The Role of Anchor Text in ClueWeb09 Retrieval

Vo Ngoc Anh Alistair Moffat

Department of Computer Science and Software Engineering
The University of Melbourne
Victoria 3010, Australia

Abstract: *This report describes the work done at The University of Melbourne with the* `ClueWeb09` *data corpus for the Web Track of TREC-2009 and TREC-2010, and for the Session Track of TREC-2010. We found that the impact-based retrieval model works well for the corpus, and that, along with some other factors, the use of an anchor text collection significantly boosts the retrieval effectiveness.*

1 Introduction

This report describes work done at the University of Melbourne for TREC-2010. In addition, it also summarizes our TREC-2009 participation as that was not reported elsewhere. For these two years, we participated in the Ad-hoc and Diversity Tasks of the Web Track, and for 2010 alone we also submitted to the Session Track. In all of these, we employed the whole English portion of the `ClueWeb09` corpus and hence participated in, by TREC definition, Category A.

Experiments were performed using our locally-developed software. The system has been developed for several years at the University of Melbourne, targeting both efficiency and effectiveness of ad-hoc retrieval. The system engages an impact-based retrieval model, impact-sorted indexes, and fast index compression.

2 Retrieval Models

As a common point for all of our submissions in TREC 2009-2010, impact-based retrieval methods were employed. However, the 2010 submissions are distinguished by the employment of of spam filtering.

2.1 Similarity Computation

For a document collection and a query, the similarity between the query and a collection document is computed using either `IMP` – the impact model [Anh and Moffat, 2005] or `IBM25` – its BM25-style formulation [Anh et al., 2008].

In short, `IMP` and `IBM25` can be considered as variations of the conventional vector-space model (see, for example, Salton [1989]) and the `BM25` formulation [Robertson et al., 1994], respectively. However, unlike their original counterparts, they operate over term impacts instead of term frequencies, and do not depend on any tuning parameter. At indexing time each document d is examined as an individual document, in isolation from any collection-wide statistics, and each distinct term t in the document in this process is associated with a *document term impact* value $\omega_{d,t}$. This impact value is an integral value between 1 and 8 inclusive, and can be regarded as a

normalized value of the frequency of t in d. This process does not use any knowledge from outside of the document. The impact values are then stored in the index.

At query time, the query q is also considered as an independent document, but for each distinct term $t \in q$, the collection frequency of t in the collection is employed (instead of the frequency of t in q), and a similar process of normalization as in the case of documents is used to define *query term impact* $\omega_{q,t}$.

Finally, the similarity $S_{d,q}$ between q and a document d is computed using one of the two methods. The first method, `IMP`, is the original impact model, which specifies that

$$S_{d,q} = \sum_{t \in d \cap q} \omega_{d,t} \cdot \omega_{q,t} \ . \tag{1}$$

The second method, `IBM25`, is a slightly modification of what described by Anh et al. [2008]. Its exact formulation is

$$S_{d,q} = \sum_{t \in d \cap q} \log \frac{N - f_t + 0.5}{f_t + 0.5} \cdot \frac{\log(1 + \omega_{d,t})}{k_1 + \log(1 + \omega_{d,t})} \cdot \frac{\log(1 + \omega_{q,t})}{k_3 + \log(1 + \omega_{q,t})} \ , \tag{2}$$

where $k_1 = 2$ and $k_3 = 1000$, and are constant across document collections.

Spam Filtering

Cormack et al. [2010] report that the use of spam filtering or re-ranking significantly improves retrieval effectiveness for most of the systems that participated in Web Track 2009. For our 2010 submissions, we employed their *fusion* spam score to remove the spammiest 30% pages from each retrieved list. We do not, however, evaluate the effect of spam filtering on our 2010 submissions.

Anchor Text and Links

For all of our submissions, the original `ClueWeb09` was indexed and employed for retrieval. In addition, for the majority of submissions, we also created and indexed the incoming anchor text collection. For that purpose, the canonical anchor text (that is, not including any surrounding text) of each link in document was extracted and gathered if the destination document was also a document in `ClueWeb09`. The resultant anchor text collection was separately indexed and queried in the same way as the original `ClueWeb09` collection.

