HyREX at INEX 2003

Mohammad Abolhassani, Norbert Fuhr, Saadia Malik
University of Duisburg-Essen, Germany

ABSTRACT

Abstract: In this paper, we describe two new approaches for processing INEX queries. For CO queries, we adopt Amati's divergence from randomness approach (aka language model) and extend it by an additional factor for considering the hierarchical level of the element to be retrieved. For CAS queries, we investigate several mappings from INEX queries to our query language XIRQL, where we tried to introduce different degrees of vagueness. Both approaches yield good retrieval results, but still leave room for improvement.

1. INTRODUCTION

The HyREX (Hypermedia Retrieval Engine for XML) system developed by our group [Fuhr & Großjohann 01], [Fuhr & Großjohann 04], [Fuhr et al. 02] supports document ranking based on index term weighting, specificity-oriented search for retrieving the most relevant parts of documents, data types with vague predicates for dealing with specific types of content and structural vagueness for vague interpretation of structural query conditions. In INEX 2002, HyREX performed very well for content-only (CO) queries, but only poorly for content-and-structure(CAS) queries (although this was due to a bug in the implementation).
In this paper, we describe a new retrieval model for CO queries based on Amati's divergence from randomness (DFR) approach. For the CAS queries, we investigated several methods for transforming INEX topics into our own query language XIRQL [Fuhr & Großjohann 01].

2. CONTENT-ONLY QUERIES

In [Fuhr & Großjohann 01], we proposed the *augmentation* method for processing content-only queries. This method gave very good results in INEX 2002. In the augmentation approach, standard term weighting formulas (we were using the BM25 formula [Robertson et al. 95] for this purpose) are used for indexing the leave nodes of the document tree. For computing the indexing weights of inner nodes, the weights from the leaves are propagated towards the inner nodes. During propagation, however, the weights are down-weighted by multiplying them with a so-called augmentation factor. This down-weighting happens whenever the indexing weight is propagated from an element that belongs to a predefined set of so-called index node root elements to its parent element. In case a term at an inner note receives propagated weights from several leaves, we compute the overall term weight by assuming a probabilistic disjunction of the leaf term weights. This way, more specific elements are preferred during retrieval

This year, we were trying to adopt the DFR approach, which is a kind of language model. Here we give only a brief description of the application of this approach to XML retrieval. A more detailed presentation can be found in [Abolhassani & Fuhr 04].

2.1 The DFR approach

[Amati & Rijsbergen 02] introduce a framework for deriving probabilistic models of IR. These models are non-parametric models of IR as obtained in the *language model* approach. The term weighting models are derived by measuring the divergence of the actual term distribution from that obtained under a random process.
In this framework, the weighting formula for a term in a document is the product of the following two factors:

1. $Prob_1$ is used for measuring the *information content* of the term in a document, and $(-\log_2 Prob_1)$ gives the corresponding amount of information.

2. $Prob_2$ is used for measuring the *information gain* of the term with respect to its 'elite' set (the set of all documents in which the term occurs). The less the term is expected in a document with respect to its frequency in the elite set (measured by the counter-probability $(1 - Prob_2)$), the more the amount of information is gained with this term.

Then the weight of a term in a document is defined as:

$$w = (1 - Prob_2) \cdot (-\log_2 Prob_1) = Inf_2 \cdot Inf_1 \qquad (1)$$

For computing the two probabilities, the following parameters are used:

N number of documents in the collection,

tf term frequency within the document (since different normalisations are applied to the term frequency, we use tf_1 and tf_2 in the following formulas),

n size of the elite set of the term,

F term frequency in elite set.

