

# Evaluation of a Feedback Algorithm inspired by Quantum Detection for Dynamic Search Tasks

Emanuele Di Buccio and Massimo Melucci

Department of Information Engineering, University of Padua, Italy  
{dibuccio,melo}@dei.unipd.it

**Abstract.** In this paper we investigate the effectiveness of Relevance Feedback algorithms inspired by Quantum Detection in the context of the Dynamic Domain track. Documents and queries are represented as vectors; the query vector is projected into the subspace spanned by the eigenvector which maximizes the distance between the distribution of quantum probability of relevance and the distribution of quantum probability of non-relevance. When relevant documents are present in the feedback set, the algorithm performs Explicit RF exploiting evidence gathered from relevant passages; if all the documents in the top retrieved are judged as non-relevant, Pseudo RF is performed.

## 1 Introduction

The contribution reported in this paper can be considered as inscribed in the line of research on Quantum Information Retrieval (QIR) that investigates models and algorithms that rely on physics-inspired metaphors or mathematical formalism of Quantum Mechanics (QM) [6]. The idea of using the mathematical formalism of QM in IR was originally introduced in [13]. A review of diverse contributions in Information Retrieval (IR) and QM is reported in [6]; the proposed contributions range over diverse issues, e.g. modelling word ambiguity, semantic spaces, contextual dimensions, or user interaction.

The approach evaluated in this paper is a Relevance Feedback (RF) algorithm inspired by Quantum Detection (QD) [7]. In that paper we investigated the effectiveness in Explicit RF and Pseudo RF settings. The participation to the Dynamic Domain Track of TREC2016 allowed us to investigate the effectiveness of the proposed approach in a search task that simulates a dynamic and interactive search process. The participating systems interacted with a simulated user called *jig*. The *jig* provides real-time feedback on a short list of documents returned by the system. The objective of the task was to adapt the retrieval algorithm based on the feedback to return a new list of search results and to get another iteration of feedback. The process repeated until the system decided to stop the search; indeed, the search task was expected to be finished as soon as possible: the system should be able to provide the right amount of information to the user and then stop.

## 2 Relevance Feedback inspired By Quantum Detection

The general RF algorithm inspired by the principles of quantum detection is depicted in Figure 2; a detailed description is reported in [7]. The initial query represented as a vector  $y$  is input to the search engine of the IR system. The engine outputs a ranked list of  $m$  documents represented by the vectors  $x$ . The engine makes use of these document vectors for generating a feature matrix to be used to estimate the *state vectors*  $\phi_0$  and  $\phi_1$  according to the available relevance assessments. From the state vectors the density matrices  $\rho_0$  and  $\rho_1$  can be computed:  $\rho_0 = \phi_0\phi_0^T$  and  $\rho_1 = \phi_1\phi_1^T$ . A density matrix is a generalization of a classical probability distribution; in particular, the density matrix corresponding to a classical probability distribution is always diagonal and has unit trace because the sum of the diagonal elements is 1.

Starting from the density matrices  $\rho_0$  and  $\rho_1$ , the eigenvectors  $\eta_0$  and  $\eta_1$  are extracted by Singular Value Decomposition (SVD) of

$$\pi\rho_1 - (1 - \pi)\rho_0 \quad (1)$$

where  $\pi$  is the priori probability of  $\rho_1$  set to  $\pi = \frac{1}{2}$  in [7] and in this paper;  $\eta_1$  is the eigenvector corresponding to the positive eigenvalue.

The query vector  $y$  is then projected onto  $\eta_1$  and then the documents vectors are re-ranked by  $x^T\eta_1\eta_1^T y$ .

