UNIVERSITY OF AMSTERDAM

UvA-DARE (Digital Academic Repository)

Topical diversity and relevance feedback

Meij, E.; He, J.; Weerkamp, W.; de Rijke, M.

Publication date
2010
Document Version
Final published version
Published in
The Eighteenth Text REtrieval Conference (TREC 2009) Proceedings

Link to publication

Citation for published version (APA):
Meij, E., He, J., Weerkamp, W., & de Rijke, M. (2010). Topical diversity and relevance feedback. In E. M. Voorhees, & L. P. Buckland (Eds.), *The Eighteenth Text REtrieval Conference (TREC 2009) Proceedings* (NIST Special Publication; No. 500-278). National Institute of Standards and Technology. http://trec.nist.gov/pubs/trec18/papers/uamsterdam-derijke.RF.pdf

General rights
It is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), other than for strictly personal, individual use, unless the work is under an open content license (like Creative Commons).

Disclaimer/Complaints regulations
If you believe that digital publication of certain material infringes any of your rights or (privacy) interests, please let the Library know, stating your reasons. In case of a legitimate complaint, the Library will make the material inaccessible and/or remove it from the website. Please Ask the Library: https://uba.uva.nl/en/contact, or a letter to: Library of the University of Amsterdam, Secretariat, P.O. Box 19185, 1000 GD Amsterdam, The Netherlands. You will be contacted as soon as possible.

UvA-DARE is a service provided by the library of the University of Amsterdam (https://dare.uva.nl)

Download date:08 Oct 2026

Topical Diversity and Relevance Feedback

Edgar Meij Jiyin He Wouter Weerkamp Maarten de Rijke

ISLA, University of Amsterdam
http://ilps.science.uva.nl/

Abstract: We describe the participation of the University of Amsterdam's Intelligent Systems Lab in the relevance feedback track at TREC 2009. Our main conclusion for the relevance feedback track is that a topical diversity approach provides good feedback documents. Further, we find that our relevance feedback algorithm seems to help most when there are sufficient relevant documents available.

1 Introduction

This year, the goal of the Relevance Feedback track is to evaluate how well a system can find good documents to serve as input for the relevance feedback algorithm, as well as the improvement gained by the feedback algorithm itself. There are two phases to the track. In the first phase, systems were to return two ranked lists (with a maximum of 5 documents) for each topic. In the second phase, all participating systems were given their own ranked list and a number of ranked lists from other groups from phase 1 and relevance assessments to perform relevance feedback.

For phase 1 we submitted two distinct runs. The first is based on an approach that attempts to maximize diversity, the second is based on a standard combination of term dependency and pseudo-relevance feedback. For phase 2 we adopt a four-step relevance feedback approach that generates ranked lists of documents for key terms in each judged document. We then combine each of these lists into two rankings (a relevant and a non-relevant one) which we then combine into a final ranking.

In the next section we first introduce our experimental setup. In Sections 3 and 4 we detail our approaches for phase 1 and 2, respectively, and we end with a concluding section.

2 Retrieval Framework

We employ a language modeling approach to IR and rank documents by their log-likelihood of being relevant given a query. Without presenting details here, we only provide our final formula for ranking documents, and refer the reader to (Balog et al., 2008) for the steps of deriving this equation:

$$\log P(D|Q) \propto \log P(D) + \sum_{t \in Q} P(t|\theta_Q) \cdot \log P(t|\theta_D). \quad (1)$$

Here, both documents and queries are represented as multinomial distributions over terms in the vocabulary, and are referred to as *document model* (θ_D) and *query model* (θ_Q), respectively. The third component of our ranking model is the *document prior* ($P(D)$), which is assumed to be uniform, unless stated otherwise. Note that by using uniform priors, Eq. 1 gives the same ranking as scoring documents by measuring the KL-divergence between the query model θ_Q and each document model θ_D, in which the divergence is negated for ranking purposes (Lafferty and Zhai, 2001).

2.1 Modeling

Unless indicated otherwise, we smooth each document model using a Dirichlet prior:

$$P(t|\theta_D) = \frac{n(t,D) + \mu P(t)}{\sum_t n(t,D) + \mu}, \quad (2)$$

where $n(t,D)$ indicates the count of term t in D and $P(t)$ indicates the probability of observing t in a large background model such as the collection:

$$P(t) = P(t|C) = \frac{\sum_D n(t,D)}{|C|}. \quad (3)$$

Here, μ is a hyperparameter that controls the influence of the background corpus which we set to the average document length.

2.2 ClueWeb

We do not use any form of stemming and remove a conservative list of 588 stopwords. We index the headings, titles, and contents as searchable fields and do not remove any HTML tags. For our submitted runs in phase 1 we used the Category B subset of ClueWeb, while for the runs in phase 2 we used Category A.

3 Phase 1

For the first phase we generated two runs based on different approaches. The first run was inspired by our approach

runID	score
ilps.1	0.8281
ilps.2	0.2885

Table 1: Results of our submitted runs in phase 1.

to the diversity task of this year's Web track (He et al., 2010), whereas the second run was a standard combination of pseudo-relevance feedback and query modeling.

