Learning in Information Retrieval: a Probabilistic Differential Approach

Benjamin Piwowarski
Laboratoire d'Informatique de Paris 6 (LIP6)
Paris, France
Benjamin.Piwowarski@lip6.fr

Abstract

Since user's relevance judgments are a source of evidence for information retrieval, learning from this feedback is an appealing idea. Many different learning techniques have successfully been used for relevance feedback. In most models, learning is either performed off line or is based on simple heuristics. The approach we propose is based on a classical probabilistic model in which learning from feedback is simple and incremental. In this paper, we extend this model, presenting a new similarity function that takes into account feedback on the entire database while computing the score between a query and a single document. As a result, when a user judges a document it modifies the whole retrieval process.

1 Introduction

A perfect document retrieval system should be able to retrieve, from a database, documents that are *relevant* to a user's query. Unfortunately, many non-relevant answers are usually also retrieved. Even if one can improve the quality of document and query indexation, this automatic process is never perfect. User relevance judgments provide evidence about the user's real information need and have been used extensively in information retrieval for learning [16]. The problem of learning can be summarized as follows: "knowing a user's query need and his/her opinion about the relevance of some documents to his/her need, what can be done to improve the system retrieval performance ?".

There exists two main learning modes for IR. The first considers retrieval as an interactive process. That is, the user gives his/her opinion on the documents found by the system, this information is then used to modify the query representation and a new search is performed. Once the retrieval session is finished, all feedback information is usually lost[1]. The other one aims at developing permanent learning models that use user's feedback to modify some internal retrieval parameters and/or document representation. In this case, feedback effect is permanent as it modifies the retrieval process – either immediately or after an update of the system. In this article, we will focus on permanent learning.

Many permanent learning techniques require batch sessions or are computationally demanding. For most of them, learning is an external process that modifies off line parameters and/or document representation. Off line learning is often computationally demanding and often, adding a new document or new feedback necessitate a whole re-learning of the system which is not desirable. On the contrary, the model we are based on is such that feedback information and document representation cannot be dissociated. In this model, learning is also simple and incremental. We further extend this model to improve learning speed and the initialization of the document's representation easier.

The paper is organized as follows. We make a review of learning models in information retrieval in section 2 and categorize them according to which part of the retrieval process they modify. In section 3, we introduce the "document centered" probabilistic approach upon which our model is based, and show the advantages and limitations of this model. We then describe our differential probabilistic model. Finally, we present the results obtained on two different test collections in section 4.

[1] Note however that Keim et al [13] propose to keep this feedback for future retrieval sessions.

2 Previous research

In information retrieval, systems that learn from user feedback attempt to modify either the query representation or the similarity function parameters or the document representation. We review methods of these different categories mentioned below and we also present ideas taken from the neural network community.

2.1 Modifying the similarity function parameters

An information retrieval system contains different parameters that can be adjusted to increase the system's performance for a specific database and a specific query style (e.g. natural language queries, keywords...). For doing so, many learning techniques aim to optimize a function depending on these parameters.

Because a retrieval system is more or less a system that outputs a ranked list of documents, Bartell et al. [3, 1, 2] suggest to optimize the Guttman's point alienation for improving retrieval. This criterion is based on non-binary relevance judgments that create a partial order on documents. Hence, a document is said to be superior to another one with respect to a query need when the user prefers this document to the other. The criterion they define reaches its minimum when the order created by the similarity function is the same that the order defined by the users. They show that this criterion is highly correlated to the average precision. The parameters of the retrieval system are then optimized so as to minimize this criterion. Such parameters can be weights of different similarity measures which are then linearly combined [1], or parameters of a similarity measure [2].

Another method used to optimize similarity parameters is the technique of regression analysis. As in probabilistic retrieval, we want to predict the value of a variable, the relevance, given some quantitative variables on the document and query relationship (like scores of some Vector Space Model similarity measures) which may contribute to the prediction. Two different approaches exist: linear regression methods (Fuhr et al. [11]) and logistic regression methods (Cooper et al. [9]). The latter approach, as it gives results in the interval [0..1], is more appropriate for the probabilistic framework.

We can also cite Yu and Raghavan [24] who propose to build semantic relationships with feedback. The basic idea is that each query feature that is not present in a relevant document is somehow semantically (e.g. synonyms) related to a document feature. They use this information to define a new similarity measure.

2.2 Modifying the query representation

The "query pool" framework, which combines interactive and permanent learning, was proposed by Keim et al. [13]. In the probabilistic model they are based on, feedback is lost when the retrieval session ends. In this approach, previous interactions (the three-tuples [query,document,relevance]) are used to modify a new query presented to the system. The principle of this approach is that the more a past query is *similar* to the new query, the more it contains relevant information for the new one. This model allows interactive and permanent learning but is computationally expensive for retrieving documents and adds the problem of evaluating similarity between queries.

All these approaches do not alter document representation, which sets clear limits on their effectiveness. In many situations, documents should be enriched – e.g. to match the query terms, and the relative weights of document terms should be carefully tuned. The next two sections present approaches which modify the document representation.