For TREC-2009, our submissions were based on either the content-only or the anchor-only collections. For TREC-2010 we also employed the fusion between content-only scores, anchor-only scores, and, in some cases, PageRank scores. When fusion was used, the individual scores were normalized so that to share to maximal score of 1 on the per-query basic, and then normalized scores are weighted and then combined. Typically, the weight of a non-content score is 0.25. Our post-TREC experiments show that the contribution of the PageRank on retrieval effectiveness was marginal. While the problem seems interesting, we are not making any investigation in this report.

3 Web Track 2009

There are two tasks in Web Track 2009 – *Ad-hoc* and *Diversity*. While the former is a conventional TREC task, the latter aims, for each query, to get a ranked list of pages that provide broad coverage of the query and avoid excessive redundancy within the list. In our submissions, we focused on the Ad-hoc Task and applied the same retrieval techniques to both of the tasks.

We assumed that in a large dataset like `ClueWeb09`, good documents would possess a non-trivial number of incoming links, and a reasonable volume of incoming anchor text. Moreover, for

Run	Description	MAP	P@5	P@10	P@20
muadimp	IMP, content only	**0.044**	0.108	0.116	0.125
muadanchor	IMP, anchor text only	0.0256	**0.296**	**0.250**	**0.179**
muadibm5	IBM25, content only	0.044	0.108	0.118	0.125

Table 1: Effectiveness performance of the 2009 Web Track Ad-hoc submissions. Each figure in bold is the highest in that column.

Run	Description	α-nDCG			precision-IA		
		@5	@10	@20	@5	@10	@20
mudvibm5	IBM25, content only	0.106	0.112	0.132	0.057	0.046	0.043
mudvimp	IMP, anchor text only	**0.220**	**0.241**	**0.268**	**0.091**	**0.073**	**0.061**

Table 2: Effectiveness performance of the 2009 Web Track Diversity submissions. Note that the run mudvibm5 is identical to muadibm5, and mudvimp – to muadanchor.

each web page, the content of its incoming anchor text would be better than the content of itself in objectively describing the page. That is, doing retrieval in the collection of incoming anchor text could yield more effective results than doing that in the original collection, especially when talking about the Diversity Task.

Table 1 and Table 2 list our submissions for the Ad-hoc and for the Diversity Tasks, respectively. The tables show a strong correlation of the effectiveness performance across the two tasks. They also show that, as anticipated, the anchor text collection does work significantly better than the original text collection in terms of retrieval accuracy. One surprise is that Table 1 shows very marginal difference on performance between IMP and IBM25. While that is not consistent with Anh et al. [2008], it might due to the change in IBM25 formulation for this work, which clearly targets the symmetrical usage of $\omega_{d,t}$ and $\omega_{q,t}$.

Subsequently, Cormack et al. [2010] showed that the use of spam filtering significantly improved the accuracy of most of the TREC 2009 submissions, including ours.

4 Web Track 2010

For the Web Track 2010 we intended to compare the use of both content and anchor components with that of content only. We again employed the two similarity formulations defined by Formula 1 and Formula 2. Unlike the previous year, we applied spam filtering to all of our submissions. Any document that had fusion score (as defined by Cormack et al. [2010]) of at least 0.70 was removed from the result lists.

When both content and anchor text are employed, for each query q two separate searches are conducted, one over the original Web document collection, and the other over the anchor text collection. The search method and setting are similar for both of the cases. Each result list is then normalized linearly so that the top score of the list is 1. Then, the lists are merged, and for each document d the final score is computed as

$$S_{d,q} = (1-\alpha) \cdot S^{\mathrm{C}}_{d,q} + \alpha \cdot S^{\mathrm{A}}_{d,q} \,, \tag{3}$$

where $\alpha = 0.25$, $S^{\mathrm{C}}_{d,q}$ is the normalized content score, and $S^{\mathrm{C}}_{d,q}$ is the normalized anchor score.

