University of Padua at TREC 2009: Relevance Feedback Track

Emanuele Di Buccio and Massimo Melucci

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

Abstract. In the Relevance Feedback (RF) task the user is directly involved in the search process: given an initial set of results, he specifies if they are relevant or not to the achievement of his information goal. In the TREC 2009 RF track the first five documents retrieved by the baseline systems were judged by the assessors and then used as evidence for the RF algorithms to be tested. The specific algorithm we tested is mainly based on a geometric framework which allows the latent semantic associations of terms in the feedback documents to be modeled as a vector subspace; the documents of the collection represented as vectors of TF·IDF weights were re-ranked according to their distance from the subspace. The adopted geometric framework was used in past works as a basis for Implicit Relevance Feedback (IRF) and Pseudo Relevance Feedback (PRF) algorithms; the participation to the RF track allows us to make some preliminary investigations on the effectiveness of the adopted framework when it is exploited to support explicit RF on much larger test collections, thus complementing the work carried out for the other RF strategies.

1 Introduction

In TREC 2009 the Information Management System (IMS) Research Group of the University of Padua (UNIPD) participated to the RF Track. The track was structured in two phases, namely Phase 1 and Phase 2. The purpose of Phase 1 was to evaluate the systems capability of retrieving good documents to be judged, that is, documents which would be good input for RF algorithms to be tested. The aim of Phase 2 was to evaluate the improvement provided by the RF algorithms when different sets of judged documents were used as input. We submitted results both to Phase 1 and Phase 2.

The specific RF algorithm we evaluated is based on the geometric framework proposed in [1], which allows different sources for feedback to be modeled as vector subspaces and their models to be exploited to predict relevance. In the previous works the framework was applied to two different sources.

The first source was the behavior of the user described in terms of interaction features gathered by monitoring the interaction between the user and the Information Retrieval (IR) system [2]. The user behavior modeled as a vector subspace was used to re-rank the documents: the most frequent keywords were

extracted from the top n re-ranked documents and keywords were adopted for expanding the textual description of the topic, which was then considered as a new, expanded query. That approach falls into the class of IRF algorithms, since interaction features can be gathered without an direct involvement of the user and their combination was used as implicit indicator of the user intents or interests.

The second source for feedback used was the "latent semantics" [3] of the terms appearing in the top n retrieved documents [1]; the top k weighted keywords in these documents were adopted to extract the most "meaningful" term groups, as in Latent Semantic Analysis (LSA). In practice the adopted approach provided a vector subspace representation of the term groups; the top m retrieved documents were re-ranked according to the distance between their vector representation in terms of the top weighted keywords and the computed subspace.

Therefore the effectiveness of the adopted geometric framework was tested respectively with regard to IRF [2] and PRF [1]. The purpose of the work carried out in the RF Track was to test the effectiveness of that framework with regard to Explicit Relevance Feedback (RF) by using a test collection of two orders of magnitude larger than those used in the previous experiments. In particular, the source for feedback used was the content of the top two documents judged as relevant by the assessors among the top five documents retrieved. The approach proposed in [1] was applied to the content of these documents in order to re-rank the top 2500 results retrieved by the baseline. The baseline adopted in Phase 1 exploited the BM25 weighting scheme [4] to provide an initial ranked list of results. Then the top ten retrieved documents were re-ranked according to presence of the topic keywords in their URL's.

The remainder of this paper is structured as follows. Section 2 briefly explains the methodology for RF adopted in this work and the role of the adopted geometric framework in such methodology. Section 3 focuses on the experiments carried out during the participation to the RF Track, moreover describing the setting adopted for indexing and retrieval both in Phase 1 and Phase 2. The results obtained by the experiments described in Section 3 are reported and discussed in Section 4. Finally, Section 5 reports some concluding remarks.

2 Methodology

The specific methodology for relevance feedback we tested in the RF Track of TREC 2009 is that proposed in [5]. The methodology is constituted by four steps: (i) selection of the source of feedback, (ii) selection and collection of the features, (iii) source modeling and (iv) relevance prediction.