2.3 Neural networks

Neural networks are widely used in machine learning tasks and several authors have developed explicit networks for retrieval, e.g. the PIRCS system (Kwok [14, 15]) and the work of Belew [4]. These models share the same basic characteristics. When a query is presented to the system, individual nodes that represent the query are activated (with varying intensity). Then this initial activation spreads through the network before activating the document nodes. The most highly stimulated documents are then retrieved. Feedback learning in such models can be done by classical backpropagation. The major drawback of such networks are the difficulty to control the learning process and the difficulty to add or remove a document.

2.4 Modifying documents representation

The Brauen's Document Vector Modification [7] is one of the simplest method used to learn from feedback. When a document is judged relevant for a query, the learning algorithm modifies the document representation for each feature present either in the query or in the document. When a document feature is present in the query (in Vector Space Model, the set of features is the same for both queries and documents), its weight is increased. When a feature is not in the query, its weight is decreased (unless it was not present in the document representation, e.g. with weight 0). As a result, if the same query is matched against this document, the similarity measure will be increased. A major drawback of this method is the lack of control in the learning process.

This principle is extended by Bodoff [6]. To avoid an uncontrolled modification of the document vector, he defines a function $stress(\mathcal{D}, \mathcal{Q})$, where $\mathcal{D}$ is the set of documents and $\mathcal{Q}$ the set of past queries. Basically, this function decreases as each document representation gets closer to queries for which they are relevant and away from queries for which they are not. But *stress* also increases as the distance between the document's (or query's) initial representation and its current representation increases, thus avoiding an excessive modification of documents vectors. The goal of learning is to minimize the *stress* function. Hence, this model attempts to reach for each document an equilibrium between its initial representation and the representation of the queries for which it is relevant.

The last model we will quote is the probabilistic model (called model 1) from Maron and Kuhns [18, 17, 19]. It is also the basis of our present work. The principle is to learn from feedback a relation between each possible query feature or term and a specific document. A document is therefore represented by a table which indicates for each term:

- the number of queries containing this term for which the document was judged relevant.

- the number of queries containing this term for which the document was judged not relevant.

This model is well suited for permanent learning since it integrates feedback directly in the document representation. As a result, the model enables incremental feedback to be implemented efficiently (it is just an update of statistics). The major drawback, as stated in [23, 10], is that the feedback modifies only the representation of a single document. Hence, the amount of feedback needed to produce a significant effect (or a good document representation) is important. Our model, as it will be shown in the next section, uses the same information for each document but considers it as *relative* to the entire database and not *limited* to one document as in model 1.

3 The differential probabilistic method

3.1 Introduction

Our approach is based on the probabilistic document-centered method. As any other probabilistic approach, its foundation lies in the *Probability Ranking Principle*. This principle states that ranking the documents in the decreasing order of probability of relevance knowing the document and the query, $P(R|q,d)$, will minimize the expected number of documents a user has to consult before finding all the relevant documents to his query (if he follows that order). This probability can't be evaluated directly, instead one uses the following criterion which provides an equivalent ranking:

$$\log \frac{P(R|d)}{P(\overline{R}|d)} + \log \frac{P(q|R,d)}{P(q|\overline{R},d)} \tag{1}$$

(1) is at the basis of the document-centered approach. Here, we have to estimate the probability to have a query q knowing that we have the document d and a relevance or non-relevance relationship. We will show that (1) can be rewritten so as to explicitely take into account the influence of all the documents present in the database when computing the score between a specific document and a specific query. This theoretical modification brings some interesting properties:

- Document initialization is easier.

- Learning is faster.

In the following, we define some key concepts of our model and show how we derive from formula 1 a new similarity measure that respects the Probability Ranking Principle. We also present simplifying assumptions we have to make – in order to compute a score between a query and a document – and explain how learning is performed. For ease of reading, the description is mainly informal, all mathematical details are deferred to appendixes B and C.

3.2 Notations

The complete list of notations is deferred to appendix A. We introduce below the main concepts which will be used in the derivation of our model. The definition of the probabilistic space and events is classical and we will follow Schaüble [22]. As for all probabilistic models, we have to solve the following basic problem: how to evaluate the probability that we have a *relevance relationship* knowing a document and a query. To evaluate this probability, we have to make a first assumption about either the query event[2] q or the document event d. In this model, as discussed before, we have chosen a document-centered approach. We consider that every query q is represented by a conjunction of events Q_c or $\overline{Q_c}$ where Q_c means *term c is in the query*. We denote $\underline{q}$ this representation of q. Thus, two queries sharing the same representation are considered to be the same event. On the contrary, a document event d is unique and can't be expressed as a conjunction of simpler events.

As a result, statistics collected to approximate probabilities will be centered on the document. We thus define:

- $R(c,d)$ as the number of queries that contains term c for which document d is relevant. We denote $R(c,D')$, where $D' \subset \mathcal{D}$, the sum of $R(c,d)$ for all $d \in D'$.
- We define similarly $\overline{R}(c,d)$ and $\overline{R}(c,D')$ in the case of non-relevance.
- $R(d)$ (resp. $\overline{R}(d)$) as the number of queries for which d is relevant (resp. non-relevant).

Note that statistics for $\overline{Q_c}$ events can be derived from $R(d)$ and $R(c,.)$. Adding feedback in this model is simple. When a document is judged relevant for a query q, we increment the value of $R(c,d)$ for every term c of the query q. Similarly, when this document is non-relevant, we increment the value of $\overline{R}(c,d)$.

