Approaches to Robust and Web Retrieval

Jaap Kamps Christof Monz[*] Maarten de Rijke Börkur Sigurbjörnsson
Language & Inference Technology Group
University of Amsterdam
`http://lit.science.uva.nl/`

Abstract: We describe our participation in the TREC 2003 Robust and Web tracks. For the Robust track, we experimented with the impact of stemming and feedback on the worst scoring topics. Our main finding is the effectiveness of stemming on poorly performing topics, which sheds new light on the role of morphological normalization in information retrieval. For both the home/named page finding and topic distillation tasks of the Web track, we experimented with different document representations and retrieval models. Our main finding is effectiveness of the anchor text index for both tasks, suggesting that compact document representations are a fruitful strategy for scaling-up retrieval systems.

1 Introduction

This year, our aim for the Web track was to experiment with different document representations and retrieval models for the home/named page finding and topic distillation tasks. The Robust track was new in 2003; our aim here was to investigate the impact of blind feedback and stemming on poorly performing topics.

For both tracks, our experiments exploited the homegrown FlexIR document retrieval system [9]. The main goal underlying FlexIR's design is to facilitate flexible experimentation with a wide variety of retrieval components and techniques. FlexIR is implemented in Perl, and supports many types of pre-processing, scoring, indexing, and term-weighting methods.

The rest of this paper is organized as follows. In two (largely self-contained) sections we describe our work for the Robust and Web tracks. Finally, we summarize our findings in a concluding section.

2 Robust Track

After describing the experimental setup for this track, we discuss our runs investigating the impact of blind feedback and stemming on the poorly performing topics.

System Description

All Robust track runs use the FlexIR information retrieval system. We employ a number of techniques:

Tokenization We remove punctuation marks, apply case-folding, and map marked characters into the unmarked tokens. We either index the words themselves, or the stems of the words. We use the Snowball stemming algorithm [13]. Snowball is a small string processing language designed for creating stemming algorithms for use in information retrieval

Retrieval model We use a multinominal language model with Jelinek-Mercer smoothing [4]. For all robust track runs, we use a uniform query term importance weight of 0.15.

Blind feedback Term weights are recomputed by using the standard Rocchio method [12], where we consider the top 10 documents to be relevant and doc-

[*] Present address: Institute for Advanced Computer Studies, University of Maryland, 3161 A.V. Williams Building, College Park, MD 20742, USA. Email: `christof@umiacs.umd.edu`.

uments ranked 501–1000 to be non-relevant. We allow at most 20 terms to be added to the original query.

Runs

We conduct two sets of experiments using (1) only the description field of the topics (D-topics), or (2) both the title and description fields (TD-topics). Using the resulting queries, we constructed the following four runs:

Words Language model run on a word-based index. This runs serves as the baseline for our stemming and feedback experiments.

Words+feedback Language model run on a word-based index, using Rocchio blind feedback.

Stems Language model run on the Snowball stemmed index.

Stems+feedback Language model run on the Snowball stemmed index, using Rocchio blind feedback.

Results

Table 1 gives the results of the runs over all 100 robust topics (best scores in boldface). The second column shows the mean average precision, the third the precision at 10 documents, the fourth the percentage of topics with no relevant document in the top 10; the fifth shows the area underneath the MAP(X) versus X curve for the worst 25 topics.

Table 1: Results for the Robust track (D top and TD bottom).

Run identifier	MAP	Prec.10	NoTop10	MAP(X)
Words	0.2065	0.3530	15.0%	0.0076
Words+feedback	0.1970	0.3420	17.0%	0.0059
Stems	**0.2319**	**0.3960**	**14.0%**	**0.0126**
Stems+feedback	0.2068	0.3570	16.0%	0.0098
Words	0.2324	0.4050	9.0%	0.0216
Words+feedback	**0.2452**	0.4110	13.0%	0.0210
Stems	0.2450	**0.4150**	**6.0**%	0.0256
Stems+feedback	0.2373	0.4040	14.0%	**0.0273**

The results of blind feedback are mixed at best. On the one hand feedback helps the overall score for the runs using TD-topics, with a best precision at 10 and a best score for mean average precision. On the other hand feedback hurts the performance on the worst scoring topics. For the runs using D-topics, feedback deteriorates scoring on all measures.

We can regard the T-field of the topics as a “gold standard” experiment on query expansion. If we compare the score of runs using TD-topics with the scores of runs using D-topics, we see an improvement on all measures and runs. In particular the improvement on the weak-scoring topic measures is substantial.

