

PROBABILISTIC LOGICAL MODELLING OF THE BINARY INDEPENDENCE RETRIEVAL MODEL

Thomas Roelleke

Department of Computer Science, Queen Mary, University of London, England
thor@dcs.qmul.ac.uk

Jun Wang

Department of Computer Science, Queen Mary, University of London, England
wangjun@dcs.qmul.ac.uk

Keywords: Binary Independence Retrieval Model, Probabilistic Relational Modelling, Integration of Database and Information Retrieval (DB+IR)

Abstract: The binary independence retrieval (BIR) model is a main pillar of information retrieval (IR); recently, the model even attracted the attention of database research on ranking tuples for SQL queries. One of the problems with the BIR model is that though it is referred to as a probabilistic model, the retrieval status value actually lacks a probabilistic interpretation since the BIR model is based on the odd (fraction) of the relevance probabilities. This makes it hard to implement the BIR model in a probabilistic reasoning framework that aggregates and generates sound probabilities. Because of the growing impact of the BIR model for database research, and because the aggregation of the BIR term weights lacks a probabilistic meaning, we investigate in this paper the probabilistic relational implementations of the BIR model. This investigation led to the following findings: The probabilistic variants of the BIR model perform at least as good as the genuine model, where slightly refined variants outperform the genuine model, but cannot achieve the performance of *tf-idf*-based retrieval.

1 Introduction

The binary independence retrieval (BIR) model ([RSJ76]) is a main pillar of information retrieval (IR); recently, the model even attracted the attention of database (DB) research on ranking tuples for SQL queries ([CDHW06]).

One of the problems with the BIR model is that though it is referred to as a probabilistic model, the retrieval status value (*RSV*) actually lacks a probabilistic interpretation since the BIR model is based on the odds (fraction) of the relevance probabilities. This makes it hard to implement the BIR model in a probabilistic reasoning framework that is restricted to sound probabilistic operations.

Regarding theoretical work on how to give a probabilistic interpretation to retrieval models, [WY95] presented a unifying framework based on probabilistic inference, but this is applied only to traditional models such as the vector-space model. [RSWHBG94] can be viewed as an extension of the BIR model where the BIR model, a term weight and a

document length correction factor are composed into a retrieval status value, leading to the BM25 formula, one of the best performing retrieval models. In particular, this work led to the Poisson-based estimate of the term frequency component in *tf-idf*, which we will recall in this paper. [Rob04] relates *tf-idf*-based retrieval to the BIR model, and [Roe03] discusses an *idf*-based estimation and semantics of probabilities. The relationship of *tf-idf* and BIR, and the probabilistic semantics of an *idf*-based probability estimation form foundations of the probabilistic relational modelling of the BIR model.

Because of the growing impact of the BIR model for DB research, and because the aggregation of the BIR term weights lacks a probabilistic meaning, we investigate in this paper the probabilistic relational implementations of the BIR model.

For facilitating the usage of IR in databases, [FR97] proposed a probabilistic relational algebra for the integration of IR and DB. The algebra is capable of describing probability aggregation, however, there is a major short-coming: probability estimation can-

not be expressed. This short-coming could be solved by introducing a new probabilistic relational operator, namely the relational Bayes ([RWVA07]). Though the relational Bayes improves the expressiveness of probabilistic relational reasoning, the BIR model in its genuine form cannot be expressed. This is because the genuine BIR formulation relies on the odds of probabilities (fraction of a probability and its complement). The odds of an event has no probabilistic interpretation. Actually, the odds value is in the interval $[0, \infty]$, which already excludes that the odds value is input to further reasoning in a probabilistic reasoning chain.

To generalise and facilitate the meaning and usage of the BIR model, we look in this paper at probabilistic variants of the BIR model. In particular, we investigate the retrieval quality of the genuine formulation against a number of alternatives, where the alternatives try to achieve a probabilistic interpretation of the *RSV* of the BIR model.

The remainder of this paper is structured as follows: We briefly review BIR foundations in section 2, and foundations of probabilistic relational modelling in section 3. Then, section 4 presents BIR implementations in a probabilistic relational framework. Section 5 reports experimental results for the BIR implementations, and we discuss the results in section 6.

2 BIR Foundations

The retrieval status value (*RSV*) of the BIR model is derived from the probability $P(r|d, q)$ of relevance (where r is the relevance event, and $d \wedge q$ is a document-query pair). Details of this derivation can be found in IR text books (e.g. [GF04]), and [RW06] reviews the derivation of the BIR model in parallel to the Poisson model and language modelling.

