namely Cross-Corpus and Per-Corpus. In particular, under Cross-Corpus training, we train using the topics from one corpus (either $NYT08$ or $TRC2$) and then test upon the topics from the other corpus and vice-versa. Under Per-Corpus training, we train and test on the same topic set using a 5-fold cross validation.

Due to slight differences in setting between the TREC 2009/2010 task formulations, we make the following changes to create a consistent setting and make cross-corpus training possible. Firstly, the TREC 2009 task ($NYT08$ topics) considered that stories both after and before each topic day might still be relevant due to differences in time-zone, which the 2010 task ($TRC2$ topics) did not. We follow the TREC 2010 setting and only rank the stories published on each topic day. Secondly, the 2010 task introduced category classification of articles, i.e. each article was judged as to the degree to which it is important on the topic day with regard to one of five news categories. Importantly, these categories can

introduce a confounding variable into the evaluation, as even a perfect article ranking system will be heavily penalised should it use a poor classifier. In this work, we focus on evaluating overall article ranking performance, and as such leave category classification for future work.

5 Results

In this section, we evaluate the performance of our learned solution and its component features for real-time news story ranking, in addition to examining the types of story (predictable vs unpredictable) that it favours. In particular, we evaluate the overall story ranking performance in Section 5.1. Section 5.2 examines the strongest features selected by our approach. In Section 5.3, we evaluate the importance of the three components of each story ranking feature used, while Section 5.4 investigates whether our approach overall is biased toward predictable events.

5.1 Story Ranking Performance

We begin by evaluating the overall story ranking performance of our approach in comparison to the TREC best system for each of the $NYT08$ (TREC 2009) and $TRC2$ (TREC 2010) topic sets. Table 3 reports the story ranking performance of the best TREC 2009 and 2010 systems as well as the performance of our best individual feature, in comparison to our learning to rank approach when trained under both Cross-Corpus and Per-Corpus regimes.

Model	Training	$NYT08$ Topics (TREC 2009)	$TRC2$ Topics (TREC 2010)
TREC Best System	N/A	0.1862	0.1898
Best Individual Feature	N/A	0.1836	0.1949
Learned Model	Cross-Corpus	0.1165	0.1689
	Per-Corpus	**0.2042***	**0.2248***

Table 3. Comparison between our learning to rank approach when trained under both Cross-Corpus and Per-Corpus training with the best TREC systems in terms of overall story ranking performance under the $NYT08$ (TREC 2009) and $TRC2$ (TREC 2010) story ranking topics. * denotes a statistically significant increase over the best individual feature (t-test $p < 0.05$).

We observe that under Cross-Corpus training, i.e. training on $NYT08$ (TREC 2009) and testing on $TRC2$ (TREC 2010), and vice versa, the performance of our approach is lower than the best TREC system. However, when moving to Per-Corpus training, i.e. a 5-fold cross validation, story ranking performance exceeds that of the best TREC system by 9% and 15% on the $NYT08$ and $TRC2$ topic sets respectively. Moreover, the resulting trained model markedly outperforms the best individual feature used alone by a similar margin. This shows that our proposed learning to rank approach can indeed be effective for real-time story ranking (under Per-Corpus training).

The lesser performance when using Cross-Corpus training indicates that the best features for story ranking are different for the two news corpora and topic sets. This is somewhat to be expected, as the $NYT08$ and $TRC2$ corpora differ markedly in both the story writing style as well as the level of noise contained. In particular, as reported earlier in Table 1, Reuters ($TRC2$) published over 17 times as many stories during each day than the New York Times ($NYT08$), of which many are non-newsworthy stock reports. Furthermore, as a result of crawling and cleaning the article content for $NYT08$ ourselves, this content is likely noisier than pre-provided $TRC2$ article content. Hence, we would expect features based upon article content story representations to be less effective on the $NYT08$ topics and not to generalise between corpora. As such, in our further experiments, we report results using Per-Corpus training only.

5.2 Strongest Story Ranking Features

To examine our approach in more detail, we investigate which of the 160 features generated contribute most to the ranking of news stories. Table 4 reports the five strongest positive and negative features selected by our approach on each topic set.

Feature	$NYT08$ Topics				$TRC2$ Topics			
Type	Components			Weight	Components			Weight
Positive	*Voting*	Headline	t_{1day}	**0.9703**	*Voting*	Headline	t_{1day}	**0.7719**
Positive	*Voting*	Summary	t_{1day}	**0.2762**	*Voting*	Content	t_{2days}	**0.1406**
Positive	*Voting*	Content	t_{1day}	**0.2646**	*Voting*	Noun-Phrases	t_{5days}	0.0468
Positive	*Relevance*	Summary	t_{2days}	**0.1213**	*Relevance*	Summary	t_{3days}	0.0196
Positive	*Voting*	QE_Blogs06	t_{3days}	0.0982	*Voting*	QE_TRC2	t_{3days}	0.0173
Negative	*Voting*	Entities	t_{8days}	-0.0063	*Voting*	Headline	t_{6days}	-0.0035
Negative	*Voting*	Content	t_{8days}	-0.0286	*Voting*	Content	t_{6days}	-0.0038
Negative	*Voting*	QE_TRC2	t_{7days}	**-0.1032**	*Voting*	QE_NYT06	t_{7days}	-0.0063
Negative	*Relevance*	Noun-Phrases	t_{1days}	**-0.1143**	*Voting*	Noun-Phrases	t_{10days}	-0.0077
Negative	*Voting*	Headline	t_{7days}	**-0.5215**	*Voting*	Summary	t_{6days}	-0.0101

Table 4. Strongest 5 positive and negative features on the $NYT08$ and $TRC2$ topic sets. Boldened feature weights indicate features with a high impact on the story ranking.