As regards the first step, the source for feedback selected is the latent semantic structure in the content of the documents used as evidence. Differently from [1], where the content of top n retrieved documents were used as source of evidence, in this work the source adopted is the content of a subset $\mathcal{F}$ of the documents judged as relevant among the top n retrieved by a first retrieval run — in the RF Track the initial run is the Phase 1 run.

The main assumption underlying this work is that some terms appearing in the documents in $\mathcal{F}$ can be used to predict what the terms used by the searchers really imply. In other words the terms appearing in the considered subset of the feedback documents are the features selected to model the considered source for feedback. The specific information adopted is the co-occurrence of the terms appearing near each others: windows of text centered around the terms can be used to capture "local co-occurrence" information. Suppose that the terms "music", "restaurant", "rock" and "jazz" are selected as features. If in the documents in $\mathcal{F}$ the term "jazz" tends to occur more frequently near "music" and/or "restaurant", maybe the searcher is more interested in restaurants where jazz music is played than in those with live rock music.

This local co-occurrence information can be extracted and prepared in a matrix as follows. Let $\mathcal{T}$ be the set of k features, namely terms, selected to describe the source and let $S \in \mathbb{R}^{k \times k}$ be a matrix whose elements are initially set to zero, namely $s_{ij} = 0$ for $1 \leq i, j \leq k$. For each term $t_i \in \mathcal{T}$ a window of text centered around each occurrence of t_i is considered; if a term $t_j \neq t_i \in \mathcal{T}$ appears in the window of text, statistical information about t_j, e.g. its total frequency in the collection, or a weight derived from such information, e.g. the TF·IDF, is added both to s_{ij} and s_{ji}.

The main question is how to obtain a usable representation of the source for feedback adopted in order to assist the prediction of the documents relevance. A possible solution is that proposed in [1], where the mathematical construct of the vector subspace is adopted to model sources. The main issue is how to obtain a vector subspace representation starting from the information collected by the observation of the selected features. A possible solution, which is the approach actually adopted in this work, is to apply Singular Value Decomposition (SVD) to S and select the first principal eigenvector.

This vector spans a subspace which can be used to re-rank the documents in the collection, that is to implement the relevance prediction step of the methodology. This goal can be achieved by the adoption of a trace-based function — the idea of using trace-based functions in IR was originally proposed in [6] and subsequently developed in [1]. Let us denote with $\mathbf{b}$ the first principal eigenvector among those provided by SVD and denote with $L(\{\mathbf{b}\})$ the subspace spanned by $\mathbf{b}$. We are interested in measuring the degree to which the latent semantic structure modeled as subspace is present in the documents of the collection, and rank the documents according to this measure. The mentioned function measures the distance between the vector representation of the document $\mathbf{y}$ and the subspace $L(\{\mathbf{b}\})$, that is the projection $\mathbf{y}_{\{\mathbf{b}\}}$ of $\mathbf{y}$ onto $L(\{\mathbf{b}\})$. More formally, the function adopted is the following:

$$\mathrm{m}_{\{\mathbf{b}\}}(\mathbf{y}) = \mathbf{y}^T \cdot \mathrm{P}_{\{\mathbf{b}\}} \cdot \mathbf{y}, \tag{1}$$

where $\mathrm{P}_{\{\mathbf{b}\}} = \mathbf{b} \cdot \mathbf{b}^T$ is the projector onto the subspace $L(\{\mathbf{b}\})$.

The measure provided by Equation 1 is a probability measure, as shown in [1], that is $\mathrm{m}_{\{\mathbf{b}\}}(\mathbf{y}) = \Pr[L(\{\mathbf{b}\})|L(\{\mathbf{y}\})]$, where $L(\{\mathbf{y}\})$ denotes the subspace spanned by $\mathbf{y}$.

3 Experiments

The IR system adopted in the experiments exploits the functionalities provided by Apache Lucene [7] for indexing and retrieval[1]. The specific choices made in regard to parsing, indexing and retrieval are described in the remainder of this section. Both the experiments for Phase 1 and Phase 2 were carried out on the TREC 2009 "Category B" dataset, constituted by 50,220,423 English web pages.

The experiments were carried out on a cluster of twenty-eight 3 GHz Intel Quad Core® E5450, which is available in our department.