The derivation is based on the odds $O(r|d, q)$:

$$O(r|d, q) := \frac{P(r|d, q)}{P(\bar{r}|d, q)} \quad (1)$$

In this paper, we do not repeat the derivation. We just summarise it briefly: The application of Bayes' theorem, and an assumption regarding the independence of term events, and an assumption regarding the occurrence of non-query terms, lead to the following formulas:

$$w_t := \frac{P(t|r) \cdot P(\bar{t}|\bar{r})}{P(t|\bar{r}) \cdot P(\bar{t}|r)} \quad (2)$$

$$RSV_{\text{BIR}}(d, q) := \sum_{t \in d \cap q} \log w_t \quad (3)$$

$$O(r|d, q) \propto RSV_{\text{BIR}}(d, q)$$

Here, w_t is the term relevance weight of the BIR model. The set of non-relevant documents (\bar{r}) can be explicit, can be derived from the set of retrieved documents, or it can be approximated by the collection c , where the justification is that the statistics in \bar{r} are approximately the same as in c .

The BIR *RSV* can be rewritten by applying the definition of the inverse document frequency (*idf*):

$$n_D(t, x) := \text{number of documents in set } x \text{ in which term } t \text{ occurs}$$

$$N_D(x) := \text{number of documents in set } x$$

$$P(t|x) := \frac{n_D(t, x)}{N_D(x)} \quad (4)$$

$$idf(t, x) := -\log P(t|x) \quad (5)$$

With these definitions, and with $\bar{r} = c$ (i.e. the statistics in non-relevant is approximated by the statistics in the whole collection), the RSV_{BIR} becomes:

$$RSV_{\text{BIR}}(d, q) := \sum_{t \in d \cap q} idf(t, c) - idf(t, r) + idf(\bar{t}, r) - idf(\bar{t}, c) \quad (6)$$

This linear combination of *idf*-values forms the basis of the probabilistic relational modelling of the BIR model presented in section 4. To prepare for the probabilistic relational implementation, we present in the next section foundations of probabilistic relational modelling.

3 Probabilistic Relational Modelling

For a pragmatic introduction into probabilistic relational modelling, consider the following example. Assume we have a relational representation of a *tf*-based document index, an *idf*-based term space, and a query.

tf		
$P(t d)$	Term	DocId
0.5	sailing	doc1
0.5	boats	doc1
0.6	sailing	doc2
0.4	boats	doc2

idf		
$P(t c)$	Term	Collection
0.1	sailing	c1
0.8	boats	c1

query	
Term	QueryId
sailing	q1
boats	q1

Given such a knowledge representation, *tf-idf*-based retrieval can be described.

Consider the formulation in PSQL (Probabilistic SQL) and PRA (Probabilistic Relational Algebra).

```
-- idf-based query term weighting:
CREATE VIEW weightedQuery AS
  SELECT ALL Term, QueryId
  FROM query, idf
  WHERE query.Term = idf.Term;

-- tf-idf-based retrieval:
CREATE VIEW retrieve AS
  SELECT DISJOINT Term, QueryId
  FROM weightedQuery, tf
  WHERE weightedQuery.Term = tf.Term;
```

The above PSQL program is equivalent to the following PRA program:

```
# idf-based query term weighting:
weightedQuery =
  Project ALL[$1,$2](
    Join[$1=$1](query, idf));

# tf-idf-based retrieval:
retrieve =
  Project DISJOINT[$4,$2](
    Join[$1=$1](weightedQuery, tf));
```

For the relations (views) “weightedQuery” and “retrieve”, we obtain:

wQuery		
Prob	Term	QueryId
0.10	sailing	q1
0.80	boats	q1

retrieve		
Prob	DocId	QueryId
0.45	doc1	q1
0.38	doc2	q1

For example, $P_{\text{retrieve}}(\text{doc1}, \text{q1}) = 0.45$ is the result of $0.1 \cdot 0.5 + 0.8 \cdot 0.5$, where $P_{\text{wQuery}}(\text{sailing}, \text{c1}) = 0.1$, and $P_{\text{tf}}(\text{sailing}, \text{doc1}) = 0.5$, and so forth. The Join over the terms leads to the multiplication of probabilities, and the disjoint projection adds the products.

The probabilistic relational modelling of information retrieval is described in detail in [FR97, RAWC05, RWWA07]. The motivation of probabilistic relational modelling is to provide an open-box approach for implementing retrieval strategies, so that it becomes possible to describe a well-defined ranking of objects in a relational database, where ranking is applied for projects, persons, products, etc, i.e. to any object. For the DB/SQL-based approach to probabilistic ranking based on the BIR model, see [CDHW04, CDHW06], where the BIR model serves as a ranking model. Next, we focus on the probabilistic relational modelling of the BIR model, where for

the purpose of this paper, we restrict to the classical case of document retrieval only.