In general, we observe that the *Voting*-based ranking model is preferred across both topic sets, indicating that it produces features better able to distinguish between newsworthy and non-newsworthy stories than the *Relevance*-based alternative. Of the eight story representations listed previously in Table 2, we see that the headline alone is the strongest story representation across topic sets. However, in contrary to our expectations, the content representations were also selected. This shows that although content is more noisy in the $NYT08$ corpus than its equivalent in $TRC2$, it appears to still provide valuable ranking evidence. Indeed, it is of note that of the positive features selected using the $NYT08$ topics, a higher weight is assigned to the shortened summary of the content than the content unaltered. This indicates that the summarisation is removing noise from the content for $NYT08$ that is unnecessary for $TRC2$, although in contrast, the Noun-Phrase representation appears be noisy. In terms

of the temporal restrictions that we place on the story ranking models for the real-time setting, we make the following two observations. Firstly, only features that use blog posts from the one or two days before the time of ranking appear to be useful. Secondly, as we relax the temporal restriction and use older blog posts, the story ranking features become negative, i.e. if a story has been discussed extensively beforehand then the story is less likely to be newsworthy. Indeed, for the $NYT08$ topics, the strongest positive feature ($Voting$ + Headline + t_{1day}) becomes the strongest negative feature by changing the temporal restriction. Notably, negative features appear not to add value on the $TRC2$ topics.

5.3 Story Ranking Components

We next examine whether the features generated by varying each of the story ranking components are useful. In particular, we follow a leave-one-out approach, whereby we discard any features generated by varying a given single component, leaving only a single instance of that component. In particular, for the story representation we keep only features using the article headline representation. Similarly, for the temporal component, keep only features that used blog posts 1 day old or less. For the ranking model we keep only those features generated by one or other of the two story ranking models considered.

Table 5 reports the story ranking performance of our learning to rank approach trained upon feature subsets. We see that by removing features generated by the $Voting$ model, the ranking performance markedly decreases. This confirms our earlier observation that the $Voting$-based model is more effective than the *Relevance*-based alternative. Indeed, we see that by removing *Relevance*-based features instead, little story ranking performance is lost.

Voting Model	*Relevance* Model	Story Representations	Time Restrictions	$NYT08$ Topics (TREC 2009)	$TRC2$ Topics (TREC 2010)
✔	✔	✔	✔	0.2042	0.2248
✘	✔	✔	✔	0.1729	0.1130
✔	✘	✔	✔	0.2034	0.2026
✔	✔	✘	✔	**0.2120**	0.1900
✔	✔	✔	✘	0.1658	**0.2313**

Table 5. Story ranking performance of our learning to rank approach in when training on different feature sets using Per-Corpus (5-fold cross validation) under the $NYT08$ (TREC 2009) and $TRC2$ (TREC 2010) story ranking topics.

Examining the story representations, performance decreases markedly on the $TRC2$ topics by discarding alternate representations, showing that indeed, story representations can have a strong impact on performance. However, unexpectedly, we see that by using the headline of the story alone, ranking performance is slightly increased on the $NYT08$ topics instead. The inability of the learner with all features to find this better solution highlights an issue with greedy learning to rank approaches. In particular, greedy learners, while effective, are not guaranteed to find the optimal solution and can be trapped in a local minima.

In terms of the temporal restrictions, we see that ranking performance is heavily degraded on the $NYT08$ topic set when only blog posts from the same day as the story (t_{1day}) are considered. On the other hand, story ranking performance on the $TRC2$ corpus is not negatively impacted, indeed performance gains are observed instead. This emphasises the differences in the $NYT08/TRC2$ topic sets. In particular, the performance gain observed on $TRC2$ indicates that the corpus contains a higher proportion of unpredictable events, i.e. those for which only very recent blog posts are relevant. Moreover, the different performances observed between the topic sets confirm our earlier observation that the learner on the $NYT08$ topics used older blog posting activity as negative features, while on the $TRC2$ topics it did not. Indeed, a key advantage that our approach has over the existing story ranking models that it employs as components, is the ability to adapt to different news corpora.

5.4 Predictable vs Unpredictable Events

Lastly, we examine whether the blogosphere lacks sufficient freshness to accurately rank stories relating to unpredictable events in real-time. In particular, we select the top 5 most newsworthy stories as returned by our learned approach for each of the 50 topics in the $TRC2$ corpus, creating a set of 250 newsworthy stories. We manually annotated each of these as reporting about predictable or unpredictable events. Should the blogosphere lack sufficient freshness, then our story ranking approach will be more likely to identify predictable events over unpredictable ones, i.e. the vast majority of the 250 news stories would relate to predictable events.