Table 3 shows the performance of our Ad-hoc submissions. Overall, the effectiveness performance is good, given that no extra information except for the spam scores was employed. The table

Run	Description	MAP	P@5	P@10	P@20
UMa10BSF	IBM25, content only	0.066	0.183	0.206	0.192
UMa10BASF	IBM25, content + anchor	**0.088**	0.383	0.356	**0.321**
UMa10IASF	IMP, content + anchor	0.087	**0.394**	**0.358**	0.319

Table 3: Effectiveness performance of the 2010 Web Track Ad-hoc submissions. In all of the runs, the 30% spammiest pages according to the fusion scores were discarded.

Run	Description	α-nDCG			precision-IA		
		@5	@10	@20	@5	@10	@20
UMd10ASF	IBM25, anchor text only	0.236	0.260	0.293	0.127	0.109	0.086
UMd10BASF	IBM25, content + anchor	0.275	**0.336**	0.379	0.162	**0.152**	**0.131**
UMd10IASF	IMP, content + anchor	**0.281**	0.335	**0.380**	**0.165**	0.144	0.130

Table 4: Effectiveness performance of the 2010 Web Track Diversity submissions. In all of the runs, the 30% spammiest pages according to the fusion scores have been discarded. Note that the run UMd10BASF is identical to UMa10BASF, and UMd10IASF is identical to UMa10IASF.

clearly shows the advantage of using fusion of content and anchor over that of content alone – by about 100% for P@5, and about 70% for P@10. Similar to our 2009 submissions, the difference in performance of IMP and IBM25 is marginal.

For the 2010 Diversity Task we argued that the anchor text collection might play a central role for effective retrieval. In fact, when different authors write outgoing anchor text to a particular web page, they might pay attention on different aspects of the page. Similarly, one author can also give a number of anchor text to the same page, likely to different coverage. In short, incoming anchor text for a page can objectively high-light various important aspects of web pages and hence is probably valuable for the Diversity search.

Our Diversity submissions are summarized in Table 4. A comparison with Table 2 reveals that our argument in regard to the role of anchor text is not contradicted. Indeed the advantage of anchor text over content-only runs is dramatic. Moreover, although the fusion between anchor text and content works better than the anchor text alone, the performance gap is relatively small.

5 Session Track 2010

The central point for this track is to measure the ability of retrieval systems to improve search accuracy after learning that users re-formulate their initial queries. It is supposed that in a search session, a user first issues a query RL1, obtains some results, then for various reasons (such as being unhappy with the results or realizing mistakes), re-formulates RL1 to RL2 and re-submits. The question is whether the retrieval systems can employ the history of RL1 to improve the overall quality of the results returned for RL2.

For the purpose of the track, each participating retrieval system submitted three output sets per session: set RL1 for the original query RL1; set RL2 for the re-formulated RL2 as a stand-alone query; and set RL3 for employing query RL3 which, in general, is a system's re-formulation of RL2 using the knowledge of both RL1 and RL2. Note that the output for RL2 is purely for the comparison purpose, that the system performance is assessed through the accuracy of the outputs of RL3 relative to that of RL1.

Methodology

For our submissions we supposed:

- that the user re-formulates the query from RL1 to RL2 because the RL1 output is of poor quality, that is, its quality is considered to be inferior;
- that the user has enough patience that she or he issues RL2 after skimming a large number (say, $2{,}000$, which the number of answers per query set by the Track's organizers) of items of the RL1 output; and
- that the user is an experienced searcher, and hence RL2 is a well-formulated and highly-accurate query.

With these two suppositions, it is assumed that

- there is no need for the system to change the query RL2, and so the query RL2 will be used to generate RL3 output; and
- since the run RL1 is considered as a failure, items appear in the RL1 output should be discouraged from appearing in the RL3 output.