4 Probabilistic Relational Modelling of the BIR Model

The PRA program in figure 1 contains the equations (views) for implementing the probabilistic variants of the BIR model. The PSQL program in figure 2 shows the PSQL program equivalent to the PRA program in figure 1.

The programs are structured into three parts:

1. A basic declaration block.
2. The definitions of the main probabilistic relations (*idf_c*, *idf_r*, *wQuery_c*, *wQuery_r*, etc).
3. The definition of the relation *bir_retrieve* to take the retrieval result.

For understanding the meaning of the PRA/PSQL programs, consider in the following a running example, for which we show the instantiation of the main probabilistic relations. The running example is organised such that we have ten documents, twenty term-document tuples, and four relevant documents.

Coll		
Prob	Term	DocId
1.0	sailing	doc1
1.0	boats	doc1
1.0	sailing	doc2
1.0	boats	doc2
1.0	sailing	doc2
1.0	sailing	doc3
1.0	east	doc3
1.0	coast	doc3
1.0	sailing	doc4
1.0	boats	doc5
1.0	sailing	doc6
1.0	boats	doc6
1.0	east	doc6
1.0	coast	doc6
1.0	sailing	doc6
1.0	boats	doc6
1.0	boats	doc7
1.0	east	doc8
1.0	coast	doc9
1.0	sailing	doc10

Relevant		
Prob	QueryId	DocId
1.0	q1	doc2
1.0	q1	doc4
1.0	q1	doc6
1.0	q1	doc8

```

#####
# Part 1: Basic declarations

# queries(QueryId):
queries = Project distinct[$2](Query);

# relevantDocs(QueryId, DocId):
relevantDocs = Project[$1,$3](Join[$1=$1](queries, Relevant));

# relColl(Term, DocId):
relColl = Project[$3,$4](Join[$2=$2](relevantDocs, Coll));

# distinct collection
distinctColl = Project distinct(Coll);

#####
# Part 2: Term probabilities and aggregation
# Part 2.1: Term probabilities

p_t_c = Bayes df[](Project[$1](Coll));
p_t_r = Bayes df[](Project[$1](relColl));

# idf for whole collection (idf_c) and
# idf for the collections constructed from relevant documents (idf_r).
idf_c = Bayes max_idf[](Project all[$1](Coll));
idf_r = Bayes max_idf[](Project all[$1](relColl));

# Query term probabilities:
wQuery_c = Project all[$1,$2](SELECT[$1=$3](Join(Query, idf_c)));
wQuery_r = Project all[$1,$2](SELECT[$1=$3](Join(Query, idf_r)));
norm_wQuery_c = Bayes[$2](wQuery_c);
norm_wQuery_r = Bayes[$2](wQuery_r);

# Part 2.2: Aggregation of query term probabilities
wQuery_subsumed = Subtract subsumed (wQuery_c, wQuery_r);
wQuery_independent = Subtract independent(wQuery_c, wQuery_r);
norm_wQuery_subsumed =
  Subtract subsumed(norm_wQuery_c, norm_wQuery_r);
norm_wQuery_independent =
  Subtract independent(norm_wQuery_c, norm_wQuery_r);

#####
# Part 3: Retrieval

bir_retrieve = Project disjoint[$4,$2](Join[$1=$1](wQuery, collIndex));

# Set wQuery and indexColl according to strategy. For example:
wQuery = wQuery_subsumed;
collIndex = distinctColl;