Furthermore, let $\lambda = F/N$ in the following.
As probability distribution for estimating $Prob_1$, different probabilistic models are regarded in [Amati & Rijsbergen 02]. In this paper, we use only two of them:

- The **binomial** model assumes that the F term occurrences are distributed independently over the N document; thus, we have a binomial distribution with $p = 1/N$. Approximating the binomial formula with the divergence yields:

$$Inf_1 = tf_1 \cdot \log_2 \frac{tf_1}{\lambda} + \left(\lambda + \frac{1}{12tf_1} - tf_1\right)$$

Table 1: Results from direct application vs. augmentation approach

document length	Dynamic		Fixed	
	B Norm.	L Norm.	B Norm.	L Norm.
Binomial	0.0109	0.0356	0.0640	0.0717
Bose-Einstein	0.0214	0.0338	0.0468	0.0606
Augmentation	0.1120			

Table 2: Results from 2nd normalisation with two basic values for β

	$\beta = 0$		$\beta = -1$	
	B Norm.	L Norm.	B Norm.	L Norm.
Binomial	0.0391	0.0586	0.0640	0.0900
Bose-Einstein	0.0376	0.0609	0.0376	0.0651

$$\cdot \log_2 e + 0.5 \log_2(2\pi \cdot tf_1) \quad (2)$$

- The **Bose-Einstein** model considers all possible distributions of the F term occurrences within the N documents and then considers all those events where the current document has tf_1 occurrences. The Geometric as limiting form of the Bose-Einstein model yields:

$$Inf_1 = -log_2 \frac{1}{1+\lambda} - tf_1 \cdot \log_2 \frac{\lambda}{1+\lambda} \quad (3)$$

For the parameter $Inf_2 = (1 - Prob_2)$ (which is also called *first normalisation*), $Prob_2$ is defined as the probability of observing another occurrence of the term in the document, given that we have seen already tf occurrences. For this purpose, Amati regards two approaches:

L Based on Laplace's law of succession, he gets

$$Inf_2 = \frac{1}{tf_2 + 1} \quad (4)$$

B Regarding the ratio of two Bernoulli processes yields

$$Inf_2 = \frac{F+1}{n \cdot (tf_2 + 1)} \quad (5)$$

These parameters do not yet consider the length of the document to be indexed. For the relationship between document length and term frequency, we apply the following formula:

$$\rho(l) = c \cdot l^{\beta} \quad (6)$$

where l is the document length, $\rho(l)$ is the density function of the term frequency in the document, c is a constant and β is a parameter to be chosen.
In order to consider length normalisation, Amati maps the tf frequency onto a normalised frequency tfn computed in the following way: Let $l(d)$ denote the length of document d and avl is the average length of a document in the collection. Then tfn is defined as:

$$tfn = \int_{l(d)}^{l(d)+avl} \rho(l) dl \quad (7)$$

Thus, the normalised term frequency tfn is computed by assuming that there would be a document of average length appended to the actual document, and that we estimate the number of term occurrences within this hypothetical document (based on the term density function $\rho(l)$).

For considering these normalisations, Amati sets $tf_1 = tf_2 = tfn$ in formulas 2–5 and then computes the term weight according to eqn 1.
For retrieval, the query term weight qtf is set to the number of occurrences of the term in the query. Then a linear retrieval function is applied:

$$R(q,d) = \sum_{t \in q} qtf \cdot Inf_2(tf_2) \cdot Inf_1(tf_1) \quad (8)$$

In [Amati & Rijsbergen 02], DFR evaluation results for different parts of the TREC collection are reported. In many cases, DFR variants give better results than the BM25 formula[1], and in some cases even yield the best overall results. Thus the DFR approach offers both a solid theoretical foundation an a high retrieval quality.

2.2 Applying divergence from randomness to XML documents

2.2.1 *Direct application of Amati's model*

In Section 2.1, we have described the basic model along with a subset of the weighting functions proposed by Amati. Given that we have two different formulas for computing Inf_1 as well as two different ways for computing Inf_2, we have four basic weighting formulas which we are considering in the following.
In a first round of experiments, we tried to apply Amati's model without major changes. However, whereas Amati's model was defined for a set of atomic documents, CO retrieval is searching for so-called *index nodes*, i.e. XML elements that are meaningful units for being returned as retrieval answer.
As starting point, we assumed that the complete collection consists of the concatenation of all XML documents. When we regard a single index node, we assume that the complete collection consists of documents having the same size as our current node. Let L denote the total length of the collection and $l(d)$ the length of the current node (as above), then we compute the number of hypothetical documents as $N = L/l(d)$. Since we assume that all documents are of equal length, no document length normalisation (eqn. 7) is necessary in this case; instead, we have an implicit consideration of document length via modifying N, which, in turn, affects λ in eqn. (2) and (3).
Table 1 shows the experimental results. The first two result columns list the average precision values for this setting when applying the four different weighting functions. We suspect that the poor performance is due to the fact that the weights derived from different doc-

[1] In [Amati & Rijsbergen 02], it is shown that BM25 actually is an approximation of one of the DFR formulas.