3.1 Parsing and Indexing

Each web-page of the TREC 2009 "Category B" dataset was parsed, particularly the following information was extracted from each record in Web ARChive (WARC) format: the TREC-ID, the URI and the content. Each of them was stored in a distinct `Field` of a Lucene `Document`[2]. All the content of the document was processed during indexing except for the text contained inside the `<script></script>` and the `<style></style>` tags. Moreover an additional field was stored, which contained the keywords extracted from the URL of the document. In particular during the extraction of the terms from the full content of the documents the presence of each term was checked in the URL; the obtained keywords were then indexed in a separate field, which was used in Phase 1 to re-rank the top ten retrieved documents.

Stop words were removed during indexing[3]. No stemming was adopted. During indexing not only statistical information about the occurrence of the terms in the documents, namely their frequency, was stored, but also information about the positions where terms occurred and offset information[4]. The information about the position of the terms was used to implement the methodology described in Section 2 and exploited for Phase 2 as described in Section 3.3.

The wall-clock time to index the 1492 records of the TREC 2009 "Category B" dataset was 45 hours, 46 minutes and 45 seconds, while the CPU time was 38 hours, 3 minutes and 39 seconds (36:29:08 user time and 01:34:30 system time).

3.2 Retrieval: Phase 1

The purpose of Phase 1 of the RF Track was to retrieve good documents to be judged, actually the documents used as input for feedback in Phase 2.

[1] The specific version adopted in the experiments was Apache Lucene 2.4.1

[2] "A `Document` represents a collection of fields [...] Each field corresponds to a piece of data that is either queried against or retrieved from the index during search" [8]

[3] The stop words list is that available at the url `http://ir.dcs.gla.ac.uk/resources/linguistic_utils/stop_words`

[4] In Lucene information about the unique terms in a field, their counts, their positions and their offsets can be stored at indexing time and then accessed by using `TermVector`s. The specific `TermVector` option chosen for the Lucene `Field` used for the "content" was `TermVector.WITH_POSITIONS_OFFSETS`

Each of the fifty topics was automatically parsed, thus extracting its constituting terms; no stemming was adopted on the obtained terms. For each term q_i in a topic we constructed a Lucene `TermQuery` for the content field, that is a query to retrieve all the documents where the term q_i appears in their content field. The `TermQuery`'s constructed for the terms q_i's in a topic were combined in a Lucene `BooleanQuery`: each `TermQuery` was considered as a optional clause, that is `TermQuery`'s were combined by logical `OR`[5].

The weighting scheme adopted was the BM25, particularly exploiting the implementation for Lucene made available in [9] which is based on the description of the BM25 presented in [10] and briefly described in the following. Let V_D be the set of terms appearing in document D; the weight w_i assigned to the term $t_i \in V_D$ is

$$w_i = \frac{tf'_i}{k_1 + tf'_i} \log \frac{N - n_i + 0.5}{n_i + 0.5}$$

where N is the total number of document in the collection, n_i is the number of documents in the collection where the term t_i appears, and k_1 is a parameter which was heuristically set to $k_1 = 2$ in the experiments. The quantity tf'_i is defined as $tf'_i = tf_i/B$, where tf_i is the term frequency of t_i, and

$$B = (1 - b) + b\frac{dl}{avdl}$$

where $dl = \sum_{t_i \in V_D} tf_i$ is the document length, and $avdl$ is the average document length in the collection. The value of b adopted in the experiments was $b = 0.75$.

The top ten retrieved documents by BM25 were re-ranked according to the number of the topic keywords among those extracted from the URL field. If two documents had the same BM25 score and the same number of topic keywords in the URL field, the documents were ranked according to the lexicographical order of their identifiers. The top five re-ranked documents were provided as results for Phase 1.

3.3 Retrieval: Phase 2

Phase 2 aimed at investigating the effectiveness of the RF algorithms when different Phase 1 runs were used as source for feedback. In other words the objective was to test the effectiveness of the algorithms with regard to different baseline systems and the documents they provided. Seven sets of judged documents were assigned to UNIPD, particularly those provided by the Phase 1 runs `CMU.1`, `hit2.1`, `ilps.2`, `PRIS.1`, `QUT.1`, `UMas.1` and `UPD.1`.

