# The Technion at TREC 2013 Web Track: Cluster-based Document Retrieval

Fiana Raiber and Oren Kurland

Faculty of Industrial Engineering and Management
Technion – Israel Institute of Technology
Haifa 32000, Israel
fiana@tx.technion.ac.il, kurland@ie.technion.ac.il

**Abstract**

Many cluster-based document retrieval methods have been proposed over the years. In our submissions to the ad hoc task of the TREC 2013 Web Track we experimented with one such highly effective method. Empirical results demonstrate the effectiveness of using our approach; specifically, with respect to other submitted runs.

## 1 Introduction

Our submissions to the ad hoc task of TREC's 2013 Web Track were based on a cluster-based document retrieval approach. We used the recently proposed ClustMRF method [7] to rank the document clusters created from the documents most highly ranked by an initial search. The cluster ranking was then transformed to a ranking over documents. ClustMRF utilizes Markov Random Fields, which enable the integration of different types of cluster-relevance evidence. For the initial search, we used several query-independent document quality measures [2]. These measures are also naturally integrated in ClustMRF.

The empirical evaluation shows that using our cluster-based approach yields performance that is substantially better than that of the initial search. Furthermore, we show that the performance of our runs was above the median in the ad hoc Web track.

## 2 Retrieval Approach

We applied the following steps to produce three submitted runs. First, we ranked all the documents in the corpus to create an initial list of $10,000$ documents[1]. Next, we re-ranked the top $1,000$ documents in the initial list using a learning-to-rank approach; the remaining $9,000$ documents maintained their original positions. We then re-ranked the top 50 documents in the list produced in the previous step using ClustMRF [7]. Finally, the ranking

[1] The requirement of TREC was to submit runs of $10,000$ documents.

created using ClustMRF was used to diversify search results. The details of the methods used in each step are provided below.

**Creating the initial list and removing spam documents.** The Markov Random Field (MRF) method with the sequential dependence model [6] was used to rank all the documents in the corpus. Documents that were assigned with a score below 50 by Waterloo's spam classifier [4] were then removed from the ranking, and the top $10,000$ remaining documents were used to create the initial list $\mathcal{D}_{\text{init}}$.

**Learning-to-rank.** The top $1,000$ documents in $\mathcal{D}_{\text{init}}$ were re-ranked using SVM$^{rank}$ [5], applied with default parameter values. Most of the 130 features that we used are based on those used in Microsoft's learning-to-rank datasets[2,3]. The rest of the features are query-independent document quality measures which were shown to be highly effective for Web retrieval [2]. These features include the ratio between the number of stopwords and non-stopwords in a document, the percentage of stopwords in a stopwords list that appear in the document, the entropy of the term distribution in a document, and the score assigned to a document by Waterloo's spam classifier [4]. Except for the latter, all the features were computed separately for the entire document, its body, title, URL and anchor text. The resultant model is henceforth referred to as **LTR**.

**Using document clusters.** As the next step, the 50 most highly ranked documents by LTR were clustered using the nearest-neighbor clustering approach. The recently proposed **ClustMRF** method [7] was used to rank these clusters based on their presumed relevance to the query. The score assigned to a document by LTR served for the document-query similarity measure that is used in ClustMRF. Finally, the ranking of clusters was transformed to a ranking of documents by replacing each cluster with its constituent documents while omitting repeats; the order of documents within a cluster was determined by the positions of the documents in the ranking produced by LTR. The remaining 950 documents in the LTR result list retained their original positions. Additional technical details are provided in Section 3.1.

**Diversifying search results.** To diversify search results, the top 50 documents in the ranking created by LTR were first re-ranked using ClustMRF as described above. The same 50 documents were then re-ranked using **MMR+ClustMRF**, a variant of the MMR method [3, 7]. The similarity between the query and the document in MMR+ClustMRF was estimated using the rank of a document in the result list produced by ClustMRF [7].

[2] www.research.microsoft.com/en-us/projects/mslr

[3] A few features were not considered in our experiments. These include the Boolean Model, Vector Space Model, LMIR.ABS, Outlink number, SiteRank, QualityScore, QualityScore2, Query-URL click count, URL click count, and URL dwell time.