```

Figure 1: Implementations of BIR model in PRA

```

-----
-- Part 1: Basic declarations:
CREATE VIEW queries AS SELECT QueryId FROM Query;

CREATE VIEW relevantDocs AS
  SELECT QueryId, DocId FROM queries, Relevant
  WHERE queries.Queryid = Relevant.QueryId;

CREATE VIEW relColl AS
  SELECT Coll.Term, Coll.DocId FROM relevantDocs, Coll
  WHERE relevantDocs.DocId = Coll.DocId;

CREATE VIEW distinctColl AS SELECT DISTINCT Term, DocId FROM Coll;
-----
-- Part 2: Term probabilities and their aggregation:
-- Part 2.1: Term probabilities:
CREATE VIEW idf_c AS
  SELECT Term FROM Coll ASSUMPTION MAX_IDF EVIDENCE KEY ();
CREATE VIEW idf_r AS
  SELECT Term FROM relColl ASSUMPTION MAX_IDF EVIDENCE KEY ();

CREATE VIEW wQuery_c AS
  SELECT Term, QueryId FROM Query, idf_c WHERE Query.Term = idf_c.Term;
CREATE VIEW wQuery_r AS
  SELECT Term, QueryId FROM Query, idf_r WHERE Query.Term = idf_r.Term;

CREATE VIEW norm_wQuery_c AS
  SELECT Term, QueryId FROM wQuery_c
  ASSUMPTION DISJOINT EVIDENCE KEY (QueryId);
CREATE VIEW norm_wQuery_r AS
  SELECT Term, QueryId FROM wQuery_r
  ASSUMPTION DISJOINT EVIDENCE KEY (QueryId);

-- Part 2.2: Term probability aggregation:
CREATE VIEW wQuery_subsumed AS
  wQuery_c MINUS SUBSUMED wQuery_r;
CREATE VIEW norm_wQuery_subsumed AS
  norm_wQuery_c MINUS SUBSUMED norm_wQuery_r;
CREATE VIEW wQuery_independent AS
  wQuery_c MINUS INDEPENDENT wQuery_r;
CREATE VIEW norm_wQuery_independent AS
  norm_wQuery_c MINUS INDEPENDENT norm_wQuery_r;
-----
-- Part 3: Retrieval:
CREATE VIEW birm_retrieve AS
  SELECT DISJOINT DocId, QueryId FROM wQuery, collIndex
  WHERE wQuery.Term = collIndex.Term;
-- Set wQuery and collIndex according to strategy. For example:
CREATE VIEW wQuery AS SELECT Term, QueryId FROM wQuery_subsumed;
CREATE VIEW collIndex AS SELECT Term, DocId FROM distinctColl;

```

Figure 2: Implementations of BIR model in PSQL

Query		
Prob	QueryId	DocId
1.0	sailing	q1
1.0	boats	q1

Central to the implementation of the BIR model are the two probabilistic relations idf_c and idf_r : idf_c is based on the discriminativeness of a term in the collection (c denotes the collection), and idf_r is based on the discriminativeness of a term in the set of the relevant document (r denotes the set of relevant documents). The relations idf_c and idf_r are based on the document-based probabilities $P(t|c)$ and $P(t|r)$, i.e. the probabilities that term t occurs in the respective set of documents.

The occurrence-based probabilities $P(t|c)$ and $P(t|r)$ are generated by the new probabilistic relational operator, the relational Bayes ([RWVA07]). Basically, the relational Bayes performs a computation that leads to the estimate $P(t|x) = \frac{n_D(t,x)}{N_D(x)}$, where x is either the collection c or the set r of relevant documents, $n_D(t,x)$ is the number of documents in set x , and $N_D(x)$ is the total number of documents.

For our running example, we obtain the following relations and tuple probabilities:

p.t.c	
Prob	Term
0.600000	sailing
0.500000	boats
0.300000	east
0.300000	coast

p.t.r	
Prob	Term
0.750000	sailing
0.500000	boats
0.500000	east
0.250000	coast

idf_c	
Prob	Term
0.424283	sailing
0.575717	boats
1.000000	east
1.000000	coast

idf_r	
Prob	Term
0.207519	sailing
0.500000	boats
0.500000	east
1.000000	coast

There are $n_D(sailing,c) = 6$ sailing documents, and $N_D(c) = 10$. Then, for example, $P_{p.t.c}(sailing) = 6/10 = 0.6$, $P_{p.t.c}(boats) = 5/10 = 0.5$, and $P_{p.t.r}(sailing) = 3/4 = 0.75$. $P_{p.t.r}(boats) = 2/4 = 0.5$. The expressions with Bayes max_idf perform an idf-based probability estimation. This corresponds to a normalisation of the form $\log(P_{p.t.c}(t))/\log(P_{p.t.c}(t_{min}))$ and yields, for example, $P_{idf.c}(sailing) \approx 0.42$, and $P_{idf.c}(boats) \approx 0.57$.