The specific algorithm we tested in Phase 2 was that described in Section 2, particularly using a subset of the relevant documents among the top five as evidence to extract the latent semantics of the terms. The methodology is summarized in the following steps:

[5] The specific boolean operator adopted for the Lucene `BooleanQuery` was `BooleanClause.Occur.SHOULD`

1. Selection of the top h relevant documents among the top five retrieved for the specific topic and the particular Phase 1 run considered. If the number of documents judged as relevant among the top five retrieved is greater than one, then the top two relevant documents are selected, that is $h = 2$. If only one document is judged as relevant, that document is selected, that is $h = 1$. If there are no relevant documents among the top five, the baseline ranked list is returned as result for Phase 2, specifically the top $m = 2500$ documents.
2. Selection of the set $\mathcal{T}$ of the top $k = 5$ weighted terms in the selected relevant documents; the weight of the keywords is computed by TF·IDF.
3. Computation of the co-occurrence matrix S by windows of text — only the full text of the selected relevant documents is used. In particular a window of text of size 11 is centered around each occurrence of a keyword $t_i \in \mathcal{T}$. If a keyword $t_j \in \mathcal{T}$ appears in the window of text centered around t_i, the TF·IDF weight of t_j is added to the elements s_{ij} and s_{ji} of S. The window of text never overlaps two distinct documents.
4. Decomposition of the co-occurrence matrix S by SVD and adoption of vector subspace $L(R_F)$ spanned by the first eigenvector $\mathbf{b}$ as model of the selected source[6].
5. Re-ranking of the top $m = 2500$ results retrieved by the baseline according to the distance between the vector representation of the documents and the computed subspace; the specific function adopted is Eq. 1, that is $\mathbf{y}^T \cdot \mathrm{P}_{\{\mathbf{b}\}} \cdot \mathbf{y}$, where $\mathbf{y}$ is the document vector normalized so that $||\mathbf{y}|| = 1$, $\mathrm{P}_{\{\mathbf{b}\}} = \mathbf{b} \cdot \mathbf{b}^T$, and $\mathbf{b}$ is the eigenvector computed in the previous step.

The results submitted for Phase 2 were the re-ranked list of documents obtained at step 5 or, as mentioned in step 1, the results provided by the baseline if there were no documents judged as relevant among the top five retrieved. The reason for the latter choice is due to the difference between the "subspace of irrelevance" and the subspace spanned by non relevant documents. Indeed, as stated in [1], if orthogonality is chosen to model mutual exclusion and $L(R_F)$ denotes the subspace of relevance, $L(R_F)^{\perp}$ may denote irrelevance. While the subspace of irrelevance is orthogonal to $L(R_F)$, $L(\overline{R}_F)$ is in general oblique — $L(\overline{R}_F)$ denotes the subspace spanned by non relevant documents. In other words, ranking according to $1 - \Pr[L(R_F)^{\perp}|L(\{\mathbf{y}\})]$ is in general different than ranking by $1 - \Pr[L(\overline{R}_F)|L(\{\mathbf{y}\})]$. If all the documents are judged by searchers as non relevant, $L(\overline{R}_F)$ can be computed but not $L(R_F)^{\perp}$. For this reason, if none of the top five retrieved documents were judged as relevant, the baseline results were returned.

4 Results

In Phase 1 the baseline we adopted was able to retrieve at least one relevant document among the top five results for 37 of the 50 topics. Table 1 reports

[6] In the experiments the JAMA package [11] was used to implement all the functionalities for constructing and manipulating matrices.

the *statAP* [12] computed for the results returned by the baseline (B) and the results returned by the adopted RF algorithm (RF) for the 49 topics[7]. Moreover Table 1 reports the percentage difference between the baseline results and the results provided by the RF algorithm. For eight of the thirty-seven topics the RF algorithm was effective in terms of *statAP* — in Table 1 the results referring to these topics are bolded — , but in general the RF negatively affected the ranked list provided by the baseline.

