University of Waterloo: Logistic Regression and Reciprocal Rank Fusion at the Microblog Track

Adam Roegiest and Gordon V. Cormack
David R. Cheriton School of Computer Science, University of Waterloo

1 Introduction

For the second iteration of the Microblog Track, two tasks were given to participants to complete. The first was to perform the same ad hoc search task as the 2011 iteration. The goal of the task was to expand on last year's methods with 60 new topics and to explore different measures of evaluation. The second task was to filter the corpus with respect to the 2011 topics in an attempt to simulate a streaming environment and how simulating user feedback can effect retrieval results.

For the ad hoc search task, we decided to expand on last year's approach by continuing to use the Wumpus Search Engine[1] and adding in a logistic regression classifier (denoted GCLR in this article), first used in the TREC 2007 Spam Track [5]. In addition, pseudo-relevance feedback was conducted this year by taking a swapdocs approach[3], which will be expanded upon later. As well, a semi-automatic logistic regression run was conducted using seed documents provided by a user.

For the filtering task, only different methods of training GCLR were examined. No manual feedback was conducted for the filtering task.

2 Experimental Setup

2.1 Corpus

From May 27th to June 3rd, the provided HTML crawler[2] was run after which the tweets were repaired using the provided repair code. Once it was determined that the HTML crawler returned tweets with null text bodies, the supplied repair code was modified to re-fetch tweets with null bodies. The modified repair script was run six times until it was no longer feasible to continue trying to repair tweets. Table 1 shows the distribution of tweets in our copy of the corpus. We note that status code 200 denotes manually posted tweets, 302 denotes Twitter supported retweets, 403 denotes tweets the posting user has protected, and 404 denotes deleted tweets. We note that any tweet marked as 403 or 404 has no associated information as it is no longer available for public use.

Table 1: Corpus distribution

HTTP Status	Num. of Tweets	Num. null text
200	14,313,888	14
302	1,137,183	25,571
403	138,560	138,560
404	552,181	552,181

No further attempt was made to retrieve the remaining 25,585 tweets as we were unsure as to whether or not the tweets were still available, e.g. not protected or deleted. In addition, we desired to begin working with the corpus and did not believe the missing tweets would significantly affect our results. However, only those tweets which had a 200, 301, or 302 status code had the potential to be returned in any of the runs described in this work.

2.2 Ad Hoc Task Methodology

The basic idea for our methodology in the ad hoc task was to extend our approach from last year by using a swapdocs approach to pseudo-relevance feedback. The swapdocs approach to pseudo-relevance feedback is to have different systems (in our case, ranking methods on different features and logistic regression) score documents with respect to a topic and then exchange each of the top k (k=20 in our case) results from the different systems as the basis for pseudo-relevance feedback instead of relying on some other form of document selection. Thus, the swapdocs approach can be seen as having two phases; the

[1] Available at `http://www.wumpus-search.org`

[2] Available at `github.com/lintool/twitter-corpus-tools`

initial seed document retrieval phase and a pseudo-relevance feedback phase. This results in the creation of N^2 results (where N is the number of systems) and as with last year these results were combined using reciprocal-rank fusion (RRF) [6]. The formula used was as follows:

$$RRFscore(t) = \Sigma_i \frac{1}{60 + r_i(t)}$$

where $r_i(t)$ is the rank of the tweet t in result set i.

For the first phase, GCLR[3] was trained on a 5:1 bias[4] of the topic statement to random tweets (selected from before the query time), such that there were 100 copies of the topic statement and 20 random tweets. For the second phase, GCLR was trained on the top 20 tweets from a system and 20 random tweets for each topic. In both phases, GCLR would classify all documents up to the query time.

Our approach to Wumpus was much the same as it was last year. Feature engineering was performed for each query by creating six different indexes for Wumpus, where the index stores each tweet's features. Table 2 briefly describes the features used for each index. Tweets were ranked using five different methods, which are outlined in Table 3, on each index. In the initial phase, each ranking method for each index was provided just the topic statement. In the second phase, Okapi-style pseudo-relevance feedback [2] was conducted using the top 20 documents from each of the systems in the initial phase. However, due to Wumpus limitations CDR could not be used in the second phase. Due to poor performance on the 2011 topics, PC was not used in the second phase. In addition, pseudo-relevance feedback was performed using a language model of all tweets before the earliest query time in the 2012 topics.