The query terms are weighted with the idf -based probabilities. This leads to the following two relations:

wQuery_c		
Prob	Term	QueryId
0.424283	sailing	q1
0.575717	boats	q1

wQuery_r		
Prob	Term	QueryId
0.207519	sailing	q1
0.500000	boats	q1

Next, we approach the critical step, namely the aggregation of the query term probabilities. To replace $-\log(P(t|c)/P(t|r))$, which is large for $P(t|c) < P(t|r)$, i.e. for terms that are more likely to occur in relevant documents than in the documents of the collection, we work with two approaches to model a probabilistic subtraction: A subsumed and an independent subtraction. The log-odds is equivalent to $idf(t,c) - idf(t,r)$. The idf -based probability estimation yields two idf -based probabilities $P(t \text{ is informative}|c)$ and $P(t \text{ is informative}|r)$ proportional to the respective idf -values. Then, we implement for the subsumed case $P(t \text{ is informative}|c) - P(t \text{ is informative}|r)$, and for the independent case $P(t \text{ is informative}|c) \cdot (1 - P(t \text{ is informative}|r))$. This yields for the running example the following relations:

wQuery_subsumed		
Prob	Term	QueryId
0.216765	sailing	q1
0.075717	boats	q1

wQuery_independent		
Prob	Term	QueryId
0.336237	sailing	q1
0.287858	boats	q1

For example, the probability for boats in the subsumed case is $0.57 - 0.5 = 0.07$, and in the independent case, we obtain $0.57 * (1 - 0.5) \approx 0.28$. The example illustrates the numerical effect of the probabilistic assumption. This effect is even stronger for normalised query term probabilities, as we illustrate next.

Normalised query term probabilities are based on a disjoint and exhaustive space of events (i.e. sum over probabilities equal to 1.0). This normalisation forms an alternative to the non-normalised “wQuery” relations.

norm_wQuery_c		
Prob	Term	QueryId
0.424283	sailing	q1
0.575717	boats	q1

norm_wQuery_r		
Prob	Term	QueryId
0.293305	sailing	q1
0.706695	boats	q1

The normalised relations (with disjoint and exhaustive tuples) allow for a well-defined disjoint projection at retrieval time. The non-normalised “wQuery” relations implement a disjunctive interpretation of the query, whereas the normalised “norm_wQuery” relations implement a more conjunctive interpretation of the query. For the subsumed and independent aggregation of the normalised query probabilities, we obtain:

norm_wQuery_subsumed		
Prob	Term	QueryId
0.130978	sailing	q1
0.000000	boats	q1

norm_wQuery_independent		
Prob	Term	QueryId
0.299839	sailing	q1
0.168861	boats	q1

Note that due to the normalisation, boats is viewed more discriminative (rarer) in the relevant documents than in the collection (probability of boats in norm_wQuery_r greater than in norm_wQuery_c). Therefore, in the subsumed case, the probability zero is assigned to boats since it is viewed as a poor term to retrieve relevant documents while not retrieving non-relevant documents.

This section illustrated the generation of the core probabilistic relations applied for implementing the BIR model in a probabilistic relational reasoning framework. The relations shown in this section explain the effect of the PSQL/PRA programs shown in figure 1 and figure 2. The high-level abstraction of retrieval functions leads to optimal flexibility and reusability, and these are main motivation for modelling IR in a probabilistic relational framework.

5 Experiments

Experiments were carried out on two collections: The smallish CACM collection (3204 documents, 2.2 MB of source data), and the medium-sized INEX collection (approx 12,000 documents/articles, 550 MB of source data, 18,000,000 XML elements, i.e. in average each document contains 1,500 elements). Hereby, the main purpose of the smallish collection is to help in the set-up phase of the experiments, and with publication, we deliver the

PRA/PSQL scripts so that other can re-run and/or extend the experiments.

Based on the different approaches to implement the BIR model in a probabilistic reasoning system, we find two dimensions: The aggregation of probabilities (how to combine idf_c and idf_r), and the retrieval (whether to retrieve from a distinct term-document representation or a non-distinct representation).

Aggregation of probabilities:

1. BIR term weight non-normalised, subsumed: Subtract subsumed(wQuery_c, wQuery_r)
2. BIR term weight non-normalised, independent: Subtract independent(wQuery_c, wQuery_r)
3. BIR query term weight normalised, subsumed: Subtract subsumed(norm_wQuery_c, norm_wQuery_r)
4. BIR query term weight normalised, independent: Subtract independent(norm_wQuery_c, norm_wQuery_r)

Retrieval: Based on distinct Coll, non-distinct Coll, tf_sum, tf_max, or tf_poissona. For the tf -variants, the definitions are based on $n_L(t, d)$, which denotes the number of locations at which term t occurs in document d .