3.1 Diversity

This run (ilps.1) tries to select documents that reflect different topical facets of a given query for relevance feedback. Intuitively, a query may have different topical facets, where some are relevant while others are non-relevant. From a clustering point of view, a set of documents that are representative for different topical facets would provide more information than documents that all focus on a single topical facet, since we can easily use a "prototypical" model to represent the single-topic set of documents.

For detecting different topical facets of the documents associated with each topic, we run hierarchical clustering on the top 50 documents from an initial retrieval run. For this kind of clustering one needs to pre-define a cut-off threshold that determines the number of clusters. However, in our scenario, we are not interested in getting a perfect clustering of the documents. Instead, we only want to detect the *significant* topical facets contained in the documents associated with a particular query. We measure the significance of a cluster with two measures: *stability* and *cluster quality*. A cluster is stable when it repeatedly occurs given different cut-off threshold and is of high quality when it results in a high Silhouette value (a measure for the quality of a cluster (Rousseeuw, 1987)). Additionally, in order to prevent outliers from dominating the top ranked clusters, we also take the cluster size into account. We rank the clusters by combining these scores, i.e., the stability, silhouette values, and cluster size, in a heuristic way by multiplying them. We leave other, more elaborate approaches for future work. Once we have obtained a ranked list of clusters, we select the top scoring documents from each cluster as our ranking.

3.2 Pseudo Relevance Feedback

For this run (ilps.2) we apply a standard combination of pseudo relevance feedback and structured query modeling. We first transform each query into a full-dependency query model (Metzler and Croft, 2005). We then perform a retrieval run and select the 10 top-ranked documents. From these documents we generate relevance models (RM-1 (Lavrenko and Croft, 2001)) and keep the 50 terms with the highest probability. We use the expanded query to retrieve our final ranked set of documents.

	Documents		IlpsRF	
runID	retrieved	relevant	MAP	P10
QUT.1	248	36	0.0688	0.1286
Sab.1	250	98	0.0581	0.0939
WatS.1	250	**110**	**0.0915**$^{\triangle}$	**0.2878**$^{\blacktriangle}$
fub.1	250	81	0.0792$^{\triangle}$	0.1694$^{\triangle}$
ilps.1	250	87	0.0705	0.2020$^{\blacktriangle}$
ilps.2	250	92	0.0680	0.1898$^{\triangle}$
twen.1	250	83	0.0776	0.1837$^{\blacktriangle}$
twen.2	250	71	0.0694	0.1816$^{\triangle}$
—	—	—	0.0639	0.0959

Table 2: Main results of our system for phase 2. The second and third column indicate the number of retrieved documents and the number of relevant documents for each of the baseline runs assigned to us. The rightmost columns contain the resulting performance of applying our feedback algorithm. Significance is tested using the Wilcoxon signed rank test against the run without any relevance feedback information (last row). We indicate significant increases (or drops) for $p < .01$ using $^{\blacktriangle}$ (and $^{\blacktriangledown}$) and for $p < .05$ using $^{\triangle}$ (and $^{\triangledown}$).

3.3 Results

Table 1 shows the aggregate score of our submitted runs for phase 1. We observe that ilps.1 is a better source of feedback documents than most other runs, whereas the opposite is true for ilps.2.

4 Phase 2

The main goal of phase 2 is to see how well each participants' relevance feedback algorithm performs, by running them on a set of 8 baseline runs (constructed in phase 1). Using each of the baseline runs and the relevance assessments, we need to identify new relevant documents. Participants were allowed to submit only one run and we suffice by reporting on our approach and its results. Comparing it to other approaches remains a topic for future work.

The leftmost columns of Table 2 lists the baseline runs that we were assigned, and the number of retrieved documents and the number of relevant documents in each run. As can be observed from this table, the information available from just the relevant documents is limited (the best run has 44% of its returned documents judged relevant). We believe that making our feedback approach dependent solely on these few documents is not a good idea and we feel we need to incorporate the non-relevant information as well to obtain the best relevance feedback results.

The general goal of a relevance feedback algorithm is to extract terms from relevant documents that distinguish them from other, non-relevant documents. One way of approaching this would be to use the non-relevant documents as a

language model against which to compare the relevant documents (Meij et al., 2008). From this comparison one could, for example, extract terms that distinguish between the relevant and non-relevant documents. Even though this is a valid approach, we feel that in the current situation this approach might not work optimally: first, the total number of judged documents is very limited (maximum of 5 documents per topic), which makes it hard to put confidence in comparing the two sets. Second, for a significant portion of the topics we have neither relevant nor non-relevant documents, and for these cases this approach would not work at all.

4.1 Approach

Building on the observations above, we arrive at the following wish list. First, a sensible approach to feedback should make use of each individual judged document as much as possible. Second, the approach should be able to handle cases in which no relevant or no non-relevant documents are known. Finally, as mentioned before, the approach should take non-relevance into account and not depend on relevant documents only. Based on these requirements we take a four-step relevance feedback approach:

1. Extract key terms from each individual document.
2. Use the extracted terms as queries.
3. Combine the result lists from step 2 in two rankings: a relevant and a non-relevant one.
4. Combine both rankings from step 3 into a final ranking.