We note that the actual queries issued to Wumpus were not just the provided topic but each whitespace delimited word was treated as its own search term. If this was not done then Wumpus would attempt to look for tweets containing exactly the topic phrase and this is not generally a desirable behaviour for a search engine.

Finally, for each set of results the only the the highest scoring 1000 tweets were used by RRF to combine results and only the top 1000 results from each run were submitted to NIST for evaluation.

Table 2: Description of engineered features

Index	Description
1	This index includes stemmed and unstemmed versions of words present in the tweet. The stemming is accomplished using a Porter stemmer. In addition, word bigrams are included in the index.
2	Identical to Index 1 but with no word bigrams present.
3	This index contains only unstemmed versions of words and word bigrams.
4	Identical to Index 3 but contains no word bigrams.
5	Instead of words as before, character 3-grams are used in the index. Subsequently, stemming is not used as it has no benefit.
6	Identical to Index 5 but uses character 4-grams.

2.2.1 Ad Hoc Runs

In this section, the runs submitted to TREC for the ad hoc task are discussed.

uwatrrflm - Baseline (RRF-2LM)

This run was our baseline and did not use the swapdocs approach. It is a repeat of our best performing run in the 2011 Microblog Track, except that only Okapi-style feedback was used to select candidate documents (as opposed to the swapdocs approach) and two tweet based language models were used (one based on the 2011 topics and one based on the 2012 topics).

uwatrrfall - Swapdocs Approach (RRF-Swapdocs)

This run was conducted exactly as outlined above.

uwatgclrbase - GCLR Initial Phase (GCLR-Initial)

This run only contains the highest scoring 1000 tweets from GCLR in the first phase, e.g. trained on the 5:1 bias of topic statement. The purpose of this run was to determine a GCLR baseline from which we could judge improvements.

uwatgclr - GCLR Manual Feedback (GCLR-Manual)

[3] GCLR uses byte 4-grams as features

[4] The logistic regression classifier was trained on repeated examples of relevant documents so that the desired ratio of relevant to non-relevant was achieved. We call this ratio "bias."

Table 3: Ranking Methods used

Method	Description
Okapi BM25[7]	BM25 is a bag-of-words retrieval method that ranks documents based upon how often query terms appear in the document and does not take into account any relationships between query terms in the document (e.g. proximity). From the 2011 topics we found that (b,k1)=(0.25, 0.4)
Ponte-Croft Language Modelling[8]	This method infers a nonparametric language model for each document and then ranks each document based on the query. That is, a document is ranked for a query using the product of the probability of producing query terms in the document and the probability of not producing other terms.
Language Modelling with Dirichlet Priors[9]	A language model, in this method, is treated as a multinomial distribution and the Dirichlet distribution is used to smooth the probability of word occurrence.
Cover-Density Ranking[4]	Given a set of query terms Q1, ..., Qn, this method builds a boolean AND for all subsets (e.g. Q1^Q2^Q9^Qn) and ranks these subsets by the sum of their terms' Inverse Document Frequency values. Then all documents are ranked based upon the largest subset they contain.
Divergence from Randomness[1]	This method uses inverse term frequency, the ratio of two Bernoulli processes, and the hypothesis that the term frequency density is inversely related to the length to create a ranking formula for documents.

This run uses manual feedback with GCLR. The seed documents for GCLR relevance training came from a manual assessor who issued queries to Wumpus and selected relevant tweets from the retrieved results (all results were from before the query time). These were then trained in 1:1 fashion with randomly selected tweets (which were deemed not relevant). For topics with less than 20 tweets, the deficit was made up by including 5 copies of the topic statement for each tweet less. However, there was no maximum cut off for tweets selected. We note that the assessor was instructed to avoid selecting tweets with duplicate content to those previously selected.