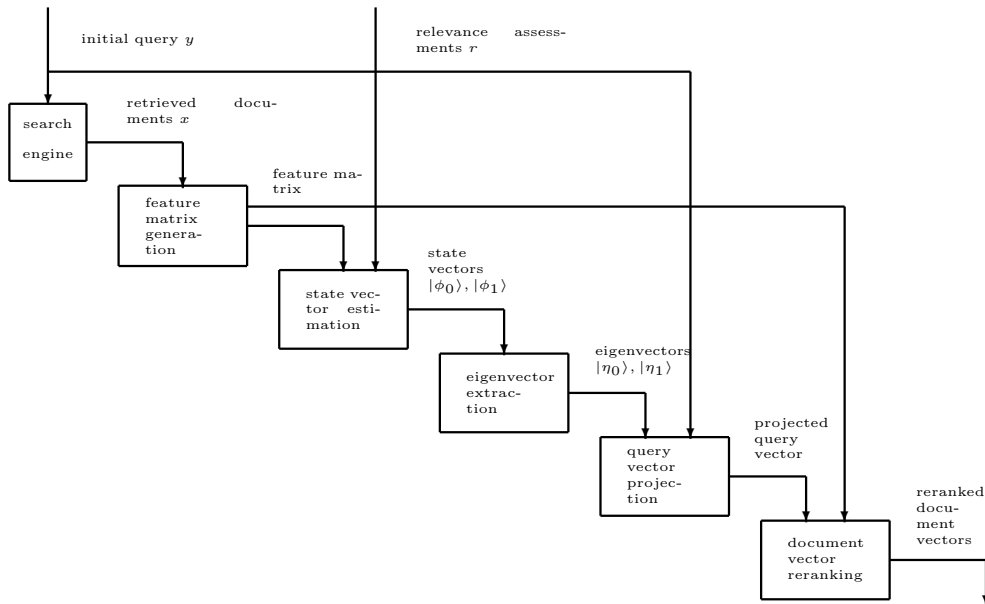


Fig. 1. General feedback algorithm inspired by Quantum Detection.

### 3 Methodology

The first prediction should be performed without any feedback. Our first prediction was based on BM25 [8] where the `topic name` was adopted to extract a description of the user information need. We retrieved  $m$  documents for each topic. The top 5 documents among the top  $m$  were provided as input to the `jig`.

Then for each iteration (interaction with the `jig`) Explicit RF or Pseudo RF were performed depending on whether relevant documents were present in the feedback set:

- **E\_QB\_RF**: if there was at least a relevant document in the feedback set
  1. The relevant passages provided by the `jig` were extracted.
  2. The distinct terms in the passages were extracted and then a term selection algorithm was used to select the top  $h$  terms; the final set of terms used to represent the query and the documents was the union of the top  $h$  extracted terms and the terms extracted from the topic name.
  3. Each document was represented as a binary vector  $x$  where the  $j$ -th element of the vector corresponded to the  $j$ -th descriptor and was 1 if the descriptor occurred in the document, false otherwise. A feature matrix was prepared with those vectors; both documents judged as relevant and non relevant are used to prepared the matrix.
  4. The feature matrix and the relevant assessments provided by the `jig` were adopted to extract the eigenvector  $\eta_1$  on which the query vector  $y$  was projected.  $\eta_1$  was computed by the Quantum Detection and Binary document representation (QB) algorithm [7].
  5. The query vector  $y$  was modified using the Rocchio’s algorithm [11]:

$$y^* = y + y^+ - y^- \quad (2)$$

where

$$y^+ = \frac{1}{|\mathcal{R}|} \sum_{d \in \mathcal{R}} x_d \quad y^- = \frac{1}{|\bar{\mathcal{R}}|} \sum_{d \in \bar{\mathcal{R}}} x_d \quad (3)$$

$\mathcal{R}$  and  $\bar{\mathcal{R}}$  are respectively the set of relevant documents and the set of non-relevant documents among those assessed by the `jig`. Basically, the new query vector is obtained as a linear combination of the original vector, the relevant document vectors and the non-relevant document vectors.

6. Documents in the residual collection were re-ranked according to the score  $x^T \eta_1 \eta_1^T y^*$ ; the residual collection was constituted by all the top  $m$  documents retrieved by BM25, not including all the documents “judged” by the `jig` till to that iteration.
- **P\_QB\_PRF**: if there were no relevant documents in the feedback set
    1. The top 100 documents from the residual collection were considered.