Based on these assumptions, we designed the following simple strategy for our submissions. The queries RL1 and RL2 are submitted to our search systems. No actual search is done for RL3. Instead, the RL3 output list is generated from the respective lists of RL2 and RL1. First, all documents d in RL2 and their respective scores s_d^2 are included in the candidate list for RL3. Then the scores of the RL1 and RL3 lists are normalized so that they share a common maximal score (on a per-query basic). Next, for each d in the RL3 list, if d also appears in the RL1 list, the score s_d^3 of d is modified to

$$s_d^3 = s_d^2 - \psi \cdot s_d^1 \,, \tag{4}$$

where ψ is called the *penalty coefficient*, and $0 \leq \psi \leq 1$. The setting $\psi = 0$ means that RL3 output is identical to that of RL2. When $\psi = 1$, the maximal penalty is set, and is equal to the score of d in RL1 output. Note that the penalty value is linearly and positively correlated to the RL1 scores, so the higher-evaluated documents in the RL1 list get larger penalty.

Submitted runs

For the Session Track, we altered our search system to allow the use of static PageRank score. In principle, the logarithm of PageRank scores was employed. Moreover, the PageRank scores were normalized locally for each query so that the maximal value of PageRank scores for the list of all documents that appear in either the result list of content-only search or anchor-only search is 1 (as in the case of normalized scores of the content-only and anchor-only results). When all content, anchor, and PageRank components are employed, the aggregated similarity score is calculated as

$$S_{d,q} = (1 - \alpha - \beta) \cdot S_{d,q}^{\mathrm{C}} + \alpha \cdot S_{d,q}^{\mathrm{A}} + \beta \cdot S_{d,q}^{\mathrm{P}} \,, \tag{5}$$

where $\alpha = \beta = 1$, $S_{d,q}^{\mathrm{C}}$, $S_{d,q}^{\mathrm{A}}$ are defined as in Formula 3, and $S_{d,q}^{\mathrm{P}}$ is the normalized pagerank score of d with respect to q.

We made use of three different run styles as listed below.

1. Style_A, characterized by the retrieval model IBM25, the use of content, anchor, and pagerank as defined by Formula 5, and the penalty coefficient $\psi = 0.25$ for the Formula 4.

Run	nsDCG@10		nsDCG_dupes@10		nDCG@10		
	RL12	*RL13*	*RL12*	*RL13*	*RL1*	*RL2*	*RL3*
Group A: `unimelb` submissions							
`Style_A`	**0.249**	0.221	**0.245**	**0.229**	**0.235**	**0.266**	0.178
`Style_B`	**0.249**	0.216	**0.245**	0.225	**0.235**	**0.266**	0.165
`Style_C`	0.214	0.189	0.217	0.198	0.201	0.236	0.165
Group D: Statistics from all TREC submissions							
max	**0.249**	**0.238**	**0.245**	**0.229**	**0.235**	**0.266**	**0.260**
median	0.204	0.178	0.207	0.187	0.189	0.214	0.170

Table 5: Effectiveness performance of the 2010 Session Track submissions. In all of the runs, the 30% spammiest pages according to the fusion scores have been discarded. In the header, *RL1*, *RL2*, and *RL3* accordingly refer to the score of `RL1`, `RL2`, and `RL3` output. See the Session Track's overview paper for the meaning of *RL12* and *RL13*.

2. `Style_B`, almost identical to `Style_A` except for the value of ψ, which is set to 1.

3. `Style_C`, a content-only run, with the similarity score defined using the `IMP` model, and ψ in Formula 4 is set to 0.25 as in the case of `Style_A`.

We submitted three runs – `UM10SibmA`, `UM10SibmbB`, and `UM10SimpA`, which followed exactly the three styles `Style_A`, `Style_B`, and `Style_C`, respectively. The performance of these submissions is listed in Table 5 under the umbrella of *G*roup A. The table also lists, under *Group D*, some statistics from all TREC submissions for this track.

Table 5 shows that, for our submissions, the runs that employ anchor text and pagerank in addition to the original content collection significantly outperform the content-only run. We also conducted another run (not reported here), which is content plus anchor as defined by Formula 3. Unfortunately, our post-TREC analysis showed that this run has a performance which is very close to that of `Style_A`, which means that our use of PageRank did not improve retrieval effectiveness. We will not make any further attempt in this report to track down the reasons for this failure.

The comparison of `Style_A` and `Style_B` in Table 5 reveals several interesting points with regards to the purpose of the Session Track and our approaches. First, the large change in the value of ψ between `Style_A` and `Style_B` did not bring too much difference in performance. Perhaps any value between 0.25 and 1.00 could give a similar effect.