2.3 Filtering Task Methodology

For the Filtering task, we used only GCLR (with byte 4-grams as features) as our means of filtering. For the filtering task, we submitted two runs for official evaluation and they are outlined below.

uwn - GCLR baseline

This run was trained again in a 5:1 bias as was done in ad hoc, such that there were 1000 copies of the given relevant document, 1000 copies of the query, and 400 random tweets (trained as not relevant). After the training phase, GCLR classified all documents and no further training was performed during classification.

uw - GCLR Feedback

This run was trained again in a 5:1 bias as was done in ad hoc, such that there were 1000 copies of the given relevant document, 1000 copies of the query, and 400 random tweets (trained as not relevant). After the training phase, GCLR classified all documents. Any time a document was deemed to be relevant then GCLR was trained based upon it's actual relevance score in the 2011 qrels. If the document was deemed relevant in the qrels it was trained as relevant; if the document was deemed relevant in the qrels it was trained as non-relevant; if the document was absent from the qrels, no training was performed.

2.4 Evaluation

Tweets for the ad hoc task were judged to be relevant by NIST assessors after the results from all participating teams were pooled. Assessors were asked to judge tweets as relevant, highly relevant, or not relevant with respect to the topic. In addition, only tweets that primarily contained the English language were allowed to be judged relevant. The primary measure for the track this year was the ROC (receiver operating characteristic) curve, however, Ian Soboroff (a track coordinator) could not determine a means of unifying the ROC curves from different topics. Accordingly, we choose to look at Precision@30 which would allow us to compare our results with our runs from last year. However, as only summary information was given for highly relevant tweets this year, we only present the percentage of topics at and above the median for them. The average relevant returned at 30 for all relevant tweets was calculated using the highest scoring 30 tweets for each topic as we used last year.

Tweets for the filtering task used the same judgements as those from last year as the filtering task was based upon the 2011 topic set. The filtering task had four measures set precision, set recall, F (with beta = 0.5), and linear utility. We present the number of topics for which our runs were at or above the median score for F, precision, and recall.

3 Results and Discussion

Tables 4 and 5 provide ad hoc summary results for topics with highly relevant tweets only and topics with relevant tweets and present the results from the highest scoring run we performed last year. From these summary results, our baseline run (RRF-2LM) appears to have performed most of the time at or above median, which may indicate that it is a good baseline. This is also true when it's performance is at least as good as our best run from last year when you factor in that only 33 topics had highly relevant tweets last year (and only 49 with relevant tweets). The swapdocs run does appear to improve upon our baseline but not by a lot.

To further examine our baseline run (RRF-2LM), we looked at the performance of the run by removing a language model and regardless of which language model was removed there appeared to be little change in our P@30 results. Although, the language model based upon this year's topics did have a better performance than our 2011 language model. This is not all that surprisingly because the language model for the 2012 topics was larger and thus more likely to be an accurate representation of the language used on Twitter during the collection period. It is worth noting that removing Ponte-Croft ranking from RRF-2LM did improve performance but not by a large enough margin that we thought it worth reporting in this paper.

From these results, we can see something that is potentially interesting. It appears that for all the extra effort required of the swapdocs approach that there is not a huge increase in performance and it is possible that just adding GCLR trained on results from interim results may boost the performance of RRF-2LM enough that there is not enough benefit in using the swapdocs approach. In addition, anecdotal evidence would suggest that RRF-2LM is also a much more time efficient algorithm than RRF-Swapdocs (as it does not have to compute N^2 combinations). Accordingly, the gains made by using the swapdocs approach may be ruled out due to real world performance concerns.

The ad hoc performance of GCLR with manually selected training documents was a nice a result and we would like to further explore how this approach can be modified to increase performance. As well, we would like to see if the results are boosted when the assessor is given a larger deadline for finding seed documents as the assessor used had a very small window for this run.

Finally, the performance of initial swapdocs phase GCLR is unsurprising given that the training bias was quickly determined using 2011 topics and was not vigorously tested. Further, it appears that this approach is limited by the language used in topic statement as it appears from the results that the performance of GCLR-Initial was only as high as it was due to its performance on a few topics. Although, this would require manual verification by comparing individual topic performance to the topic statement to see if there is some correlation.