The results for Snowball stemming are positive overall. Stemming helps both the overall performance, with a best score for precision at 10, as well as the performance of the worst scoring topics, with a best score for the percentage of topics with a top 10 relevant document. For runs using D-topics, stemming gives the best score for all measures. The use of both stemming and feedback gives the best score for the area under the MAP(X) curve for the runs using TD-topics, but does not promote performance on the other measures.

We also break down the score over the 50 old topics (in Table 2) and the 50 new topics (in Table 3). Note that the area underneath MAP(X) versus X curve (in the last column) is now calculated for the worst 12 topics. For both the old and new topics, the effectiveness of feedback and stemming is comparable to the effectiveness on all topics. There is, however, a striking difference in the performance between the two types of topics: the new topics

Table 2: Results for the old topics (D top and TD bottom).

Run identifier	MAP	Prec.10	NoTop10	MAP(X)
Words	0.1066	0.2640	**14.0%**	0.0064
Words+feedback	0.0969	0.2460	20.0%	0.0039
Stems	**0.1164**	**0.3020**	18.0%	**0.0108**
Stems+feedback	0.1065	0.2640	18.0%	0.0085
Words	0.1349	0.3180	12.0%	0.0142
Words+feedback	**0.1377**	0.3200	16.0%	0.0143
Stems	0.1327	**0.3300**	**6.0**%	0.0185
Stems+feedback	0.1361	**0.3300**	16.0%	**0.0204**

Table 3: Results for the new topics (D top and TD bottom).

Run identifier	MAP	Prec.10	NoTop10	MAP(X)
Words	0.3064	0.4420	16.0%	0.0142
Words+feedback	0.2971	0.4380	14.0%	0.0105
Stems	**0.3475**	**0.4900**	**10.0%**	**0.0294**
Stems+feedback	0.3071	0.4500	14.0%	0.0216
Words	0.3300	0.4920	**6.0**%	0.0433
Words+feedback	0.3528	**0.5020**	10.0%	0.0368
Stems	**0.3572**	0.5000	**6.0**%	**0.0551**
Stems+feedback	0.3386	0.4780	12.0%	0.0478

give a much higher mean average precision score. This is an obvious consequence of the way the old topics were selected for inclusion in this year's Robust track. As a result, the worst topic measures are dominated by the old topics.

3 Web Track

After describing our experimental setup for this track, we discuss our runs for the home/named page finding task (known-item search), followed by the runs for the topic distillation task (key resource search).

System Description

All Web track runs use the FlexIR information retrieval system. We employ a number of techniques:

Document representation We create indexes for (1) the full documents, (2) the text in the title tags, (3) the anchor texts pointing toward the document. For the anchor texts index, we unfold relative links and normalize URLs, and do not index repeated occurrences of the same anchor text [10].

Tokenization We remove HTML-tags, punctuation marks, apply case-folding, and map marked characters into the unmarked tokens. We either index the free-text without further processing, or use the Snowball stemming algorithm [13].

Retrieval model We use three retrieval models. First, a statistical language model [4] with a uniform query term importance weight of either 0.35 or 0.70. Second, the Okapi weighting scheme [11] with tuning parameters $k = 1.5$ and $b = 0.8$. Third, the `Lnu.ltc` weighting scheme [1] with *slope* at 0.1 or 0.2; the pivot was set to the average number of unique words per document.

Combination We use the standard combination methods such as CombSUM and CombMAX [3], or weighted fusion [14]. We combine either full length runs, or limit the combination to the top n results. Unless indicated otherwise, we normalize the scores before combining them.

Minimal span weighting We calculate a minimally matching span for each document. Intuitively, a minimal matching span is the smallest text excerpt from a document that contains all terms which occur in the query and the document. Minimal span weighting depends on three factors (for details, see [2, 5, 8]).

1. *document similarity*: The document similarity is computed for the whole document, i.e., positional information is not taken into account. Similarity scores are normalized with respect to the maximal similarity score for a query.
2. *span size ratio*: The span size ratio is the number of unique matching terms in the span over the total number of tokens in the span.
3. *matching term ratio*: The matching term ratio is the number of unique matching terms over the number of unique terms in the query, after stop word removal.

In two separate sections, we will now address our runs and results for the home/named page finding task, and the topic distillation task.

3.1 Home/Named Page Finding Task

Runs

We submitted the following five official runs for the home/named page finding task:

UAmsT03WnOWS CombSUM of top 1000 of Okapi on word-based and stemmed full document indexes.

UAmsT03WnLM Language model run ($\lambda = 0.70$) on word-based full document index.