# 3 Evaluation

## 3.1 Experimental setup

We submitted runs both for the full ClueWeb12 dataset and for its Category B subset, hereafter denoted **CatA** and **CatB**, respectively[4].

The Indri toolkit[5] was used for experiments. Queries and documents were stemmed using the Krovetz stemmer. The INQUERY stopword list [1] was used to remove stopwords from queries, but not from documents; and, to compute the two stopwords-based features used by LTR.

The IDs of the runs which were submitted are *clustmrfbf* and *clustmrfaf*, where the ClustMRF method was implemented for CatB and CatA, respectively, and *mmrbf* where the MMR+ClustMRF method was implemented for CatB.

**Free-parameter values.** The free-parameter values of LTR, ClustMRF and MMR+ClustMRF were learned using the Category B of the ClueWeb09 dataset with queries 1-200 from TREC 2009-2012.

The values of $\lambda_T$, $\lambda_O$, and $\lambda_U$, the three free parameters of the MRF model that was used to create $\mathcal{D}_{\text{init}}$, were set to 0.85, 0.1, and 0.05, respectively, following previous recommendations [6]. The Dirichlet smoothing parameter that was used to create $\mathcal{D}_{\text{init}}$ and to compute the LMIR.DIR feature for LTR was set to 1000. The BM25 feature used by LTR was computed with $k1$=1 and $b$=0.5; LMIR.JM was computed with $\lambda$=0.1. SVM$^{rank}$ was used to learn feature weights in ClustMRF; NDCG@$k$ of the $k$ documents in a cluster was used as the score of a cluster. For each cluster size $k$ ($\in \{5, 10, 20\}$) a ranking of documents was created using ClustMRF; the value of $k$ was selected to optimize MAP@50 of the resultant document ranking. The interpolation parameter of MMR+ClustMRF, which controls the tradeoff between the relevance and diversity estimates, was set to a value in $\{0.1, 0.2, \ldots, 0.9\}$ to optimize ERR-IA@20. All other implementation details are the same as in Raiber and Kurland [7].

The two-tailed paired t-test with $p \leq 0.05$ was used for testing statistical significance of performance differences.

## 3.2 Experimental results

### 3.2.1 Main result

The results are presented in Table 1. We can see that LTR outperforms the initial ranking (Init), which was induced using standard MRF, in all relevant comparisons. Yet, LTR is always outperformed by ClustMRF for both CatA and CatB; the differences are statistically significant in the majority of the cases. This finding is consistent with previous work which showed that ClustMRF can be used to improve the performance of various result lists that are produced by effective retrieval methods [7].

[4] For consistency with the results published by TREC, the evaluation for both CatA and CatB was performed using the qrels of the full ClueWeb12 dataset.

[5] www.lemurproject.org/indri

| | | MAP@1000 | NDCG@20 | ERR@20 | P-IA@20 | $\alpha$-NDCG@20 | ERR-IA@20 |
|---|---|---|---|---|---|---|---|
| CatA | Init | 14.0 | 20.3 | 13.1 | 29.1 | 57.7 | 50.7 |
| | LTR | $18.1^{i}$ | $26.8^{i}$ | $16.7^{i}$ | $35.1^{i}$ | 63.6 | 54.5 |
| | ClustMRF$^{\star}$ | $\mathbf{19.3}^{i}_{l}$ | $\mathbf{31.0}^{i}_{l}$ | $\mathbf{18.4}^{i}_{l}$ | $\mathbf{42.5}^{i}_{l}$ | $66.8^{i}$ | 56.7 |
| | MMR+ClustMRF | $18.3^{i}_{c}$ | $29.0^{i}_{c}$ | $18.2^{i}_{l}$ | $38.0^{i}_{c}$ | $\mathbf{67.1}^{i}_{l}$ | **57.0** |
| CatB | Init | 3.5 | 11.9 | 8.3 | 18.6 | 50.3 | 42.3 |
| | LTR | $4.8^{i}$ | $15.3^{i}$ | 10.1 | $22.5^{i}$ | 55.5 | 46.3 |
| | ClustMRF$^{\star}$ | $\mathbf{6.0}^{i}_{l}$ | $\mathbf{19.8}^{i}_{l}$ | $12.3^{i}_{l}$ | $\mathbf{29.2}^{i}_{l}$ | $58.6^{i}$ | 50.1 |
| | MMR+ClustMRF$^{\star}$ | $5.9^{i}_{lc}$ | $19.4^{i}_{l}$ | $\mathbf{12.4}^{i}_{l}$ | $27.9^{i}_{l}$ | $\mathbf{59.0}^{i}$ | $\mathbf{50.5}^{i}_{l}$ |