The results for the filtering track were not particularly good, as we expected based on previous TREC Filtering Track experiments. We are at a loss to explain why the "best" scores for recall, precision, and F1 are so high.

4 Future Work

Given the performance of GCLR with manually selected training documents, we are likely to continue exploring how performance can be increased further. As well, it would be nice to determine a means of incorporating manual intervention in the Filtering Task, perhaps by attempting to find "relevant" documents in the tweets that occur before the oldest tweet time condition in the filtering task. It may be beneficial to examine the role of different features in improving tweet classification using GCLR and perhaps determine if there is an ideal bias when training classification/filtering methods for tweets.

In addition, it would likely boost performance to have a pre-screening stage of filtering where all non-English tweets are filtered out so that there is less potential of returning non-English tweets. We had desired to do this last year but felt that it may be a waste of time to attempt to do so this year given various time constraints.

5 Conclusions

This year at the Microblog Track, we saw that performance of all systems for the ad hoc task seemed to be much improved from the 2011 iteration. In particular, that manually seeding logistic regression achieved good results and that applying swapdocs to the task may not be as beneficial as one might hope. The filtering task was largely underwhelming and we did not achieve very good results and are unsure of how the “best” scores were achieved.

References

[1] Gianni Amati and Cornelis Joost Van Rijsbergen. Probabilistic models of information retrieval based on measuring the divergence from randomness. *ACM Trans. Inf. Syst.*, 20:357–389, October 2002.

[2] Bodo Billerbeck and Justin Zobel. Questioning query expansion: an examination of behaviour and parameters. In *Proceedings of the 15th Australasian database conference - Volume 27*, ADC ’04, pages 69–76, 2004.

[3] Charles L. Clarke, Gordon V. Cormack, Thomas R. Lynam, Chris Buckley, and Donna Harman. Swapping documents and terms. *Inf. Retr.*, 12(6):680–694, Dec 2009.

[4] Charles L. A. Clarke, Gordon V. Cormack, and Elizabeth A. Tudhope. Relevance ranking for one to three term queries, 2000.

[5] Gordon V. Cormack. University of waterloo participation in the trec 2007 spam track. In *Sixteenth Text REtrieval Conference (TREC-2007)*, Gaithersburg, MD, 2007. NIST.

[6] Gordon V. Cormack, Charles L A Clarke, and Stefan Buettcher. Reciprocal rank fusion outperforms condorcet and individual rank learning methods. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, SIGIR ’09, pages 758–759, 2009.

[7] K. Sparck Jones, S. Walker, and S. E. Robertson. A probabilistic model of information retrieval: development and comparative experiments. In *Information Processing and Management*, pages 779–840, 2000.

[8] Jay M. Ponte and W. Bruce Croft. A language modeling approach to information retrieval. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR ’98, pages 275–281, 1998.

[9] Chengxiang Zhai and John Lafferty. A study of smoothing methods for language models applied to ad hoc information retrieval. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR ’01, pages 334–342, 2001.

Table 4: Ad Hoc Results for highly relevant tweets only

	2012 Runs				2011 Runs
	GCLR-Initial	GCLR-Manual	RRF-Swapdocs	RRF-2LM	RRF-1LM
Percent above median	10.17	64.41	57.63	54.24	66.67
Percent at median	25.42	20.34	27.12	23.72	21.21
Avg. rel. ret. 30	3.14	8.03	6.59	6.19	3.82

Table 5: Ad Hoc Results for all relevant tweets

	2012 Runs				2011 Runs
	GCLR-Initial	GCLR-Manual	RRF-Swapdocs	RRF-2LM	RRF-1LM
Avg. rel. ret. 30	6.83	13.71	11.64	11.37	12.65

Table 6: Filtering results for F-measure, Recall and Precision

Runs	F (beta=0.5)		Recall		Precision	
	Above median	At median	Above median	At median	Above median	At median
uwn	7	3	7	1	26	3
uw	3	1	3	1	36	2