Topic	B (stAP)	RF (stAP)	Δ_{RF-B} (%)	Topic	B (stAP)	RF (stAP)	Δ_{RF-B} (%)
1	0.14444	0.00031	-99.78468	28	0.23796	0.07298	-69.33195
2	0.62849	0.12205	-80.58025	29	0.00220	—	0.00000
3	**0.07932**	**0.10391**	**30.99605**	30	0.18557	0.01058	-94.30020
4	0.01892	0.00100	-94.70874	**31**	**0.15710**	**0.24758**	**57.59663**
5	0.16549	0.00813	-95.08602	32	0.25684	0.10356	-59.67980
6	0.05699	—	0.00000	33	0.43541	0.32751	-24.78026
7	0.03044	0.00294	-90.35278	34	0.00781	0.00156	-79.98719
8	0.00247	—	0.00000	35	0.20071	0.19446	-3.11295
9	0.07686	0.01832	-76.16712	**36**	**0.05640**	**0.12389**	**119.68437**
10	**0.28083**	**0.42035**	**49.68539**	37	0.12500	0.00228	-98.17360
11	0.10908	0.02643	-75.76685	38	0.18799	—	0.00000
12	0.27450	0.20720	-24.51484	39	0.25447	0.05185	-79.62345
13	0.00560	—	0.00000	40	0.15270	0.11565	-24.26390
14	0.02117	—	0.00000	41	0.32043	0.18921	-40.95185
15	**0.17731**	**0.22655**	**27.77088**	42	0.00000	—	0.00000
16	0.17146	0.08028	-53.17796	43	0.14216	—	0.00000
17	**0.06113**	**0.07385**	**20.80784**	44	0.02379	0.00169	-92.90068
18	0.09633	0.05081	-47.25253	45	0.25197	0.03685	-85.37581
19	0.00000	—	0.00000	46	0.69705	0.33444	-52.02037
21	0.37863	0.14138	-62.65867	47	0.29992	0.20607	-31.29334
22	0.43105	0.07411	-82.80633	48	0.20898	0.02941	-85.92811
23	0.03668	—	0.00000	**49**	**0.07552**	**0.10736**	**42.16290**
24	0.13116	—	0.00000	**50**	**0.11841**	**0.36787**	**210.67824**
25	0.03912	0.01756	-55.12067	—	—	—	—
26	0.08003	0.03321	-58.51035	All	0.16549	0.10148	-38.68247
27	0.21324	—	0.00000				

Table 1: Comparison between the *statAP*'s computed for the results returned by the baseline (B) and the results returned by the RF algorithm (RF). Δ_{RF-B} denotes the *statAP* percentage difference between the baseline and the RF algorithm. For the topics with no relevant documents among the top five retrieved, only the baseline *statAP* is reported since $\Delta_{RF-B} = 0$.

[7] Topic 20 was dropped since none of the Phase 1 runs returned relevant documents for such topic.

Run ID	*statAP*
`CMU.1`	0,08267
`hit2.1`	0,09385
`ilps.2`	0,09844
`PRIS.1`	0,17784
`QUT.1`	0,13487
`Umas.1`	0,09700
`UPD.1`	0,10148

Table 2: *statAP* computed over all the topics with regard to the seven Phase 1 sets assigned to UNIPD.

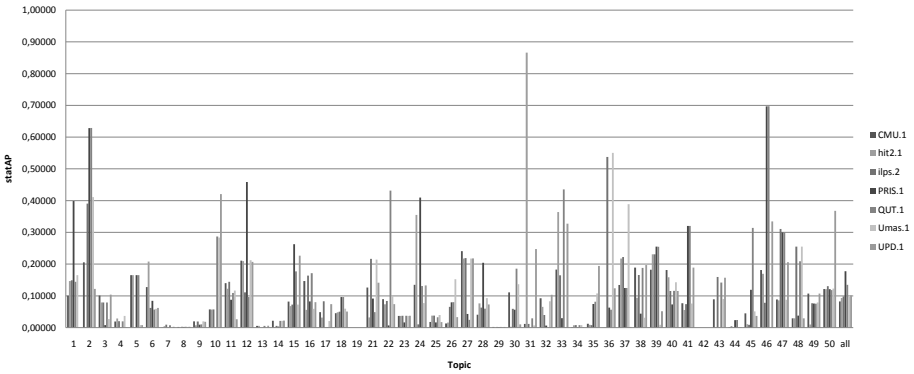

Fig. 1: *statAP* reported for the different runs based on the assigned Phase 1 sets with regard to all the topics.