2. The distinct terms in the top 100 documents were extracted and then a term selection algorithm was used to select the top  $h$  terms; the final set of terms used to represent the query and the documents was the union of the top  $h$  extracted terms and the terms extracted from the topic name.
3. Each document was represented as a binary vector  $x$  where the  $j$ -th element of the vector corresponded to the  $j$ -th descriptor and was 1 if the descriptor occurred in the document, false otherwise. A feature matrix was prepared with those vectors.
4. The feature matrix was adopted to extract the eigenvector  $\eta_1$  on which the query vector  $y$  was projected. Also in this case, the QB algorithm was used to compute  $\eta_1$ .
5. The query vector  $y$  was modified as follows:

$$y^* = y + y^+ \quad (4)$$

where  $y^+$  is obtained as in the E\_QB\_RF, but assuming that the top 100 documents were all relevant —  $\mathcal{R}$  was therefore the set of top 100 retrieved documents in the residual collection at the current iteration.

6. Documents in the residual collection were re-ranked according to the score  $x^T \eta_1 \eta_1^T y^*$ ; the residual collection was constituted by all the top  $m$  documents retrieved by BM25, not including all the documents provided to the `jig` till to that iteration.

One of the IR system requirements was a criterion to stop when the “right” amount of information were provided to the user. We exploited two alternative criteria:

- perform a fixed number of iterations of feedback,  $I$ ;
- stop after 2 consecutive iterations of feedback with no relevant documents in the top 5 returned results (on the basis of the judgements returned by the `jig`) or after  $I$  iterations of feedback.

The current methodology ignores information on the subtopics provided by the `jig`. This information could be useful for the selection of the expansion terms — the query expansion step was not included in the original proposal [7]. Indeed, since the retrieved results should satisfy the diverse subtopics, the current approach (merge all the terms from the diverse passages in a single candidate list of terms for query expansion) could not be an effective strategy.

## 4 Experiment

### 4.1 Runs

The two submitted runs consist in two instances of the methodology described in Section 3; the main difference between the two instances is in the stopping criterion:

UPD\_IA\_BiQBFi: *automatic* run where BM25 was used for the initial prediction and 4 iterations of feedback based on QB.

UPD\_IA\_BiQBDiJ: *automatic* run where BM25 was used for the initial prediction and *maximum* 4 iterations of feedback based on QB; in particular, after two Pseudo RF-based re-ranking no additional iterations were performed.

Both the runs rely on following instantiation of the methodology:

- re-ranking of the residual top 1000 documents among those retrieved by BM25
- terms for the new query representation were the union among the topic terms (extracted from the topic name field) and the top 35 terms extracted by WPQ weight [9] among those occurring in the feedback passages (Explicit RF case) or in the feedback documents (Pseudo RF case).

The WPQ weight for a term  $t$  is defined as follows:

$$\text{WPQ}_t = g_t \cdot (p_t - q_t) \quad (5)$$

where  $p_t$  is the probability that a given relevant document is assigned the term  $t$  and  $q_t$  is the “equivalent” non relevant probability;  $g_t$  is the RSJ [10] weight

$$g_t = \log \frac{p_t (1 - q_t)}{(1 - p_t) q_t}$$

The estimation of  $p_t$  and  $q_t$  is performed as follows:

$$p_t = \frac{r_t + 0.5}{R + 1} \quad q_t = \frac{n_t - r_t + 0.5}{N - R + 1} \quad (6)$$

where  $n_t$  and  $r_t$  are respectively the number of documents in the collection and the number of relevant documents where  $t$  occurs;  $R = |\mathcal{R}|$  and  $N$  are respectively the number of relevant documents in the feedback set and the number of documents in the collection.

## 4.2 Test Collection and Measures

The test collection adopted is constituted of two datasets, one for each of the domains considered in this track edition: the Polar Domain and the Ebola Domain. Each dataset is formatted using the Common Crawl Architecture schema from the DARPA MEMEX project, and stored as sequences of CBOR objects.

The *Ebola dataset* refers to the outbreak in Africa in 2014-2015 and is constituted of web pages from the affected countries, PDFs and tweets. Only documents contained in the `Ebola-web-01-2015` and the `Ebola-web-03-2015` subsets were adopted for the experimental evaluation; the two subsets refer respectively to the set of web pages crawled during January 2015 and during March 2015. The total number of documents of the Ebola dataset used in our experiments is 194,481.