1. Join[\$1=\$1](wQuery, distinctColl): Considers term weight maximal ones per document.
2. Join[\$1=\$1](wQuery, Coll): Considers term weight to the power of $n_L(t, d)$ in a document.
3. Join[\$1=\$1](wQuery, tf_sum): Aggregate the non-distinct tuples before retrieval into a distinct tf relation. $tf_sum(t, d) := \frac{n_L(t, d)}{\sum_{t'} n_L(t', d)} = \frac{n_L(t, d)}{N_L(d)}$. High emphasis on the document length: Large tf -values for short documents, small tf -values for long documents.
4. Join[\$1=\$1](wQuery, tf_max): $tf_max(t, d) := \frac{n_L(t, d)}{\max_{t'} n_L(t', d)} = \frac{n_L(t, d)}{n_L(t_{max}, d)}$. This tf -variant has less emphasis on document length than tf_sum has.
5. Join[\$1=\$1](wQuery, tf_poissona): $tf_poissona(t, d) := \frac{n_L(t, d)}{1 + n_L(t, d)}$. Since the curve of this non-linear estimate is similar to the Poisson distribution, and also because the tf -value can be interpreted as a Poisson-like probability, we refer to the $n/(K + n)$ -variant as a Poisson approximation.

With four candidates on the aggregation dimension, and five candidates on the retrieval dimension, this leads from a combinatorial point of view to twenty possible implementations. Since the result of the aggregation shows that the subsumed aggregation of non-normalised query term weights is

the best probabilistic relational implementation of the BIR model, we choose only this best candidate to be combined with the *tf*-based retrieval functions. Thus, table 3 shows the quality results for fourteen of the twenty possible implementations.

For a graphical overview over the results, consider the sub-figures in figure 4 and figure 5.

6 Discussion

The genuine BIR model shows surprisingly strong performance for the smallish CACM collection, but this cannot be confirmed for the larger INEX collection. Similar, there is strong performance of *tf*_max with independent aggregation of *idf*-based term weights, where again, this cannot be confirmed for the bigger collection. The strong performance of these variants is to be explained by the smallish nature of the CACM collection.

The graphical overview underlines the more substantial and outstanding performance of the *tf*_poissona-based retrieval for the INEX collection. The probabilistic relational implementation closest up to this best candidate are the implementations that join over a non-distinct representation of the collection index, i.e. there is a built-in *tf* in these implementations of the BIR model. The BIR model with a non-distinct collection index can be explained by replacing the distinct document view (document is a set of distinct terms) in the BIR model with a location-oriented view (document is a set of multiple terms, i.e. multiple occurrences are preserved).

Algebraically, the following disjoint projection is executed for retrieval: `Project disjoint[$docId, $querId](Join[$term=$term](wQuery, collIndex))`. This algebra expression corresponds to the following probabilistic form:

$$\sum_{t \in d \cap q} P_{wQuery}(t, q) \cdot P_{collIndex}(t, d)$$

Hereby, *each* occurrence of *t* in *d* is considered, i.e. the set $d \cap q$ contains non-distinct elements. Then, the *idf*-based query term probability in “wQuery” is considered as many times as the respective term occurs in the retrieved document.

This finding, namely that *tf*-sensitive implementations of BIR perform better than implementations based on a distinct collection index, matches expectation. A more surprising and important result is that for the medium-size collection INEX, all probabilistic variants are better or at least as good as the genuine model. Even the variants based on a distinct collection index perform as good or better than the genuine model. This allows to draw the conclusion that

the penalising effect of the genuine model (bad terms may have a negative effect on the RSV) is not significant to retrieval, maybe even counter-productive. This was a very important finding and confirmation for our research on probabilistic logical modelling, where, in principle, we could work with negative probabilities, however, as one might expect, the inclusion of negative probabilities significantly increases the task to guarantee well-formed probabilistic logical expressions. Overall, the finding that the most basic probabilistic variants lead to retrieval performance comparable to the genuine BIR model, and that the slightly more refined variants (based on a non-distinct collection index) clearly outperform the genuine model, support the hypothesis that probabilistic logical modelling leads to good retrieval quality, even though the modelling is restricted to a sound probabilistic framework.

7 Summary and Conclusion

This paper presented and investigated probabilistic relational implementation of the BIR model. The BIR model is a main foundation of IR. Since it is based on log-odds, the retrieval status value lacks a probabilistic interpretation. This is a problem in probabilistic reasoning, where the requirement is to produce well-defined probabilities, so that the result can serve as input to further reasoning. Also, a sound probabilistic reasoning framework would leave the probabilistic paradigm when introducing log-odds.