UAmsT03WnLn3 CombMAX on the top 25 of `Lnu.ltc` runs ($slope = 0.2$) on the three stemmed indexes: full documents, titles, and anchor texts.

UAmsT03WnLM3 Weighted fusion of language model runs ($\lambda = 0.70$) on the three word-based indexes: 0.7 full documents, 0.2 titles, and 0.1 anchor texts.

UAmsT03WnMSW Minimal span weighting based on the `Lnu.ltc` run ($slope = 0.1$) on the stemmed full document index.

Results

The results of the official runs for the home/named page finding task are shown in Table 4 (best scores in boldface). The second column gives the mean reciprocal rank, the third the number and percentage of topics with a relevant document in the top 10, the fourth the number and percentage of topics for which no relevant document is found (in the top 50). The language model run combining the non-stemmed documents, titles, and anchors scores best with an average reciprocal rank of 0.5185. The `Lnu.ltc` weighted combination of the three stemmed indexes scores second best.

Table 4: Results for home/named page finding.

Run identifier	MRR	Top 10	not found
`UAmsT03WnOWS`	0.3833	178 (59.3%)	70 (23.3%)
`UAmsT03WnLM`	0.3592	170 (56.7%)	81 (27.0%)
`UAmsT03WnLn3`	0.4982	**218** (72.7%)	**38** (12.7%)
`UAmsT03WnLM3`	**0.5185**	214 (71.3%)	46 (15.3%)
`UAmsT03WnMSW`	0.4073	189 (63.0%)	64 (21.3%)

Table 5 shows the mean average precision of the base runs used in combinations for our official runs. All `Lnu.ltc` runs use a slope of 0.2, and all language model runs use a uniform term weight of 0.70. Here, we retrieve up to 1,000 documents per topic, leading to slightly higher MRRs than the official runs using a maximum of 50 documents. We see an interesting difference between the three retrieval models: where the `Lnu.ltc` and Okapi models score best on the full document representation, the language model runs on the anchor text index score more than 20% better than the runs on the full document index. In fact, our best score on a single index is on the language model run on the non-stemmed anchor text index. There is no clear benefit of the use of a stemming algorithm on the mean reciprocal ranks: stemming improves the score for four out of the nine comparative runs.

Table 5: MRR for home/named page finding base runs.

Index type		`Lnu.ltc`	Okapi	LM
Documents	Words	0.3750	0.3795	0.3604
	Stems	0.3697	0.3833	0.3616
Titles	Words	0.2339	0.3421	0.3536
	Stems	0.3655	0.3334	0.3487
Anchors	Words	0.3068	0.3593	**0.4436**
	Stems	0.2934	0.3379	0.4278

There is another interesting difference between the retrieval models, which has to do with combination. The combination of Okapi runs on the document stems and words, `UAmsT03WnOWS`, does not improve over document stems run. The combination of the three stemmed `Lnu.ltc` runs, run `UAmsT03WnLn3`, does improve 34.8% over the best scoring stemmed runs. The combination of the three non-stemmed language model runs, `UAmsT03WnLM3`, improves 16.9% over the best scoring base runs. Finally, the run using the matching-span weighting uses a `Lnu.ltc` full document base run with a different slope of 0.1 scoring a MRR of 0.2742. The resulting run, `UAmsT03WnMSW`, improves no less than 48.5% over the underlying base run.

Table 6: Results for home page topics.

Run identifier	MRR	Top 10	not found
`UAmsT03WnOWS`	0.2567	67 (44.7%)	55 (36.7%)
`UAmsT03WnLM`	0.2462	64 (42.7%)	60 (40.0%)
`UAmsT03WnLn3`	0.4105	97 (64.7%)	**26** (17.3%)
`UAmsT03WnLM3`	**0.4402**	**101** (67.3%)	33 (22.0%)
`UAmsT03WnMSW`	0.2708	73 (48.7%)	53 (35.3%)

We also break down the score over the 150 home page topics (in Table 6) and the 150 named page topics (in Table 7). Here we see a much better performance on the named page topics. This is perhaps unexpected because named page finding is conceived to be a more difficult task than home page finding. The simple explanation is that we decided not to apply special home page finding strategies. Although techniques like slash-counts or URL priors are effective for home page finding [7], they seem to hurt the named page topics considerably. Even without a particular home page bias, home pages can be retrieved with reasonable effectiveness, as is witnessed by our results for the home page topics in Table 6.

Table 7: Results for named page topics.