The *Polar dataset* is intended to support the investigation of climate change in Polar regions. The dataset is a collection of web crawls that results in 1,741,530 records; detailed information on the dataset can be found in [5]. As for the Ebola dataset, only a subset was used in this track edition; the total number of records used is 244,536.

**Table 1.** Statistics on the adopted datasets.

Dataset	Subset used	Number of documents used
Ebola	ebola-web	194,481
Polar	all	244,536

The evaluation aimed both at measuring the speed of completion for an entire search task and the capability to satisfy the diverse subtopics in a topic. The list of measures adopted in the track are listed below:

- Cube Test (CT) and Average Cube Test (ACT) [14]
- $\alpha$ -NDCG [2] and Average  $\alpha$ -NDCG
- nERR-AI [12] and Average nERR-AI
- nSDCG [4]
- Precision up to the current iteration

## 4.3 Experimental System and Settings

The implementation of the methodology described in Section 3 relies mainly on Apache Lucene version 4.7.2 and a Octave script to implement the Feedback algorithm inspired to Quantum (Signal) Detection described in Section 2.

As for parsing and indexing, the CBOR records were parsed through the module made available in [3]; then the functionalities provided by Galago [1] to perform HTML parsing of Web pages were used to extract the document content.

The parsed content was indexed by Apache Lucene using a Standard Analyzer. No stopwords were used during indexing and no stemming was adopted. All the records of the two datasets were indexed in a single index — the total number of documents indexed was 439,017.

As for retrieval with no feedback, the BM25 [8] weighting scheme was implemented in order to rely directly on the Apache Lucene index API. The default values were adopted for the BM25 parameters:  $b = 0.75$  and  $k_1 = 1.2$ . No stemming was adopted. The stopwords made available in the Lemur Toolkit were adopted for retrieval.

#### 4.4 Results

Results are reported in Tables 2–9 for the first 5 iterations (4 iterations of feedback) since this is the maximum number of iterations used by both the proposed stopping criteria. Results in terms of ACT and Precision are also reported respectively in Figure 2 and Figure 3. The results at the first prediction (iteration 0, no feedback) in terms of the primary measures are quite far from the average and the median value — see Table 2 and Table 3. The first iteration of feedback seems to be the most effective; in terms of CT the result obtained is the median value.

When considering run UPD\_IA\_BiQBFiJ, for 10/53 topics we observed that an iteration of Pseudo RF was followed by an iteration of Explicit RF and therefore that P\_QB\_PRf was able to provide relevant documents at high rank positions — the list of those topics is: DD16-14, DD16-18, DD16-25, DD16-29, DD16-31, DD16-33, DD16-36, DD16-40, DD16-46, DD16-52.

## 5 Final Remarks

This paper reported on the evaluation of an algorithm for relevance feedback inspired by Quantum Detection at the Dynamic Domain track of TREC 2016.

Even if the obtained results in terms of the primary measures – Average Cube Test and Cube Test – are not close to the average and median value computed over all the runs, the proposed approach seems to be more promising when precision is considered as the evaluation measure. Moreover, for a number of topics with no relevant documents in the feedback set, the proposed approach was able to retrieve relevant documents among the top 5 through the pseudo relevance feedback variant, P\_QB\_PRf.

One limitation of the current methodology is that it treats the Dynamic Domain task as a feedback task. At each iteration, the information on the previous feedback interactions is not used to extract of the eigenvectors  $\eta_0$  and  $\eta_1$  or to select terms from the candidate set; moreover, no domain knowledge is actually used. Another limitation of the current methodology is that it ignores information on the subtopics provided by the `jig`. This information could be useful for the selection of the expansion terms; since the retrieved results should satisfy the diverse subtopics, the current approach (merge all the terms from

the diverse passages in a single candidate list of terms for query expansion) could not be an effective strategy. Finally, the current methodology relies on a binary document representation; in [7] variants relying on Frequency-based or Normalised Frequency-based document representation were proposed. Future works will address these limitations of the proposed methodology and investigate the effectiveness of the other Quantum Detection algorithms based on different document representations.