Second, the policy of penalty actually hurt the retrieval performance, instead of improving it as we hoped for. And the larger is the penalty coefficient, the worse is the main system performance attribute, which is defined by the columns `RL13`. Perhaps:

- The supposition that `RL1` failed is not totally correct. Actually, it performed well relative to `RL2`.

- The supposition that users are patient enough to read all $2{,}000$ answers of `RL1` before deciding to issue the reformulated queries is (of course) incorrect. Perhaps, users are impatient enough to read only top-10 results (as endorsed by the track's guideline). That is, the penalty should be applied only to these top-10 documents, and ones that are demonstrably similar to them, and not to any other.

Post-TREC Experiments

Based on the above two arguments, we conducted a number of post-TREC experiments, by changing the way the `RL3` output is generated. No change was made to the `RL1` and `RL2` output. The

Run	nsDCG@10		nsDCG_dupes@10		nDCG@10		
	RL12	*RL13*	*RL12*	*RL13*	*RL1*	*RL2*	*RL3*
Group A: `unimelb` submissions, penalty to all items in *RL1*							
`Style_A`	**0.249**	0.221	**0.245**	0.229	**0.235**	**0.266**	0.178
Group B: Penalty applied only to the top-10 of *RL1*							
`Style_A`	**0.249**	0.241	**0.245**	**0.250**	**0.235**	**0.266**	0.243
`Style_B`	**0.249**	0.241	**0.245**	**0.250**	**0.235**	**0.266**	0.242
Group C: Rewards instead of penalty, and only to those in top-10 of *RL1*							
`Style_A`	**0.249**	**0.249**	**0.245**	0.245	**0.235**	**0.266**	**0.269**
`Style_B`	**0.249**	**0.249**	**0.245**	0.245	**0.235**	**0.266**	0.268
Group D: Statistics from all TREC submissions							
max	**0.249**	0.238	**0.245**	0.229	**0.235**	**0.266**	0.260

Table 6: Effectiveness performance of the post-TREC experiments for Session Track. Some figures from the official submissions and the Track's statistics are also include under *Group A* and *Group D*. In all of the runs, the 30% spammiest pages according to the fusion scores have been discarded.

additional experiments are reported in Table 5 under the labels of *Group B* and *Group C*. In *Group B* we limited the penalty only to the top-10 documents of the `RL1` output list, and hence all of other documents have scores as defined by the normalized scores of `RL2`. The table shows that this policy helps improve the performance of `RL3`, but the performance is still considerably worse than that of `RL2`. That means, penalty is likely not a good approach.

Experiments in *Group C* were designed to check the reverse policy. Instead of penalties, rewards were given to the top-10 documents of the `RL1` list. That is, the Formula 4 becomes

$$s_d^3 = s_d^2 + \psi \cdot s_d^1 \,. \tag{6}$$

The two styles `Style_A` and `Style_B` under the section *Group B* were conducted in this manner, and with no change to the ψ values. It can be seen that this time, `RL3` outperforms `RL2` in terms of accuracy, although the gap is modest. The small success of the reverse policy confirm that the policy is not worthy, at least in the context of this year's Session Track. The problem is, of course, the relatively good initial performance of `RL1`.

Overall, our submissions had excellent performance in terms of effectiveness of the Ad-hoc Task, but failed to address the main criterion of the Track – the *RL13* scores. It is, however, difficult to analyze the reasons for this failure. On the one hand, it can be said that our methodology, especially the policy of applying a penalty to documents in the `RL1` output, is unsupported, and we need to seek alternative approaches in order to have better performance for `RL3`. On the other hand, the failure is partly due to the assumption that the `RL1` output is poor, and that the query `RL2` is much better than the query `RL1`. None of these two assumptions is correct for this year. Given that over the whole Track the maximal score for `RL3` is *lower* than that for `RL2`, we unfortunately face an uncertain question of whether the settings for this year's Track were appropriate.