Run identifier	MRR	Top 10	not found
`UAmsT03WnOWS`	0.5098	111 (74.0%)	15 (10.0%)
`UAmsT03WnLM`	0.4721	106 (70.7%)	21 (14.0%)
`UAmsT03WnLn3`	0.5859	**121** (80.7%)	12 (8.0%)
`UAmsT03WnLM3`	**0.5969**	113 (75.3%)	13 (8.7%)
`UAmsT03WnMSW`	0.5438	116 (77.3%)	**11** (7.3%)

3.2 Topic Distillation Task

Runs

We submitted the following five official runs for the topic distillation task:

UAmsT03WtOk3 Weighted fusion of Okapi runs on the three stemmed indexes: 0.7 full documents, 0.2 titles; and 0.1 anchor texts.

UAmsT03WtLM3 Weighted fusion of language model runs on the three stemmed indexes: 0.7 full documents ($\lambda = 0.35$), 0.2 titles ($\lambda = 0.7$), and 0.1 anchor texts ($\lambda = 0.7$). We combine the probabilities without normalization.

UAmsT03WtOkI Weighted fusion of 0.9 Okapi run on the stemmed full document index with 0.1 of a link topology measure. We applied the realized indegree on the top 10 documents [10]. This is a variant of HITS [6] where we consider the fraction of inlinks that is in the local set—roughly a `tf·idf` measure for link topology.

UAmsT03WtLMI Weighted fusion of 0.9 language model run ($\lambda = 0.35$) on the stemmed full document index with 0.1 of the realized indegree of the top 10 documents.

UAmsT03WtOkC Weighted fusion of 0.8 Okapi run on the stemmed full document index with 0.2 of a URL-based reranking. The reranking was done by clustering the found pages by their base URLs, and to only return the page with the lowest slash-count per cluster.

Results

The results of the official runs for the topic distillation task are shown in Table 8 (best scores in boldface). The second column shows the mean average precision, the third to fifth columns show the precision at 10, 20, and 30 documents, respectively. The best score is obtained by `UAmsT03WtOk3`, the fusion of Okapi runs on the three stemmed indexes. The second best score is obtained by `UAmsT03WtOkC`, a URL-based clustering of the Okapi full documents run. Before discussing the results of our experiments, we first evaluate the results of the runs used to create our official runs.

Table 8: Results for topic distillation.

Run identifier	MAP	Prec. at 10, 20, 30		
`UAmsT03WtOk3`	**0.1344**	**0.0980**	**0.0810**	**0.0787**
`UAmsT03WtLM3`	0.1019	0.0840	0.0630	0.0533
`UAmsT03WtOkI`	0.0862	0.0760	0.0660	0.0567
`UAmsT03WtLMI`	0.0412	0.0280	0.0260	0.0267
`UAmsT03WtOkC`	0.1127	0.0860	0.0650	0.0540

Table 9 shows the results of the base runs used in combination for our official runs. All these runs use the Snowball stemming algorithm [13]. We see a remarkable divergence between the scoring for Okapi and the language model. The Okapi model performs comparable on all the three indexes, documents, titles, and anchors. The language model performs poorly on the document and title indexes, but excels for the anchor text index. The combination of the three Okapi runs, `UAmsT03WtOk3`, improves significantly over the best underlying run (MAP +38.4%, Precision at 10 +25.6%). The combination of language model runs, `UAmsT03WtLM3`, uses far from optimal relative weights and, as a result, does not improve over the anchor text run. The runs using the hyperlink graph topology do not result in significant improvement. The Okapi run `UAmsT03WtOkI` slightly improves its precision at 10 over the document run; whereas the language model run `UAmsT03WtLMI` slightly decreases its precision at 10 over the document run. Finally, the Okapi run clustering per base URL, `UAmsT03WtOkC`, does improve over the Okapi document run (MAP +25.1%, Precision at 10 +16.2%).

Table 9: Results for topic distillation stemmed base runs.

Run type	MAP	Prec. at 10, 20, 30		
Doc. Okapi	0.0901	0.0740	0.0580	**0.0527**
Title Okapi	0.0870	0.0780	**0.0590**	0.0453
Anchor Okapi	0.0971	0.0780	0.0560	0.0493
Doc. LM (0.35)	0.0386	0.0300	0.0320	0.0293
Title LM (0.70)	0.0434	0.0480	0.0360	0.0293
Anchor LM (0.70)	**0.1068**	**0.0860**	0.0560	0.0473

4 Conclusions

In this paper we have described our participation in the TREC 2003 Robust and Web tracks.

For the Robust track, we experimented with the impact of stemming and feedback on the worst scoring topics. Our results suggest that blind feedback can help overall performance but does not increase the effectiveness on the lowest scoring topics. Our results also suggest that applying a stemming algorithm does benefit both the overall performance, as well as the performance of the worst scoring topics. This result sheds some new light on the role of morphological normalization in information retrieval.