**Table 2.** Results in terms of Average Cube Test (ACT)

Iteration	UPD_IA_BiQBDiJ	UPD_IA_BiQBFi	avg	median
0	0.1050	0.1050	0.1472	0.1516
1	0.1107	0.1107	0.1361	0.1352
2	0.1051	0.1051	0.1259	0.1242
3	0.0984	0.0984	0.1190	0.1116
4	0.0938	0.0925	0.1140	0.1092

**Table 3.** Results in terms of Cube Test (CT)

Iteration	UPD_IA_BiQBDiJ	UPD_IA_BiQBFi	avg	median
0	0.1698	0.1698	0.2049	0.2174
1	0.1281	0.1281	0.1388	0.1281
2	0.1020	0.1020	0.1131	0.1097
3	0.0862	0.0863	0.1015	0.0981
4	0.0774	0.0760	0.0946	0.0898

**Table 4.** Results in terms of  $\alpha$ -NDCG

Iteration	UPD_IA_BiQBDiJ	UPD_IA_BiQBFi	avg	median
0	0.2275	0.2275	0.2999	0.2952
1	0.2736	0.2736	0.3339	0.3142
2	0.2922	0.2922	0.3431	0.3145
3	0.3009	0.3016	0.3482	0.3142
4	0.3066	0.3073	0.3510	0.3142



**Table 5.** Results in terms of nERR-IA

Iteration	UPD_IA_BiQBDiJ	UPD_IA_BiQBFi	avg	median
0	0.2011	0.2011	0.2821	0.2691
1	0.2238	0.2238	0.2985	0.2777
2	0.2307	0.2307	0.3018	0.2780
3	0.2333	0.2335	0.3033	0.2779
4	0.2348	0.2350	0.3040	0.2778

**Table 6.** Results in terms of Average  $\alpha$ -NDCG

Iteration	UPD_IA_BiQBDiJ	UPD_IA_BiQBFi	avg	median
0	0.0176	0.0176	0.0265	0.0274
1	0.0105	0.0105	0.0161	0.0162
2	0.0075	0.0075	0.0127	0.0121
3	0.0064	0.0058	0.0110	0.0098
4	0.0058	0.0047	0.0099	0.0089

**Table 7.** Results in terms of Average nERR-AI

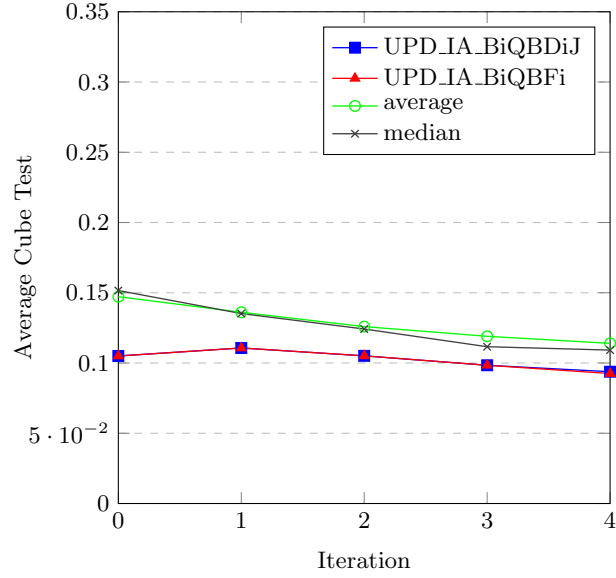
Iteration	UPD_IA_BiQBDiJ	UPD_IA_BiQBFi	avg	median
0	0.0151	0.0151	0.0242	0.0231
1	0.0086	0.0086	0.0138	0.0139
2	0.0059	0.0059	0.0107	0.0098
3	0.0051	0.0046	0.0092	0.0083
4	0.0046	0.0037	0.0083	0.0073