3.3 Probabilistic assumptions for IR

For the computational tractability of our model, we make simplifying assumptions. The first two assumptions relate to the queries and are mandatory while the two following ones relate to the documents and can be easily replaced by other *a priori* assumptions.

The following assumption is usual in probabilistic IR and can be thought as the minimal assumption we have to make in order to have a simple relationship between $\underline{q}$ and the conjunction of Q_c and $\overline{Q_c}$ [22, 10]:

Hypothesis 1 (Linked dependence) *For a given $D' \subset \mathcal{D}$ and a given query q, we have the following relationship:*

$$\frac{P(\underline{q}|R,D')}{P(\underline{q}|\overline{R},d)} = \prod_{c\in\underline{q}} \frac{P(Q_c|D',R)}{P(Q_c|D',\overline{R})} \prod_{c\notin\underline{q}} \frac{P(\overline{Q_c}|D',R)}{P(\overline{Q_c}|D',\overline{R})}$$

This assumption is slightly weaker than the binary independance and also avoids logical inconsistencies [8]. Computing for every document a partial score for each possible query feature would take too much time for a fast retrieval. We use the following assumption to ignore every term c which is not present in the query:

Hypothesis 2 (Term absence) *Terms that are not in the query representation do not have any effect on the relevance or non-relevance of a document. Formally, we have*

$$\frac{P(\underline{q}|R,D')}{P(\underline{q}|\overline{R},d)} = \prod_{c\in\underline{q}} \frac{P(Q_c|D',R)}{P(Q_c|D',\overline{R})}$$

[2] We use a simplified notation for most probabilistic events. For example, the event q is confused with the query $q \in \mathcal{Q}$.

We will further assume that the document distribution is uniform and that all the documents have an equal *a priori* relevance probability.

Hypothesis 3 *For all* $d, d' \in \mathcal{D}$, $P(d) = P(d')$ *and* $P(R|d) = P(R|d')$.

3.4 Principle

The proposed model does not look for a direct estimate of the relevance of a specific document d. On the contrary, we aim to evaluate this probability for what is called here an *anti-document*. This anti-document can be seen as a representation of the database when the document d is removed. We are interested in the following question: if we remove this document from the database, will the probability to find a relevant document increase or decrease ? If it increases, then this document is probably not relevant. In the other case, the decrease value will tell us about the relevance of this document.

We introduce below a score value $S(q,d)$, for measuring the similarity of q and d and ranking documents. In S probabilities are expressed as a function of the event anti-document, denoted $\neg d$. From a theoretical point of view, the first thing to do is to show the equivalence between the order implied by formula 1 and a formula S. In appendix B, we show that the order implied by $P(R|q,d)$ is not changed when ordering with $-\frac{P(R|\underline{q},\neg d)}{P(\overline{R}|\underline{q},\neg d)}$. We then derive this result in order to obtain an equivalent formula where each term can be easily approximated. This formula gives us the form of the probability that has to be estimated. That is, the probability to have a query containing term c in the set of queries relevant to a document in $\mathcal{D} \setminus d$. In appendix C, we give an estimate of such a probability. Finally, we obtain the following formula which approximates a similarity function that orders documents according to the Probability Ranking Principle:

$$S(q,d) = -|\underline{q}| \log \frac{1 - \frac{\overline{R}(d)+2}{\overline{R}(\mathcal{D})+2|\mathcal{D}|}}{1 - \frac{R(d)+2}{R(\mathcal{D})+2|\mathcal{D}|}} - \sum_{c \in \underline{q} \cap \underline{d}} \log \frac{1 - \frac{R(c,d)}{R(c,\mathcal{D})+|\mathcal{D}|-1}}{1 - \frac{\overline{R}(c,d)}{\overline{R}(c,\mathcal{D})+|\mathcal{D}|-1}}$$

This model could work without any document initialization, using feedback progressively. Nevertheless, the information contained in documents can help to bootstrap the system. We have chosen empirically the following initialization, for each document d and term c – where $\lfloor x \rfloor$ is the integer part of x, $N_c(d)$ is the number of term c occurrence in d, and $\max_d$ is the $N_c(d)$ maximum for all term $c \in d$:

$$\begin{aligned}
R(c,d) &= \begin{cases} \lfloor 1 + (N_R - 1)\varphi(c,d) \rfloor + 1 & \text{if } N_c(d) > 0 \\ 1 & \text{if } N_c(d) = 0 \end{cases} \\
\overline{R}(c,d) &= \begin{cases} \lfloor 1 + (N_{\overline{R}} - 1)(1 - \varphi(c,d)) \rfloor + 1 & \text{if } N_c(d) > 0 \\ 1 & \text{if } N_c(d) = 0 \end{cases}
\end{aligned}$$

where $\varphi(c,d) = \frac{\log(N_c(d)+1)}{\log(\max_d + 1)}$. And for each document $d \in \mathcal{D}$:

$$R(d) = 2 + N_R \tag{2}$$

$$\overline{R}(d) = 2 + N_{\overline{R}} \tag{3}$$

A constant of 1 appears in $R(c,d)$ and $\overline{R}(c,d)$ and a constant of two appears in $R(d)$ and $\overline{R}(d)$. They are used to model the uncertainty when term c is not present in d. This initialization can be seen as the first "feedback" for the document d, where N_R (resp. $N_{\overline{R}}$) is the number of queries for which the document is relevant (resp. non-relevant) – the only terms appearing in these queries being the terms contained in the document. This initialization also determines the inertia of our system: the higher N_R is, the more time this system take before reacting to feedback.