Table 3: Results from 2nd normalisation with two other values for β

	$\beta = -0.75$		$\beta = -0.80$	
	B Norm.	L Norm.	B Norm.	L Norm.
Binomial	0.0799	0.1026	0.0768	0.1005
Bose-Einstein	0.0453	0.0653	0.0448	0.0654

Table 4: Average precision for the Bose-Einstein L Norm combination with various values of α

α	2	4	9	16	20	32	64	96	104	116	128
prec.	0.0726	0.0865	0.0989	0.1059	0.1077	0.1083	0.1089	0.1094	0.1087	0.1081	0.1077

ument lengths are not comparable, i.e. that our 'implicit' document length normalisation via modifying the hypothetical total number of documents N is not feasible.

As an alternative method, we computed the average size of an index node. The two last columns in table 1 show a much better retrieval quality for this case.

In the subsequent experiments, we focused on the second approach. By referring to the average size of an index node we were also able to apply document length normalisation according to Equation 6. Table 2 shows the corresponding results for $\beta = 0$ and $\beta = -1$. The results show that length normalisation with $\beta = -1$ improves retrieval quality in most cases. These results were also in conformance with Amati's findings that $\beta = -1$ gives better results than $\beta = 0$.

Subsequently we tried some other values for β. Table 3 shows the corresponding results for $\beta = -0.75$ and $\beta = -0.80$, with which we got better results.

Overall, using a fixed average document length, and length normalisation, gave better results than those achieved in the first round. However, the resulting retrieval quality was still lower than that of the augmentation approach (see table 1). Thus, in order to arrive at a better retrieval quality, we investigated other ways than straightforward application of Amati's model.

2.2.2 *Considering the hierarchical structure of XML documents*

In order to consider the hierarchical structure of XML documents, we investigated different ways for incorporating structural parameters within the weighting formula. Regarding the basic ideas, as described in Section 2.1, the most appropriate way seemed to be the modification of the Inf_2 parameter, which refers to the 'elite' set. Therefore, we computed Inf_1 as above, by performing document length normalisation with respect to the average size of an index node.

For computing Inf_2, we also applied document length normalisation first, thus yielding a normalised term frequency tfn. Then we investigated several methods for 'normalising' this factor with respect to the hierarchical document structure; we call this process *third normalisation*. For this purpose, we introduced an additional parameter $h(d)$ specifying the height (or level) of an index node relative to the root node (which has $h = 1$).

Using the level information, we first tried several heuristic formulas like $tf_2 = tfn \cdot h(d)^\alpha$ and $tf_2 = tfn \cdot h(d)^{-\alpha}$, which, however, did not result in any improvements. Finally, we came up with the following formula:

$$tf_2 = tfn \cdot (h(d)/\alpha) \qquad (9)$$

Here α is a constant to be chosen, for which we tried several values. However, the experiments showed that the choice of α is not critical. This weighting formula gives higher weights to terms occurring in deeper elements of the document tree. This way, we try to achieve the INEX CO goal of retrieving the most specific elements answering the query.

Table 4 shows the results for the combination of Bose-Einstein and Laplace normalisation, for which we got significant improvements. This variant also gave better results in Amati's experiments. In further experiments not listed here we tried to combine 3rd normalisation with the binomial model; however, this resulted in a decrease of retrieval quality.

3. CONTENT-AND STRUCTURE(CAS) TOPICS

The query language XIRQL of our retrieval system HyREX is very similar to the INEX CAS topic specification. However, our experience from INEX 2002 has shown that a 'literal' interpretation of the CAS queries does not lead to good retrieval results. Thus, we were looking for 'vague' interpretations of the INEX topics. Since XIRQL has a high expressiveness, we did not want to change the semantics of XIRQL (by introducing vague interpretations of the different language elements). Instead, we focused on the transformation from the INEX topic specification into XIRQL.