Therefore, we investigated probabilistic relational/logical implementations of the BIR model. The approach is to replace odds by a subtraction of the respective probabilities, where for the subtraction, it makes sense to consider independence and inclusion assumptions. Furthermore, for the retrieval, we investigated retrieval based on the distinct collection index (as required for the *binary* independence retrieval model), versus retrieval based on the original non-distinct collection index where a term-document pair may occur multiple times. The join with the non-distinct collection is in a probabilistic relational framework easy to perform, since it even frees the computation from building a distinct collection index.

The overall result is that the probabilistic variants of the genuine (odds-based) BIR model perform, for a sufficiently large collection, at least as good as the genuine model, where the retrieval based on the non-distinct representation leads to significantly better retrieval quality (as expected, since non-distinct representation builds in an implicit term frequency effect). We compared the performance to genuine *tf-idf*-

BIR implementation	CACM		INEX	
	MAP	P@10	MAP	P@10
Genuine	0.2845	0.5625	0.0861	0.2190
non-normalised, distinct, subsumed	0.2096	0.4242	0.1076	0.2782
non-normalised, non-distinct, subsumed	0.2457	0.5353	0.1531	0.4233
normalised, distinct, subsumed	0.1488	0.3074	0.0885	0.2279
normalised, non-distinct, subsumed	0.2066	0.4589	0.1497	0.4102
non-normalised, tf_max, subsumed	0.2885	0.6005	0.2659	0.6176
non-normalised, tf_poissona, subsumed	0.3019	0.5792	0.4060	0.8177
non-normalised, tf_sum, subsumed	0.2265	0.4916	0.2426	0.4944
non-normalised, distinct, independent	0.2133	0.4547	0.1027	0.2584
non-normalised, non-distinct, independent	0.2266	0.5089	0.1446	0.4218
normalised, distinct, independent	0.2755	0.5113	0.1005	0.2451
normalised, non-distinct, independent	0.2601	0.5088	0.1474	0.4206
non-normalised, tf_max, independent	0.2854	0.6145	0.2334	0.5557
non-normalised, tf_poissona, independent	0.3005	0.5859	0.3294	0.6746
non-normalised, tf_sum, independent	0.2211	0.4973	0.2241	0.4895

Figure 3: MAP and P@10

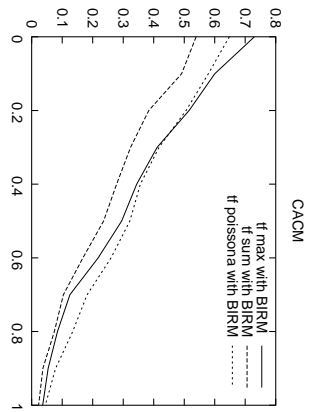
based retrieval, and found that the pre-computation of tf (based on $tf(t, d) = n(t, d)/(1 + n(t, d))$), which can be viewed as a Poisson approximation) clearly outperforms the BIR strategies with the non-distinct collection index and its implicit tf .

In this paper, we focussed on the probabilistic logical implementation and the retrieval quality. In future work, we will report on the computational costs regarding the processing of the BIR implementations.

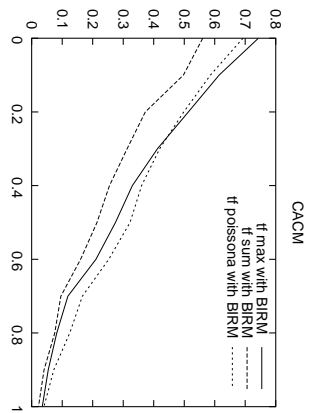
The PRA/PSQL scripts in this paper are available for download.