This first initialization, although very simple, gives good result with our model (see section 4). Indeed, as our model computes a score between a query and a *document in its database context*, we do not have to take into account the distribution of each feature over all the documents. As a result, document initialization can be computed only from the indexation of this document. Consequently, adding or removing a document in the database is easy.

4 Evaluation

For evaluation, we used two different collections[3]. The first one is the Cranfield collection that contains 1398 documents, 225 queries and 1837 relevance feedback judgments. The second is the CISI collection that contains 1460 documents, 112 queries and 3114 relevance feedback judgments. For each collection, queries are randomly split into two different sets[4], the "training set" and the "evaluation set".

The evaluated algorithms are:

- *TF-IDF* : the classical SMART algorithm [20] (Vector Space Model), e.g. the TF-IDF weighting scheme with cosine similarity measure.
- $DIFF_n$ for $n = 0, 1, 10$: the differential method. Three different experimental conditions are evaluated. The first one ($n = 0$) does not use any feedback from the training set. In the second one ($n = 1$), all the training queries are used to update the document representation. In the third one ($n = 10$), each training query "simulates" the effect of 10 queries with the same text and the same relevant documents.

The evaluation measures chosen are:

- Table 2 and figure 1 and 2: the precision-recall curves.
- Table 3: an equivalent of the "precision at n documents" table. Instead of the precision we chose to show the mean number of relevant documents in the *n* first documents given by the retrieval system, which is equivalent.

The results show the validity of our approach when one considers these following two observations:

- when no feedback is used, our model performance is close to the SMART results. This is important since it shows that our new similarity measure can compare with a reference model.
- Feedback improves the system overall performance. The feedback effect is even stronger when the query weight is increased (e.g. $DIFF_{10}$). This last result suggests that we can take advantage of non-binary relevance judgments like "highly relevant", "very relevant", etc. For each such judgment, we could present to the system the three-tuple (query,document,relevance) a number of times depending on this judgment.

The two collections are rather small and we used them here only to assess the feasibility of our approach. Both are build from short documents, the main difference is in the ratio of relevant document per query which is much higher in CISI. Initialization is crucial for the behavior of our method and could still be improved. The superior performance of $DIFF_{10}$ compared to $DIFF_1$ is certainly partly due to this raw initialization. On the other hand, the goal of this method is to discover automatically good document representations.

5 Conclusion

An extension of the document-centered probabilistic model has been proposed. Our approach uses a new similarity function that takes into account all the information available in the database to compute the score between a query and a single document. The advantages with regard to the former model are a real learning effect and an easier document initialization. Unlike other approaches, learning is very simple (it is just an update of statistics) and allows to build a complete information retrieval system that learns permanently. This approach uses the query indexation to replace progressively an empirical document initialization[5]. An interesting extension would be the use of a query specific linguistic knowledge to get more accurate features. To improve the system, one could also consider the use of a hierarchical document database where each node would contain query terms that have the same importance for all the documents it contains (or that sub-nodes contain). As a result, retrieval would be faster and more precise (as feedback can be shared by many documents).

[3] `http://www.dcs.gla.ac.uk/idom/ir_resources`

[4] See the table 1 for statistics about the different sets

[5] Note that this initialization can surely be improved

	Cranfield		CISI	
	training set	evaluation set	training set	evaluation set
Number of queries	158	67	67	44
Mean number of feedback by query	8.4	7.5	27.3	29.3
Mean number of feedback by document	0.95	0.36	1.25	0.88
Mean number of feedback for a relevant document in the evaluation set	1		1.5	
Percentage of documents relevant to a query in the evaluation set which have feedback in the training set	61 %		78 %	

Table 1: Training and evaluation sets characteristics. Most lines are self explanatory. The last line is the percentage among documents which are relevant to a query from the evaluation set of those which are also relevant to a query from the training set. Such documents are modified during training ($DIFF_1$ and $DIFF_{10}$).

Recall level	Precision							
	Cranfield				CISI			
	TF-IDF	DIFF$_0$	DIFF$_1$	DIFF$_{10}$	TF-IDF	DIFF$_0$	DIFF$_1$	DIFF$_{10}$
0.1	74.4 %	73.2 %	73.3 %	**76.9 %**	**39.1** %	32.8 %	37.1 %	37.4 %
0.4	41.0 %	41.1 %	42.6 %	**51.1 %**	19.6 %	16.8 %	18.6 %	**20.4 %**
0.7	21.4 %	20.0 %	20.9 %	**29.4 %**	8.8 %	8.1 %	9.5 %	**14.4 %**
1.0	10.1 %	08.4 %	08.9 %	**14.4 %**	3.7 %	4.2 %	**4.5 %**	4.2 %

Table 2: Precision-recall table for Cranfield and CISI.