**Table 8.** Results in terms of nsDCG

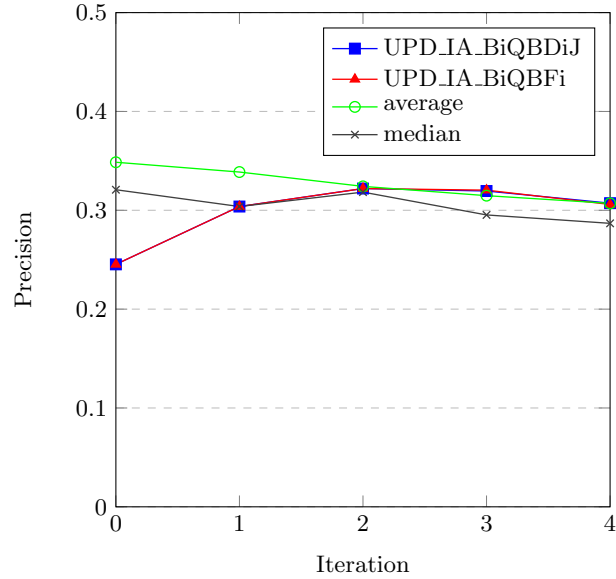
Iteration	UPD_IA_BiQBDiJ	UPD_IA_BiQBFi	avg	median
0	0.1412	0.1412	0.1879	0.1901
1	0.0909	0.0909	0.1098	0.0940
2	0.0718	0.0718	0.0863	0.0759
3	0.0602	0.0603	0.0776	0.0688
4	0.0543	0.0529	0.0729	0.0556

**Table 9.** Results in terms of Precision

Iteration	UPD_IA_BiQBDiJ	UPD_IA_BiQBFi	avg	median
0	0.2453	0.2453	0.3486	0.3208
1	0.3038	0.3038	0.3387	0.3038
2	0.3220	0.3220	0.3242	0.3184
3	0.3195	0.3204	0.3148	0.2953
4	0.3072	0.3059	0.3070	0.2868



**Fig. 2.** Average Cube Test at each iteration for the two runs UPD\_IA\_BiQBdIJ and UPD\_IA\_BiQBfI; average and median value computed over all the runs are also reported. Iteration 0 denotes the first prediction with no feedback.



**Fig. 3.** Precision at each iteration for the two runs UPD\_IA\_BiQBdIJ and UPD\_IA\_BiQBfI; average and median value computed over all the runs are also reported. Iteration 0 denotes the first prediction with no feedback.

## References

1. Lemur Components: Galago. <http://lemurproject.org/galago.php>.
2. C. L. Clarke, M. Kolla, G. V. Cormack, O. Vechtomova, A. Ashkan, S. Büttcher, and I. MacKinnon. Novelty and diversity in information retrieval evaluation. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval - SIGIR '08*, page 659, New York, New York, USA, 2008. ACM Press.
3. FasterXML, LLC. Fasterxml/jackson-dataformat-cbor. <https://github.com/FasterXML/jackson-dataformat-cbor>.
4. K. Järvelin, S. L. Price, L. M. L. Delcambre, and M. L. Nielsen. Discounted Cumulated Gain Based Evaluation of Multiple-query IR Sessions. In *Proceedings of the IR Research, 30th European Conference on Advances in Information Retrieval, ECIR'08*, pages 4–15, Berlin, Heidelberg, 2008. Springer-Verlag.
5. C. Mattmann. TREC Dynamic Domain Polar Dataset. <https://github.com/christmattmann/trec-dd-polar/>.
6. M. Melucci. *Introduction to Information Retrieval and Quantum Mechanics*, volume 35 of *The Information Retrieval Series*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2015.
7. M. Melucci. Relevance Feedback Algorithms Inspired By Quantum Detection. *IEEE Transactions on Knowledge and Data Engineering*, 28(4):1022–1034, apr 2016.
8. S. Robertson and H. Zaragoza. The Probabilistic Relevance Framework: BM25 and Beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389, 2009.
9. S. E. Robertson. On term selection for query expansion. *Journal of Documentation*, 46(4):359–364, 1990.
10. S. E. Robertson and K. S. Jones. Relevance weighting of search terms. *Journal of the American Society for Information Science*, 27(3):129–146, 1976.
11. J. J. Rocchio. *Relevance Feedback in Information Retrieval*, pages 313–323. 1971.
12. T. Sakai and R. Song. Evaluating diversified search results using per-intent graded relevance. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '11*, pages 1043–1052, New York, NY, USA, 2011. ACM.
13. C. J. van Rijsbergen. *The Geometry of Information Retrieval*. Cambridge University Press, 2004.
14. H. Yang, J. Frank, and I. Soboroff. TREC 2015 Dynamic Domain Track. 2015.