However, in contrast, our results show that 46% of the top stories were unpredicable, while 54% were predicatable (a close to even spread). This indicates that, at least for the simulated real-time setting introduced by TREC, there is no evidence to indicate that bloggers react too slowly for unpredictable stories to be effectively ranked.

6 Conclusions

In this paper, we proposed a novel learning to rank approach which leverages current blog posts to rank news stories in a real-time manner. In particular, we used existing news story ranking models in conjunction with varying story representations and temporal restrictions to generate 160 story ranking features. We evaluated our proposed learning approach within the context of the TREC 2009 and 2010 Blog track top stories identification task. Our results show that the proposed approach is effective at ranking news stories in real-time. Indeed, it improves upon both the best TREC 2009 and TREC 2010 systems and its best internal feature by over 9% and 13% respectively. Moreover, we examined both the individual features and story ranking components used by our learning to rank approach, highlighting those that were most useful in terms of impact on the story ranking. Lastly, we investigated whether our approach based upon measuring bloggers response to news stories, was biased toward predictable events, due to a lack of timely posting regarding unpredicatable events. However, we found that there was no evidence to indicate that this was the case, indeed 46% of top stories ranked by our system were related to unpredicatble events.

References

1. Newspaper Association of America (NAA): Newspaper Web sites attract more than 70 million visitors in June; over one-third of all Internet users visit newspaper Web sites (2010), http://www.naa.org/PressCenter/SearchPressReleases/2009/NEWSPAPER-WEB-SITES-ATTRACT-MORE-THAN-70-MILLION-VISITORS.aspx, accessed on 25/01/2010
2. Jones, R., Diaz, F.: Temporal profiles of queries. ACM Trans. Inf. Syst. 25(3), 14 (2007)
3. Kohlschütter, C., Fankhauser, P., Nejdl, W.: Boilerplate detection using shallow text features. In: Proceedings of WSDM'10.
4. Lee, Y., Jung, H.y., Song, W., Lee, J.H.: Mining the blogosphere for top news stories identification. In: Proceeding of SIGIR'10.
5. Leidner, J.L.: Thomson Reuters releases TRC2 news corpus through NIST (2010), http://jochenleidner.posterous.com/thomson-reuters-releases-research-collection, accessed on 16/01/2011
6. Lin, Y.F., Wang, J.H., Lai, L.C., Kao, H.Y.: Top stories identification from blog to news in TREC 2010 Blog track. In: Proceedings of TREC'10.
7. Lioma, C., Macdonald, C., Plachouras, V., Peng, J., He, B., I., O.: University of Glasgow at TREC 2006: Experiments in Terabyte and Enterprise Tracks with Terrier. In: Proceedings of TREC'06.
8. Liu, T.Y.: Learning to rank for Information Retrieval. In: Foundations and Trends in Information Retrieval: Vol. 3: No 3. pp. 225–331 (2009)
9. Macdonald, C., Ounis, I.: The TREC Blogs06 collection : Creating and analysing a blog test collection. Tech report. *Univ. of Glasgow.*
10. Macdonald, C.: The Voting Model for People Search. Ph.D. thesis, Univ. of Glasgow (2009)
11. Macdonald, C., Ounis, I.: Learning models for ranking aggregates. In: Proceedings of ECIR'11.
12. Macdonald, C., Soboroff, I., Ounis, I.: Overview of TREC-2009 Blog track. In: Proceedings of TREC'09). NIST.
13. Matheson, D.: Weblogs and the epistemology of the news: Some trends in online journalism. New Media and Society 6(4), 443–468 (2004)
14. McCreadie, R., Macdonald, C., Ounis, I.: News article ranking: Leveraging the wisdom of bloggers. In: Proceedings of RIAO 2010.
15. Mejova, Y., Ha Turc, V., Foster, S., Harris, C., Arens, B., Srinivasan, P.: TREC Blog and TREC Chem: A view from the corn fields. In: Proceedings of TREC'09.
16. Metzler, D.A.: Automatic feature selection in the Markov random field model for Information Retrieval. In: Proceedings of CIKM'07.
17. Mishne, G., de Rijke, M.: A study of blog search. In: Lecture Notes in Computer Science, vol. 3936, pp. 289–301. Springer (2006)
18. Santos, R.L.T., Macdonald, C., Ounis, I.: Voting for related entities. In: Proceedings of RIAO'10.
19. Schmid, H.: Treetagger. TC project at the Institute for Computational Linguistics of the University of Stuttgart (1994)
20. Sussman, M.: The state of the Blogosphere 2009 (2009), http://technorati.com/blogging/article/state-of-the-blogosphere-2009-introduction/, accessed on 13/05/2010
21. Thelwall, M.: Bloggers during the London attacks: Top information sources and topics. In: Proceedings of WWW'06 blog workshop.
22. Xu, X., Liu, Y., Xu, H., Yu, X., Peng, Z., Cheng, X., Xiao, L., Nie, S.: ICTNET at Blog track TREC 2010. In: Proceedings of TREC'10.