XIRQL is an extension of XPath [Clark & DeRose 99] by IR concepts. We assume that XML document elements have specific data types, like e.g. person names, dates, technical measurement values and names of geographic regions. For each data type, there are specific search predicates, most of which are vague (e.g. phonetic similarity of names, approximate matching of dates and closeness of geographic regions). In addition to Boolean connectors, there also is a weighted sum operator for computing the scalar product between query and document term weights.

The general format of a of an INEX query is

//TE[filter] or

//CE[filter]//TE[filter]

Where TE stands for Target Element and CE stands for Context Element.

In XIRQL, single query conditions can be combined in the following way:

Conjunctions(and) Filter conditions(conditions within [..]) can be combined by the and operator

Disjunctions(or) Filter conditions can be combined by the or operator.

Weighted Sum (wsum) and Precedence Weighted sum notation can be used to indicate the importance of a query term, e.g.

```
//article[wsum(
0.7,.//atl//#PCDATA $stem$ "image",
0.3,.//atl//#PCDATA $stem$ "retrieval"
)]
```

Phrases Since HyREX has no specific phrase operator (yet), we represented phrases as conjunctions of the single words, e.g.

```
//article[wsum(
1.0,.//atl//#PCDATA [. $stem$ "image"
$and$ . $stem$ "retrieval"],
1.0,. $stem$ "colour")]
```

3.1 Experimentation

In order to search for better transformations from INEX CAS topics into XIRQL, we performed a number of experiments using the INEX 2002 topics (which we transformed into the 2003 format). For generating our XIRQL queries, we used only titles and keywords of the topics. In the following we briefly characterise the different kinds of transformations investigated. We illustrate each method by showing the resulting XIRQL expression for the following INEX topic(articles about image retrieval methods based on colour, contour, shape, texture and semantics):

```
//article[about(.//atl,'image retrieval'
) and about(.,'image retrieval colour
shape texture')]
```

3.1.1 CAS-I

The first transformation assumes a very strict interpretation of the INEX query. Except for the query terms, we always assume a conjunction of conditions:

1. Only query title is used.
2. Phrases are represented using conjunctions.
3. Query terms are represented using disjunctions
4. Mandatory ('+' prefixed) terms are handled by conjunctions

```
//article[(.//atl//#PCDATA[
. $stem$ "image" $and$
. $stem$ "retrieval"]) $and$
(.//#PCDATA[ . $stem$ "image"] $or$
 .//#PCDATA[ . $stem$ "retrieval"]
 $or$ .//#PCDATA[ . $stem$ "colour"]
 $or$ .//#PCDATA[ . $stem$ "shape"]
 $or$ .//#PCDATA[ . $stem$ "texture"]
 )]
```

3.1.2 CAS-II

Here we tried a vague interpretation of the query, by combining the different conditions via weighted sum, and mandatory terms just get higher weights.

1. Only query title is used.
2. Phrases are represented using conjunctions.
3. Terms are represented using weighted sum notation and assigned weight 1.
4. Mandatory terms are assigned higher weights.

```
/article[ wsum(1.0,.//atl//#PCDATA[
. $stem$ "image" $and$
. $stem$ "retrieval"],
 1.0, ... $stem$ "image",
 1.0, ... $stem$ "retrieval",
 1.0, ... $stem$ "colour",
 1.0, ... $stem$ "shape",
 1.0, ... $stem$ "texture")]
```

3.1.3 CAS-III

This variant is is a combination of CAS-I and CAS-II:

1. Only query title is used.
2. Phrases are represented using conjunctions.
3. Terms are represented using weighted sum notation and XPath notations. These two notations are joined with or operator.
4. '+' prefixed terms are assigned higher weight 5 and also represented as phrases.

```
//article[(.//atl//#PCDATA[
. $stem$ "image" $and$
. $stem$ "retrieval"])
$and$
(.//#PCDATA[ . $stem$ "image"] $or$
 .//#PCDATA[ . $stem$ "retrieval"] $or$
 .//#PCDATA[ . $stem$ "colour"] $or$
 .//#PCDATA[ . $stem$ "shape"] $or$
 .//#PCDATA[ . $stem$ "texture"]) $or$
 wsum(1.0,.//atl//#PCDATA $stem$ "image",
  1.0,.//atl//#PCDATA $stem$ "retrieval",
  1.0, ... $stem$ "image",
  1.0, ... $stem$ "retrieval",
  1.0, ... $stem$ "colour",
  1.0, ... $stem$ "shape",
  1.0, ... $stem$ "texture")]
```

3.1.4 CAS-IV

This variant is similar to CAS-I, but considers terms from both the title and the keywords.

1. Query titles and keywords are used. Keywords are considered in case there are less than 3 query terms in the title.
2. Phrases are represented using conjunctions.
3. Terms are represented using disjunctions
4. '+' prefixed terms are handled as phrases.

```
//article[  (  .//atl//#PCDATA[
. $stem$ "image" $and$
. $stem$ "retrieval"]) $and$
 (.//#PCDATA[ . $stem$ "image"] $or$
  .//#PCDATA[ . $stem$ "retrieval"]
  $or$ .//#PCDATA[ . $stem$ "colour"]
  $or$ .//#PCDATA[ . $stem$ "shape"]
  $or$ .//#PCDATA[ . $stem$ "texture"
  ])]
```

3.1.5 CAS-V

This is a more vague variant of CAS-II, where we combine even the components of a phrase via wsum.

1. Only query title is used.
2. Phrases are also handled as terms and assigned weight 1.0.
3. Terms are combined by wsum operator.
4. Higher weight (5) is assigned to terms prefixed with '+'.

Table 5: Query variations summary

	query part	notation	terms	phrases	+prefixed terms
CAS-I	title	XPath	or	and	and
CAS-II	title	wsum	weight 1.0	and	weight 5.0
CAS-III	title	XPath & wsum	or & weight 1.0	and	and & weight 5.0
CAS-IV	title & keywords	XPath	or	and	and
CAS-V	title	wsum	weight 1.0	weight 1.0	weight 5.0

Table 6: Results:Experimentation with INEX 2002 CAS topics

Query Variation	Average Precision			
	ignore empty		consider empty	
	strict	generalised	strict	generalised
CAS-I	0.2640	0.2338	0.1692	0.1508
CAS-II	0.1325	0.1215	0.0859	0.0798
CAS-III	0.1724	0.1415	0.1045	0.0916
CAS-IV	0.1297	0.1179	0.0959	0.0877
CAS-V	0.1327	0.1077	0.0806	0.0872

```
//article[
wsum( 1.0,.//atl//#PCDATA $stem$ "image",
 1.0,.//atl//#PCDATA $stem$ "retrieval",
 1.0, ... $stem$ "image",
 1.0, ... $stem$ "retrieval",
 1.0, ... $stem$ "colour",
 1.0, ... $stem$ "shape",
 1.0, ... $stem$ "texture")
]
```

3.1.6 *Evaluation*

Using the two strict and the generalised variants of the INEX evaluation metrics [Gövert & Kazai 03], we got the results shown in table 8. Depending on the query complexity, some of the queries could not be processed by HyREX; columns headed by 'ignore empty' give performance figures where these queries are ignored, whereas 'consider empty' means that these queries are considered with zero precision. One can see that the strict interpretation CAS-I yields the best results, whereas all vague interpretations lead to a lower retrieval quality. We conclude that — at least for the strict interpretation of the CAS queries — vague interpretations of the query logic by replacing conjunctions with disjunctions or weighted sums do improve results, they lead to a lower retrieval quality.