Mean number of relevant documents at...	Cranfield				CISI		
	TF-IDF	DIFF$_0$	DIFF$_1$	DIFF$_{10}$	TF-IDF	DIFF$_0$	DIFF$_1$
1 document	0.66	0.69	0.72	**0.73**	0.32	**0.42**	**0.42**
2 documents	1.12	1.13	1.15	**1.21**	0.58	0.71	**0.77**
3 documents	1.52	1.49	1.54	**1.63**	1.00	1.00	**1.03**
5 documents	2.01	1.97	2.03	**2.23**	1.71	1.65	**1.78**
10 document	2.73	2.72	2.82	**3.16**	2.90	2.84	**2.94**
R documents	35.7 %	34.1 %	36.0 %	**40.5 %**	**19.7 %**	18.8 %	19.2 %

Table 3: Precision at *n* documents for Cranfield and CISI.

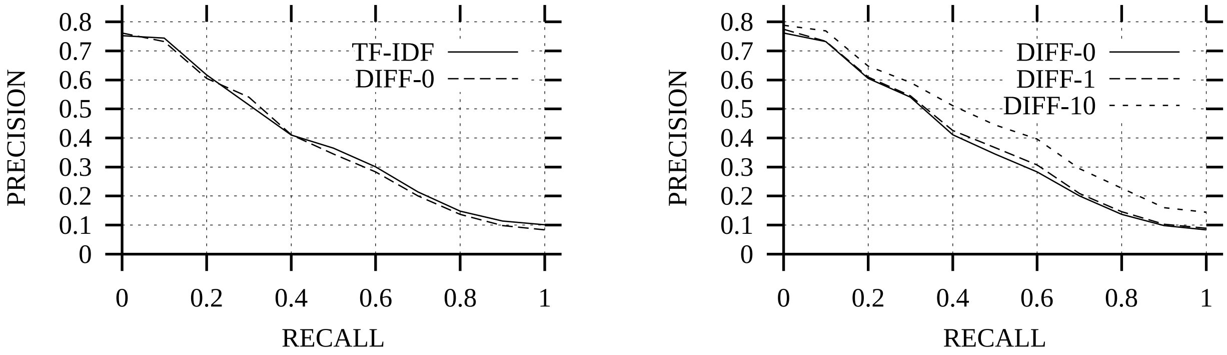

Figure 1: Cranfield 1400 collection. Left: comparison with TF-IDF-cosine. Right: feedback effect

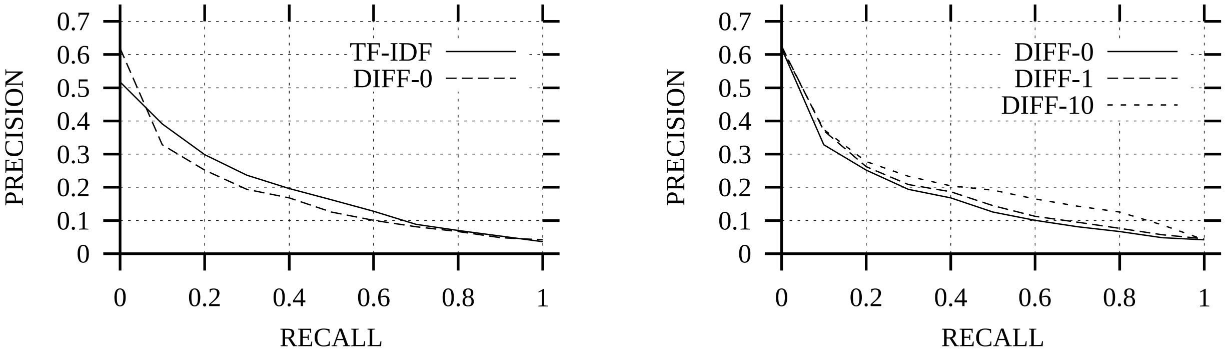

Figure 2: CISI collection. Left: comparison with TF-IDF-cosine. Right: feedback effect

Acknowledgments

I would like to thank the LexiQuest society where this work has been done and where I found powerful linguistic resources to index documents and queries. A particular thanks to Roger Le Borgne (LexiQuest) and Patrick Gallinari (LIP6) who supervised this work, and also to Hugo Zaragoza and Massih-Reza Amini who helped me to prepare this article.

References

[1] BARTELL, B., COTTRELL, G. W., AND BELEW, R. Learning to retrieve information. In *Current trends in connectionism: Proceedings of the Swedish Conference on Connectionism* (LEA: Hillsdale, 1995), L. Niklasson and M. Boden, Eds.

[2] BARTELL, B., COTTRELL, G. W., AND BELEW, R. Optimizing similarity using multi-query relevance feedback. *Journal of the American Society for Information Science 49*, 8 (1998), 742–761.

[3] BARTELL, B. T., COTTRELL, G. W., AND BELEW, R. K. Optimizing parameters in a ranked retrieval system using mutli-query relevance feedback. In *Proceedings Symposium on Document Analysis and Information Retrieval* (Las Vegas, Nevada, Apr. 1994), University of Nevada.

[4] BELEW, R. K. Adaptative information retrieval: Using a connectionnist representation to retrieve and learn about documents. In Belkin and van Rijsbergen [5], pp. 11–20.