Table 2 reports the *statAP* computed over all the topics with regard to the Phase 1 runs assigned to UNIPD for Phase 2. The results show that `PRIS.1` and `QUT.1` were able to provide more effective evidence to perform RF than the UNIPD Phase 1 run (`UPD.1`). But when the *statAP*'s are considered with regard to each topic — see Figure 1 — there is no Phase 1 set which provides good evidence for feedback to the tested RF algorithm for all the topics.

In regard to effectiveness of `UPD.1` as source for feedback, Table 3 reports the number of runs the UNIPD Phase 1 set was worse (<) and better (>) than the other Phase 1 runs with regard to all the groups to which `UPD.1` was assigned — the values reported are computed over all the topics.

5 Concluding Remarks

The results reported show how the RF algorithm tested is less effective than the baseline in terms of *statAP*.

One of the reasons for these results may be the little evidence used for feedback: only the content of the top two documents judged as relevant among the

Measure	<	>
ϵ-map	13	7
mapA	4	3
P10A	4	3
stAP	12	8

Table 3: Impact of Phase 1 UNIPD set on the other groups which used such run as evidence for feedback.

top five retrieved were used. This suggests an investigation of the impact of the adopted number of relevant documents on the effectiveness of the RF algorithm.

The adoption of the `AND` operator instead of `OR` to construct the queries from the topic keywords may improve the obtained results in terms of precision. Moreover one issue to be investigated is the selection of the features to build the model of the source. Indeed not necessarily the features with the highest TF·IDF weights in the feedback documents are those most useful for feedback — in several cases we observed that some of the features selected were not related to the topic. The approach adopted to model the content of the feedback documents as a vector subspace seems to help in the event of the wrong selected features. Indeed the weights assigned by the first eigenvector to those features are lower — often near to zero — than that assigned to features related to the topic. Moreover the query was not expanded, that is the new query did not necessarily includes the topic terms, but only the selected features: this choice might have hurt the effectiveness of the algorithm. As a consequence the way a better selection of the features affects the tested RF algorithm will be matter of investigation.

References

1. M. Melucci. A basis for information retrieval in context. *ACM Transactions on Information Systems*, 26(3):1–41, 2008.
2. M. Melucci and R.W. White. Utilizing a geometry of context for enhanced implicit feedback. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, pages 273–282, 2007.
3. S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41:391–407, 1990.
4. S. E. Robertson and S. Walker. Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval. In *Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 232–241, New York, NY, USA, 1994.
5. E. Di Buccio and M. Melucci. Exploiting individual users and user groups interaction features: methodology and infrastructure design. In *Proceedings of the Second Workshop on Very Large Digital Libraries*, Corfu, Greece, 2009.
6. C.J. van Rijsbergen. *The Geometry of Information Retrieval*. Cambridge University Press, New York, NY, USA, 2004.

7. Apache Lucene - Overview. http://lucene.apache.org/java. Last update: October 10, 2009.
8. O. Gospodnetić and E. Hatcher. *Lucene in Action*. Manning Publications Co., first edition, 2004.
9. J. P. Iglesias. Integrating BM25 & BM25F into Lucene. http://nlp.uned.es/~jperezi/Lucene-BM25/. Last update: December 26, 2008.
10. S. Robertson and H. Zaragoza. The Probabilistic Ranking Method: BM25 and beyond. SIGIR 2007 Tutorial. http://www.yr-bcn.es/dokuwiki/doku.php?id=prm, July 2007.
11. JAMA : A Java Matrix Package. http://math.nist.gov/javanumerics/jama/. Last update: July 13, 2005.
12. J. Allan, B. Carterette, J. A. Aslam, V. Pavlu, B. Dachev, and E. Kanoulas. Overview of the TREC 2007 Million Query Track. In *Proceedings of TREC*, 2007.