Table 1: Main result. The best result for an experimental setting (CatA/CatB) and evaluation metric is boldfaced. '*i*', '*l*', and '*c*' mark statistically significant differences with the initial ranking (Init), LTR and ClustMRF, respectively. The three submitted runs are marked with '$\star$'.

| NDCG@20 | ERR@20 | P-IA@20 | $\alpha$-NDCG@20 | ERR-IA@20 |
|---|---|---|---|---|
| 90 | 88 | 92 | 82 | 80 |

Table 2: The percentage of queries for which the performance of the ClustMRF run for CatA (*clustmrfaf*) is better than or equal to the median performance attained by other TREC runs.

We can also see that, in general, ClustMRF is the most effective method for MAP@1000 and NDCG@20, while MMR+ClustMRF is the most effective in terms of $\alpha$-NDCG@20 and ERR-IA@20. This finding can be attributed to the fact that the value of the interpolation parameter in MMR+ClustMRF was set by optimizing ERR-IA@20, whereas in ClustMRF the size of clusters was selected by optimizing MAP@50 and the weights of feature were learned by maximizing relevance and not diversity. All in all, the findings presented above attest to the high effectiveness of using document clusters to re-rank the documents in an (effective) initial result list.

### 3.2.2 Comparison with other TREC runs

We next compare the performance of ClustMRF with that attained by other runs submitted to TREC. To that end, we computed the percentage of queries for which the performance attained by ClustMRF was higher or equal to the median performance attained by other runs. The results for CatA are presented in Table 2. We can see that ClustMRF was at least as effective as the median run for about 90% of the queries for NDCG@20, ERR@20, and P-IA@20, and for about 80% of the queries for $\alpha$-NDCG@20 and ERR-IA@20. Overall, we can conclude that the performance of ClustMRF was above the median.

## 4 Conclusions

We described our submissions to the ad hoc task of TREC's 2013 Web Track. We used a learning-to-rank approach to re-rank an initially retrieved list. Clusters of similar documents created from the top ranked documents in that re-ranked list were ranked using ClustMRF

[7]. The ranking of clusters was then transformed to a ranking of documents. In addition, to diversify the results, the document ranking produced by ClustMRF was used to estimate the similarity between the query and a document in MMR [3, 7]. Empirical results demonstrate the effectiveness of using ClustMRF to re-rank the top ranked documents in a (highly effective) result list.

## 5 Acknowledgments

This work has been supported by and carried out at the Technion-Microsoft Electronic Commerce Research Center. This work has also been supported in part by Microsoft Research through its Ph.D. Scholarship Program.

## References

[1] J. Allan, M. E. Connell, W. B. Croft, F.-F. Feng, D. Fisher, and X. Li. INQUERY and TREC-9. In *Proceedings of the Ninth Text Retrieval Conference (TREC-9)*, pages 551–562, 2000. NIST Special Publication 500-249.

[2] M. Bendersky, W. B. Croft, and Y. Diao. Quality-biased ranking of web documents. In *Proceedings of WSDM*, pages 95–104, 2011.

[3] J. G. Carbonell and J. Goldstein. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of SIGIR*, pages 335–336, 1998.

[4] G. V. Cormack, M. D. Smucker, and C. L. A. Clarke. Efficient and effective spam filtering and re-ranking for large web datasets. *Journal of Informaltiom Retrieval*, 14(5):441–465, 2011.

[5] T. Joachims. Training linear svms in linear time. In *Proceedings of KDD*, pages 217–226, 2006.

[6] D. Metzler and W. B. Croft. A Markov random field model for term dependencies. In *Proceedings of SIGIR*, pages 472–479, 2005.

[7] F. Raiber and O. Kurland. Ranking document clusters using markov random fields. In *Proceedings of SIGIR*, pages 333–342, 2013.