[5] BELKIN, N. J., AND VAN RIJSBERGEN, C. J., Eds. *Proceedings of the ACM SIGIR 12th International Conference on Research and Development in Information Retrieval* (Cambridge, Massachusetts, USA, June 1992), ACM Press.

[6] BODOFF, D. A re-unification of two competing models for document retrieval. *Journal of the American Society for Information Science 50*, 1 (Jan. 1999), 49–64.

[7] BRAUEN, T. Document vector modification. In Salton [20], ch. 24, pp. 456–484.

[8] COOPER, W. S. Some inconsistencies and misidentified modelling assumptions in probabilistic information retrieval. *ACM Transactions On Information Systems 13* (1995), 100–111.

[9] COOPER, W. S., CHEN, A., AND GEY, F. C. Full text retrieval based on probalistic equations with coefficients fitted by logistic regression. In Harman [12], pp. 57–66.

[10] CRESTANI, F., LALMAS, M., VAN RIJSBERGEN, C. J., AND CAMPBELL, I. ”is this document relevant ?... probably” : a survey of probabilistic models in information retrieval. *ACM Computing surveys 30*, 4 (Dec. 1998), 528–552.

[11] FUHR, N., PFEIFER, U., BREMKAMP, C., POLLMANN, M., AND BUCKLEY, C. Probabilistic learning approaches for indexing and retrieval with the trec-2 collection. In Harman [12], pp. 67–74.

[12] HARMAN, D. K., Ed. *NIST Special Publication 500-215: The Second Text REtrieval Conference (TREC-2)* (Gaithersburg, MD, Aug.-Sept. 1993), Department of Commerce, National Institute of Standards and Technology.

[13] KEIM, M., LEWIS, D. D., AND MADIGAN, D. Bayesian information retrieval: Preliminary evaluation. In *Preliminary Papers of the Sixth International Workshop on Artificial Intelligence and Statistics* (Ft. Lauderdale, Florida, Jan. 1997), D. Madigan and P. Smyth, Eds., pp. 303–310.

[14] KWOK, K. A neural network for probabilistic information retrieval. In Belkin and van Rijsbergen [5], pp. 11–20.

[15] KWOK, K. L., GRUNFELD, L., AND LEWIS, D. D. Trec-3 ad-hoc, routing retrieval and thresholding experiments using PIRCS. In *NIST Special Publication 500-226: Overview of the Third Text REtrieval Conference (TREC-3)* (Gaithersburg, MD, Nov. 1994), D. K. Harman, Ed., U. S. Dept. of Commerce, National Institute of Standards and Technology, pp. 247–255.

[16] LEWIS, D. D. Learning in information retrieval. In *Machine Learning: Proceedings of the Eighth International Workshop (ML 91)* (San Mateo, CA, 1991), L. A. Birnbaum and G. C. Collins, Eds., Morgan Kaufmann, pp. 235–239.

[17] MARON, M. Probabilistic approaches to the document retrieval problem. In Salton and Schneider [21], pp. 98–107.

[18] MARON, M., AND KUHNS, J. On relevance, probabilistic indexing and information retrieval. *JACM 7* (July 1960).

[19] ROBERTSON, S., MARON, M., AND COOPER, W. The unified probabilistic model for ir. In Salton and Schneider [21], pp. 109–117.

[20] SALTON, G., Ed. *The SMART Retrieval System: Experiments in Automatic Document Processing*. Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1971.

[21] SALTON, G., AND SCHNEIDER, H.-J., Eds. *Research and development in Information Retrieval* (Berlin, May 1983), vol. 146 of *Lecture notes in computer science*, Springer-Verlag, Berlin.

[22] SCHÄUBLE, P. *Multimedia Information Retrieval*. Kluwer Academic Publishers, 1997.

[23] SPARK JONES, K., WALKER, S., AND ROBERTSON, S. A probabilistic model of information retrieval: development and status. Tech. Rep. 446, Computer Laboratory, University of Cambridge, Aug. 1998.

[24] YU, C. T., AND RAGHAVAN, V. V. Single-pass method for determining the semantic relationships between terms. *Journal of the American Society for Information Science* (Nov. 1977), 345–354.

Appendix

A Definitions

We denote $\Omega = \mathcal{Q} \times \mathcal{D}$ the sample space where $\mathcal{Q}$ is the set of queries and $\mathcal{D}$ the set of documents. $P = 2^{\Omega} \rightarrow [0, 1]$ assigns every event E a probability $P(E)$. We will consider the following events in this probability space:

- The event *having the document* d_h, noted $E(d_h)$ or d_h, $E(d_h) = \{(q,d) \in \underline{\mathcal{Q}} \times \mathcal{D} | d = d_h\}$ and the event *having a document in* $\mathcal{D}' \subset \mathcal{D}$, noted $E(\mathcal{D}')$ or $\mathcal{D}'$, which is the union of $E(d)$ for all $d \in D_h$.
- The event *having the query* q_h, noted $E(q_h)$ or $\underline{q_h}$, $E(q_h) = \{(q,d) \in \mathcal{Q} \times \mathcal{D} | \underline{q} = \underline{q}_h\}$.
- The event *relevance*, noted $E(R)$ or R, $E(R) = \{(q,d) \in \mathcal{Q} \times \mathcal{D} | d \text{ is a good answer for } q\}$.
- The event *the characteristic c is in the query*, noted $E(Q_c)$ or Q_c, $E(Q_c) = \{\{(q,d) \in \mathcal{Q} \times \mathcal{D} | c \in \underline{q}\}$.