REFERENCES

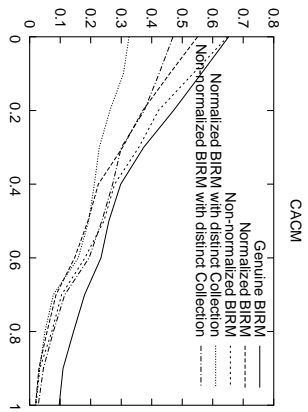
- [CDHW04] Surajit Chaudhuri, Gautam Das, Vagelis Hristidis, and Gerhard Weikum. Probabilistic ranking of database query results. In *VLDB*, pages 888–899, 2004.
- [CDHW06] Surajit Chaudhuri, Gautam Das, Vagelis Hristidis, and Gerhard Weikum. Probabilistic information retrieval approach for ranking of database query results. *ACM Trans. Database Syst.*, 31(3):1134–1168, 2006.
- [FR97] N. Fuhr and T. Roelleke. A probabilistic relational algebra for the integration of information retrieval and database systems. *ACM Transactions on Information Systems (TOIS)*, 14(1):32–66, 1997.
- [GF04] David A. Grossman and Ophir Frieder. *Information Retrieval. Algorithms and Heuristics, 2nd ed.*, volume 15 of *The Information Retrieval Series*. Springer, 2004.
- [RAWC05] Thomas Roelleke, Elham Ashoori, Hengzhi Wu, and Zhen Cai. The QMUL IR team with probabilistic SQL at enterprise TREC. In *Text Retrieval Conference (TREC) of National Institute of Standards and Technologies (NIST)*, Washington, 2005.
- [Rob04] S.E. Robertson. Understanding inverse document frequency: On theoretical arguments for idf. *Journal of Documentation*, 60:503–520, 2004.
- [Roe03] Thomas Roelleke. A frequency-based and a Poisson-based probability of being informative. In Jamie Callan, Gordon Cormarck, Charles Clarke, David Hawking, and Alan Smeaton, editors, *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Toronto, Canada*, pages 227–234, 2003.
- [RSJ76] S.E. Robertson and K. Sparck Jones. Relevance weighting of search terms. *Journal of the American Society for Information Science*, 27:129–146, 1976.



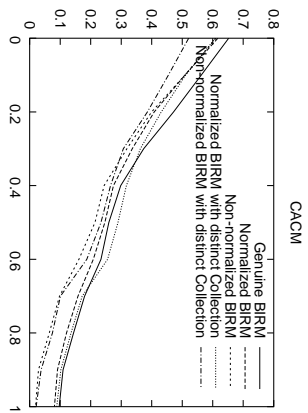
(b) subsumed, tf



(d) independent, tf

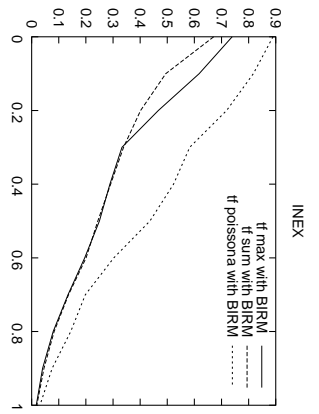


(a) subsumed

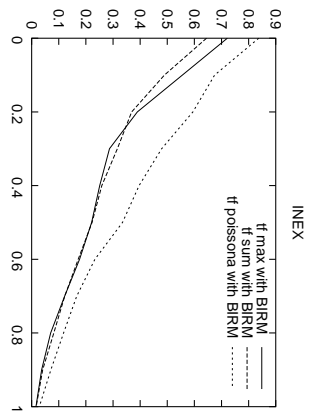


(c) independent

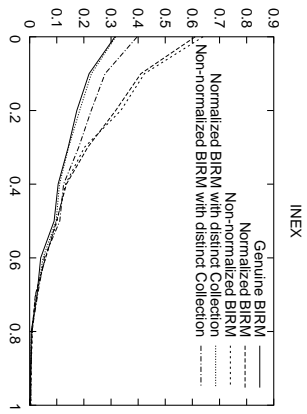
Figure 4: Precision/Recall of the genuine BIR versus probabilistic implementations: CACM collection



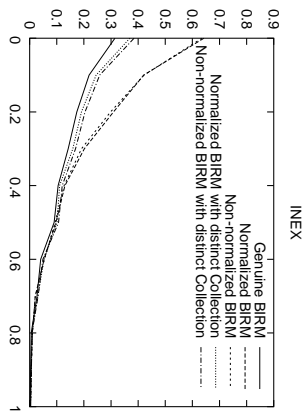
(b) subsumed, tf



(d) independent, tf



(a) subsumed



(c) independent

Figure 5: Precision/Recall of the genuine BIR versus probabilistic implementations: INEX collection

- [RSWHBG94] S. Robertson, S. Jones S. Walker, M.M. Hancock-Beaulieu, and M. Gatford. Okapi at trec-3. In *Text REtrieval Conference*, 1994.
- [RW06] Thomas Roelleke and Jun Wang. A parallel derivation of probabilistic information retrieval models. In *ACM SIGIR*, 2006.
- [RWWA07] Thomas Roelleke, Heng Zhi Wu, Jun Wang, and Hany Azzam. Modelling retrieval models in a probabilistic relational algebra with a new operator: The relational bayes. *VLDB Journal*, 2007. to appear.
- [WY95] S.K.M. Wong and Y.Y. Yao. On modeling information retrieval with probabilistic inference. *ACM Transactions on Information Systems*, 13(1):38–68, 1995.