6 Technical Notes

The experiments described in this paper were conducted using a HPC cluster located at RMIT University. Various parameters of the system are listed at `http://its-ru-hpc-mgmt.cs.rmit.edu.au/doku.php?id=rmit_hpc_specifications`. In short, the cluster consists of 34 machines and a storage unit. Each machine has eight 2.3 GHz CPUs and 32 GB RAM. We share the system

with other users. For simplicity, we use the word "node" to refer to a CPU, not a physical machine. Our system employed 32 nodes for both indexing and querying. These 32 nodes normally belonged to only a few machines, but we were not able to choose or specify particular nodes or machines, or number of nodes per machine.

For a document collection, the principal component of its index is the inverted file, where each distinct term of the collection is associated with an inverted list. We made use of an *impact-sorted* index. The impact-sorted inverted list for a term t is a list of equal-impact blocks. Each block represents one distinct impact value k, and contains the sequence of document numbers in which t appears and has an impact score of k. Inside a block, document numbers are arranged in increasing order, to facilitate compression. The blocks are arranged in decreasing order of associated impacts, so as to support effective pruning.

Compression is applied to inverted files. In all of our experiments the word-synchronized compression scheme *Simple8* [Anh and Moffat, 2010] was used for inverted list compression. This method provides a good balance between index space and decoding speed, and is especially good for skipping operation.

For the content-only collection, the wall-clock time for indexing was approximately 16 hours. Note that this time included time for creating a fully positional index, and then extracting a working, non-positional index. The query time was not recorded because of the small number of queries.

7 Conclusions

TREC-2009 and TREC-2010 marked the first time our team, as well as many other teams, worked with a large text collection of over ten terabytes. We managed to perform the tasks in a reasonable time for both indexing and querying. In terms of effectiveness performance, we got good results. Given that all of our runs did not rely on any external resources like external databases or commercial search engines, the results were quite encouraging. That shows that the impact-based retrieval is competitive, despite of the fact that it is simple and does not involve tuning parameters.

Across different tasks in the two years, we noticed the crucial role of the anchor text collection for effective retrieval. In one of the tasks, the use of anchor text alone yields the performance better than that of content alone, and almost as good as using both content and anchor text. In all tasks, the use of anchor text in addition to the content significantly improve the effectiveness performance.

There are a number of problems need to be addressed in the upcoming TRECs, including finding the way to effectively employ PageRank scores; designing new, more effective, approaches for the Session Track; dealing with the specific features of the Diversity Task; and improving both effectiveness and efficiency of the retrieval model.

Acknowledgement This work was supported by the Australian Research Council. We thank RMIT University (Australia) for letting us to use their high-performance computing cluster for all the experiments described in this report.

References

V. Anh and A. Moffat. Simplified similarity scoring using term ranks. In G. Marchionini, A. Moffat, J. Tait, R. Baeza-Yates, and N. Ziviani, editors, *Proc. 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 226–233, New York, NY, USA, August 2005. ACM Press, New York.

V. Anh and A. Moffat. Index compression using 64-bit words. *Software – Practice & Experience*, 40(2):131–147, 2010. Source code available from www.cs.mu.oz.au/ alistair/coders-64bit/.

V. Anh, R. Wan, and A. Moffat. Term impacts as normalized term frequencies for BM25 similarity scoring. In A. Amir, A. Turpin, and A. Moffat, editors, *Proc. 15th Int. Symp. String Processing and Information Retrieval*, pages 51–62, Melbourne, Australia, November 2008. LNCS 5280, Springer. URL `http://dx.doi.org/10.1007/978-3-540-89097-3_7`.

G. Cormack, M. Smucker, and C. Clarke. Efficient and effective spam filtering and re-ranking for large web datasets. srXiv:1004.5168, 2010.

S. Robertson, S. Walker, S. Jones, M. Hancock-Beaulieu, and M. Gatford. Okapi at TREC–3. In D. Harman, editor, *Proc. Third Text REtrieval Conference (TREC–3)*, pages 109–126, Gaithersburg, MD, November 1994. National Institute of Standards and Technology (Special Publication 500-225). URL `http://potomac.ncsl.nist.gov:80/TREC/t3_proceedings.html`.

G. Salton. *Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer*. Addison Wesley, Reading, Massachusetts, 1989.