B Document - anti-document relationship

The definition of the document event implies that for $d \neq d'$ we have:

$$P(D'|\underline{q},R) = P(\vee_{d \in D'} d|\underline{q},R) = \sum_{d \in D'} P(d|\underline{q},R) \tag{4}$$

With the notation $\neg d = E(\mathcal{D} \setminus \{d\})$,

$$\begin{aligned}
P(R|\underline{q},\neg d) &= \frac{P(R|\underline{q})}{P(\neg d|\underline{q})} P(\neg d|\underline{q},R) \text{ with Bayes,} \\
&= \frac{P(R|\underline{q})}{P(\neg d|\underline{q})} (1 - P(d|\underline{q},R)) \text{ with (4)} \\
&= \frac{P(R|\underline{q})}{P(\neg d|\underline{q})} \left(1 - \frac{P(d|\underline{q})}{P(R|\underline{q})} P(R|d,\underline{q})\right) \text{ with Bayes} \\
&= \frac{P(R|\underline{q}) - P(d)P(R|d,\underline{q})}{P(\overline{R}|\underline{q}) - P(d)P(\overline{R}|d,\underline{q})}
\end{aligned}$$

$$\text{Then } \frac{P(R|\underline{q},\neg d)}{P(\overline{R}|\underline{q},\neg d)} = \frac{P(R|\underline{q}) - P(d)P(R|d,\underline{q})}{(1 - P(R|\underline{q})) - P(d)(1 - P(R|d,\underline{q}))} \tag{5}$$

With formula 5, one can show that $\frac{P(R|\underline{q},\neg d)}{P(\overline{R}|\underline{q},\neg d)}$ is a strictly increasing function of $P(R|\underline{q},d)$. We can thus create an order which respect the *Probability Ranking Principle* by following the natural ordering of $-\frac{P(R|\underline{q},\neg d)}{P(\overline{R}|\underline{q},\neg d)}$. Using 1, we obtain

$$\begin{aligned}
-\log \frac{P(R|\underline{q},\neg d)}{P(\overline{R}|\underline{q},\neg d)} &= -\log \frac{P(R|\neg d)}{P(\overline{R}|\neg d)} - \log \frac{P(\underline{q}|R,\neg d)}{P(\underline{q}|\overline{R},\neg d)} \\
&= -\log \frac{P(R)}{P(\overline{R})} - \sum_{c \in \underline{q}} \log \frac{P(Q_c|R,\neg d)}{P(Q_c|\overline{R},\neg d)} \text{ using hypothesis 1, 2 and 3.} \qquad (6)
\end{aligned}$$

C Probability estimate

The probability of having a term c in a query for which the document d is relevant is $p_{c,d} = P(Q_c \wedge d \wedge R)$. Let $X^i_{c,d}$ be a random variable that takes the value 1 when the event $E(Q_c) \wedge E(d) \wedge E(R)$ is true, and 0 otherwise. $X^i_{c,d}$ thus follows a Bernoulli law with parameter $p_{c,d}$. Let $Y_{c,d} = X^1_{c,d} + \cdots + X^{N_q}_{c,d}$ be a random variable denoting the number of times document d is relevant to a query which contains the term c, where N_q denotes the number of past queries for which we have feedback. We can easily make the assumption that $X^i_{c,d}$ and $X^j_{c,d}$, $i \neq j$, are independent. As a result, $Y_{c,d}$ is Binomial distributed with parameters $p_{c,d}$ and N_q, and:

$$\begin{aligned}
P(Y_{c,d} = k) &= C^k_{N_q} p^k_{c,d} (1 - p_{c,d})^{|N_q| - k} \text{ for } k = 0, \cdots, |N_q| \\
\mathbb{E}(Y_{c,d}) &= |N_q| p_{c,d}
\end{aligned}$$

With $Y_{c,D'} = \sum_{d \in D'} Y_{c,d}$ we obtain:

$$\begin{aligned}
\mathbb{E}(Y_{c,D'}) &= |N_q| P(Q_c \wedge D' \wedge R) = |N_q| P(R) P(D'|R) P(Q_c|D',R) \\
&\approx R^*(c,\neg d) \\
\text{and } \frac{\mathbb{E}(Y_{c,\neg d})}{\mathbb{E}(Y_{c,\mathcal{D}})} &= P(D') P(Q_c|D',R) \\
&\approx \frac{R^*(c,\neg d)}{R^*(c,\mathcal{D})}
\end{aligned}$$

where $R^*(c,\neg d)$ and $\overline{R}^*(c,d)$ denote the expected value of $R(c,\neg d)$ and $\overline{R}(c,d)$ we would have if we knew the relevance judgment between each query and each document.