For the Web track, we saw very similar results for both the home/named page finding task and the topic distillation task. Using the hyperlinks in the collection for creating an anchor text index turns out to be very effective. Also, the use of HTML-structure in the documents to elicit their titles turns out to be effective. Combining these alternative document representations with a standard document index led to our best scores for both tasks.

A further general observation is the effectiveness of compact document representations, such as indexing only document titles, or only anchor texts pointing toward documents. These compact document representations result in performance that meets or exceeds the performance of a massive full document text index. This result suggests that it is feasible to create effective retrieval indexes for even larger web collections, provided that the appropriate document representation is chosen.

Acknowledgments Jaap Kamps was supported by NWO under project numbers 400-20-036 and 612.066.302. Christof Monz was supported by NWO under project numbers 612-13-001 and 220-80-001. Maarten de Rijke was supported by NWO under project numbers 612-13-001, 365-20-005, 612.069.006, 612.000.106, 220-80-001, 612.000.207, and 612.066.302.

References

[1] C. Buckley, A. Singhal, and M. Mitra. New retrieval approaches using SMART: TREC 4. In D. Harman, editor, *The Fourth Text REtrieval Conference (TREC-4)*, pages 25–48. National Institute for Standards and Technology. NIST Special Publication 500-236, 1996.

[2] C. Clarke, G. Cormack, and T. Lynam. Exploiting redundancy in question answering. In D. H. Kraft, W. B. Croft, D. J. Harper, and J. Zobel, editors, *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 358–365. ACM Press, New York NY, USA, 2001.

[3] E. Fox and J. Shaw. Combination of multiple searches. In D. Harman, editor, *The Second Text REtrieval Conference (TREC-2)*, pages 243–252. National Institute for Standards and Technology. NIST Special Publication 500-215, 1994.

[4] D. Hiemstra. *Using Language Models for Information Retrieval*. PhD thesis, Center for Telematics and Information Technology, University of Twente, 2001.

[5] V. Jijkoun, G. Mishne, C. Monz, M. de Rijke, S. Schlobach, and O. Tsur. The University of Amsterdam at the TREC 2003 question answering track. In *The Twelfth Text REtrieval Conference (TREC 2003)*. National Institute for Standards and Technology, 2004.

[6] J. M. Kleinberg. Authoritative structures in a hyperlinked environment. *Journal of the ACM*, 46:604–632, 1999.

[7] W. Kraaij, T. Westerveld, and D. Hiemstra. The importance of prior probabilities for entry page search. In K. Järvelin, M. Beaulieu, R. Baeza-Yates, and S. H. Myaeng, editors, *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 27–34. ACM Press, New York NY, USA, 2002.

[8] C. Monz. *From Document Retrieval to Question Answering*. ILLC dissertation series 2003-04, University of Amsterdam, 2003.

[9] C. Monz and M. de Rijke. Shallow morphological analysis in monolingual information retrieval for Dutch, German and Italian. In C. Peters, M. Braschler, J. Gonzalo, and M. Kluck, editors, *Evaluation of Cross-Language Information Retrieval Systems, CLEF 2001*, volume 2406 of *Lecture Notes in Computer Science*, pages 262–277. Springer, 2002.

[10] C. Monz, J. Kamps, and M. de Rijke. The University of Amsterdam at TREC 2002. In E. M. Voorhees and L. P. Buckland, editors, *The Eleventh Text REtrieval Conference (TREC 2002)*, pages 603–614. National Institute for Standards and Technology. NIST Special Publication 500-251, 2003.

[11] S. Robertson, S. Walker, and M. Beaulieu. Experimentation as a way of life: Okapi at TREC. *Information Processing & Management*, 36:95–108, 2000.

[12] J. Rocchio, Jr. Relevance feedback in information retrieval. In G. Salton, editor, *The SMART Retrieval System: Experiments in Automatic Document Processing*, Prentice-Hall Series in Automatic Computation, chapter 14, pages 313–323. Prentice-Hall, Englewood Cliffs NJ, 1971.

[13] Snowball. Stemming algorithms for use in information retrieval, 2003. `http://www.snowball.tartarus.org/`.

[14] C. C. Vogt and G. W. Cottrell. Predicting the performance of linearly combined IR systems. In W. B. Croft, A. Moffat, C. J. van Rijsbergen, R. Wilkinson, and J. Zobel, editors, *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 190–196. ACM Press, New York NY, USA, 1998.