Below we elaborate on these steps.

4.1.1 Extract key terms and run as queries

We compare each judged document to a background collection and identify key terms that distinguish this document. As background collection we take the full collection and we select only terms that occur at least four times in the document (to avoid selecting infrequent terms and typos). The weights of the resulting terms are normalized, leaving us with a weighted representation (or "query") for each document. We use this query to retrieve a set of new documents. We now have, for each judged document, a ranked list of documents which are highly similar. Examples of two queries, one relevant and one non-relevant, are displayed without their weights in Table 3. Additionally, we create a baseline ranking based on the original query terms.

4.1.2 Construct relevant and non-relevant rankings

We then combine the ranked lists from the previous step into two separate rankings: one for the relevant documents and one for the non-relevant documents. We do so by normalizing the retrieval scores for each topic and ranking using min-max normalization (Lee, 1995) and use CombMNZ (Fox and Shaw, 1994) to combine the relevant rankings into one, and the non-relevant rankings into one. We are now left with two new rankings, one being a ranking of relevant documents and the other a ranking of non-relevant documents.

Relevant	greyhounds, rescuing, doberman purebred, adoption, shih, collie, rescues
Non-relevant	adoption, transracial, photolisting

Table 3: Examples of the key terms from a relevant and non-relevant document for topic RF09-38, "dogs for adoption."

4.1.3 Construct final ranking

The final ranking is then constructed from the relevant and non-relevant rankings: we simply subtract the non-relevant score for each document from its relevant score. The idea behind this step is that a document that is ranked high for many relevant documents, but is hardly ever returned for non-relevant documents, receives a high final score. Documents that are mixed, i.e., showing up in both rankings, would get ranked below these documents, and documents that are ranked high in the non-relevant ranking and are nowhere to be found in relevant rankings, drop all the way to the bottom.

The approach described above fulfills our requirements in that it (i) takes full advantage of each individual document, (ii) can handle cases where no relevant or no non-relevant information is available, and (iii) takes non-relevance into account.

4.2 Results

Table 2 shows the result of applying our relevance feedback algorithm to our assigned input rankings from phase 1. From this table we observe that there seems to be a correlation between the number of relevant documents in the phase 1 ranking and the resulting, final performance. The last row of the table indicates the performance of our system without any relevance feedback information. We note that using relevance feedback information helps in all cases but one. Further, the improvement of applying our relevance feedback algorithm is significant for early precision in most cases. Finally, we observe that the absolute MAP values are quite low.

5 Conclusion

We have presented our approaches to the two phases of this year's TREC relevance feedback track. For phase 1 we found that an approach based on diversity outperforms a standard approach based on pseudo relevance feedback. As to phase 2, we have found that our proposed approach helps

most when there are sufficiently many relevant documents in the initial ranking.

6 Acknowledgments

This research was supported by the DAESO and DuOMAn project carried out within the STEVIN program which is funded by the Dutch and Flemish Governments under project number STE-09-12, and by the Netherlands Organisation for Scientific Research (NWO) under project numbers 640.001.501, 640.002.501, 612.066.512, 612.061.814, 612.061.815, 640.004.802, and by the Virtual Laboratory for e-Science project, which is supported by a BSIK grant from the Dutch Ministry of Education, Culture and Science and is part of the ICT innovation program of the Ministry of Economic Affairs.

7 References

Balog, K., Weerkamp, W., and de Rijke, M. (2008). A few examples go a long way: constructing query models from elaborate query formulations. In *SIGIR '08*. ACM.

Fox, E. and Shaw, J. (1994). Combination of multiple searches. In *The Second Text REtrieval Conference (TREC-2)*. National Institute of Standards and Technology (NIST).

He, J., Meij, E., Balog, K., Weerkamp, W., Hofmann, K., Tsagkias, M., and de Rijke, M. (2010). Heuristic ranking and diversification of web documents. In *Eighteenth Text REtrieval Conference (TREC 2009)*. National Institute of Standards and Technology (NIST).

Lafferty, J. and Zhai, C. (2001). Document language models, query models, and risk minimization for information retrieval. In *SIGIR '01*. ACM.

Lavrenko, V. and Croft, B. W. (2001). Relevance based language models. In *SIGIR '01*. ACM.

Lee, J. (1995). Combining multiple evidence from different properties of weighting schemes. In *SIGIR '95*. ACM.

Meij, E., Weerkamp, W., He, J., and de Rijke, M. (2008). Incorporating non-relevance information in the estimation of query models. In *Seventeenth Text REtrieval Conference (TREC 2008)*. National Institute of Standards and Technology (NIST).

Metzler, D. and Croft, W. B. (2005). A markov random field model for term dependencies. In *SIGIR '05*. ACM.

Rousseeuw, P. (1987). Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *J. Comput. Appl. Math.*, 20(1):53–65.