In order to estimate $R^*(c,D')$, we have to take into account the assumption on the equal relevance *a priori* of any document. It implies that the number of queries for which a document is relevant is the same. We denote by R_q this number. The value $\frac{R(c,D')+|D'|}{R(D')+2|D'|}$ is the ratio of query-document couples linked by the relevance relationship with a document in D' for which the query contains the term c. As a result, we get an approximation of $R^*(c,D')$:

$$R^*(c,D') \approx \frac{R(c,D') + |D'|}{R(D') + 2|D'|} \times |R_q| \times |D'|$$

Following, we have:

$$\frac{R^*(c,\neg d)}{R^*(c,\mathcal{D})} \approx \frac{\frac{R(c,\neg d)+|\mathcal{D}|-1}{R(\neg d)+2(|\mathcal{D}|-1)} \times |R_q| \times (|\mathcal{D}|-1)}{\frac{R(c,\mathcal{D})+|\mathcal{D}|}{R(\mathcal{D})+2|\mathcal{D}|} \times |R_q| \times |\mathcal{D}|}$$

and using the fact that $R(c,\neg d) = R(c,\mathcal{D}) - R(c,d)$ and $R(\neg d) = R(\mathcal{D}) - R(\neg d)$, we obtain an estimate of $P(Q_c|\neg d,R)$:

$$\tilde{P}(Q_c|\neg d,R) = \frac{1 - \frac{R(c,d)+1}{R(c,\mathcal{D})+|\mathcal{D}|}}{1 - \frac{R(d)+2}{R(\mathcal{D})+2|\mathcal{D}|}} \times \frac{|\mathcal{D}|-1}{|\mathcal{D}|}$$

A similar result can be obtain for $P(Q_c|\neg d,\overline{R})$ allowing us to state that:

$$\frac{\tilde{P}(Q_c|\neg d,R)}{\tilde{P}(Q_c|\neg d,\overline{R})} = \frac{1 - \frac{R(c,d)+1}{R(c,\mathcal{D})+|\mathcal{D}|}}{1 - \frac{\overline{R}(c,d)+1}{\overline{R}(c,\mathcal{D})+|\mathcal{D}|}} \times \frac{1 - \frac{\overline{R}(d)+2}{\overline{R}(\mathcal{D})+2|\mathcal{D}|}}{1 - \frac{R(d)+2}{R(\mathcal{D})+2|\mathcal{D}|}} \tag{7}$$

Recalling formula 6, we can thus derive a computable estimate of formula 1, where $c \in \underline{d}$ denote that $R(c,d) \neq 0$ or $\overline{R}(c,d) \neq 0$:

$$\begin{aligned}
-\log \frac{\tilde{P}(R|\underline{q},\neg d)}{\tilde{P}(\overline{R}|\underline{q},\neg d)} &= -\log \frac{P(R)}{P(\overline{R})} - \sum_{c \in \underline{q}} \log \frac{\tilde{P}(Q_c|\neg d,R)}{\tilde{P}(Q_c|\neg d,\overline{R})} \\
&= -\log \frac{P(R)}{P(\overline{R})} - \sum_{c \in \underline{q}} \log \frac{1 - \frac{\overline{R}(d)+2}{\overline{R}(\mathcal{D})+2|\mathcal{D}|}}{1 - \frac{R(d)+2}{R(\mathcal{D})+2|\mathcal{D}|}} - \sum_{c \in \underline{q} \cap \underline{d}} \log \frac{1 - \frac{R(c,d)+1}{R(c,\mathcal{D})+|\mathcal{D}|}}{1 - \frac{\overline{R}(c,d)+1}{\overline{R}(c,\mathcal{D})+|\mathcal{D}|}} - \sum_{c \in \underline{q} \setminus \underline{d}} \log \frac{1 - \frac{1}{R(c,\mathcal{D})+|\mathcal{D}|}}{1 - \frac{1}{\overline{R}(c,\mathcal{D})+|\mathcal{D}|}} \\
&= -\log \frac{P(R)}{P(\overline{R})} - |\underline{q}| \log \frac{1 - \frac{\overline{R}(d)+2}{\overline{R}(\mathcal{D})+2|\mathcal{D}|}}{1 - \frac{R(d)+2}{R(\mathcal{D})+2|\mathcal{D}|}} - \sum_{c \in \underline{q} \cap \underline{d}} \log \frac{1 - \frac{R(c,d)}{R(c,\mathcal{D})+|\mathcal{D}|-1}}{1 - \frac{\overline{R}(c,d)}{\overline{R}(c,\mathcal{D})+|\mathcal{D}|-1}} - \sum_{c \in \underline{q}} \log \frac{1 - \frac{1}{R(c,\mathcal{D})+|\mathcal{D}|}}{1 - \frac{1}{\overline{R}(c,\mathcal{D})+|\mathcal{D}|}} \\
&= K(q) - |\underline{q}| \log \frac{1 - \frac{\overline{R}(d)+2}{\overline{R}(\mathcal{D})+2|\mathcal{D}|}}{1 - \frac{R(d)+2}{R(\mathcal{D})+2|\mathcal{D}|}} - \sum_{c \in \underline{q} \cap \underline{d}} \log \frac{1 - \frac{R(c,d)}{R(c,\mathcal{D})+|\mathcal{D}|-1}}{1 - \frac{\overline{R}(c,d)}{\overline{R}(c,\mathcal{D})+|\mathcal{D}|-1}} \qquad (8)
\end{aligned}$$

where $K(q)$ is a function that depends only upon the query.