4. INEX 2003 SUBMISSIONS & RESULTS

Our CO submissions in INEX 2003 include:

- factor 0.5
- factor 0.2
- difra_sequential

The first two submissions use the "augmentation" method (the same as in our 2002 INEX submission) with 0.5 and 0.2 as "augmentation facto", respectively. The third submission is based on the "DFR" method. Here, we chose the best configuration according to our experiments results, i.e. Bose-Einstein and L Normalisation with the parameters $\alpha = 96$ and $\beta = -0.80$.
Table 7 lists the evaluation results of our submissions, based on different metrics, in INEX 2003. The results show that the latter two submissions both performed well, with the augmentation method still slightly better than the DFR approach.
For the CAS topics, two subtasks were defined in INEX: strict CAS(SCAS) and vague CAS (VCAS). SCAS enforces the strict interpretation of CAS topics while in case of VCAS, query conditions can be treated vaguely. For the latter also a list of equivalent tags was defined.; as long as the retrieved component is structurally similar to the user's interest (target element), it is considered to be correct.
With regard to these two subtasks, we submitted three runs based on the query transformation CAS-I ... III for the SCAS task as SCAS03-I ... III, while the other two transformations CAS-IV and CAS-V were used as VCAS submissions VCAS03-I and VCAS03-II, respectively. Since our system could not process all transformations with the alias list of element names (leading to the corresponding disjunction of structural conditions), the alias list was not applied for two of the submissions:

- SCAS03-I-alias
- SCAS03-II-alias
- SCAS03-III-noalias
- VCAS03-I-alias
- VCAS03-II-alias
- VCAS03-I-noalias

Table 8 shows the evaluation results of our submissions in INEX 2003. The results confirm the outcome of our own experiments. SCAS03-I-alias is the best of our submitted runs and performed quite well(ranked at 5th and 9th out of 38 for strict and generalised quantisations respectively) in comparison to other approaches.

5. CONCLUSIONS

The results from INEX 2003 show that HyREX yields good retrieval performance both for CO and CAS queries. For the CO queries, our extension to the basic DFR approach takes into account only the level of a retrieved element (via third normalisation). However, there are numerous other parameters that could considered, such as e.g. element names, element-specific node length, or

Table 7: Average precision for our CO submissions in INEX 2003

Submission	Average Precision					
	inex_eval		inex_eval_ng			
			consider overlap		ignore overlap	
	strict	generalised	strict	generalised	strict	generalised
factor 0.5	0.0703	0.0475	0.1025	0.0623	0.0806	0.0590
factor 0.2	0.1010	0.0702	0.1409	0.0903	0.1219	0.0964
difra_sequential	0.0906	0.0688	0.1354	0.0774	0.1217	0.0920

Table 8: Average precision for our CAS submissions in INEX 2003

Submission	Average Precision & Ranking			
	strict		generalised	
	avg. precision	ranking	avg.precision	ranking
SCAS-I-alias	0.2594	5	0.2037	9
SCAS-II-alias	0.2213	18	0.1744	18
SCAS-III-noalias	0.2034	19	0.1707	18

element specific prior probabilities. By investigating the influence of these factors, we will continue our work on the DFR approach towards a full-fledged language model for XML retrieval. On the CAS side, besides dealing with some weaknesses of the current implementation, we will investigate further methods for 'vague' interpretations of this type of queries, especially with regard to structural conditions.

6. REFERENCES

Abolhassani, M.; Fuhr, N. (2004). Applying the Divergence From Randomness Approach for Content-Only Search in XML Documents. In: *26th European Conference on Information Retrieval Research (ECIR 2004)*. Springer, Heidelberg et al.

Amati, G.; Rijsbergen, C. (2002). Probabilistic models of information retrieval based on measuring the divergence from randomness. *ACM Transactions on Information Systems (TOIS) 20(4)*, pages 357–389.

Clark, J.; DeRose, S. (1999). *XML Path Language (XPath) Version 1.0*. Technical report, World Wide Web Consortium. `http://www.w3.org/TR/xpath20/`.

Fuhr, N.; Großjohann, K. (2001). XIRQL: A Query Language for Information Retrieval in XML Documents. In: Croft, W.; Harper, D.; Kraft, D.; Zobel, J. (eds.): *Proceedings of the 24th Annual International Conference on Research and development in Information Retrieval*, pages 172–180. ACM, New York.

Fuhr, N.; Großjohann, K. (2004). XIRQL: An XML Query Language Based on Information Retrieval Concepts. (accepted for publication).

Fuhr, N.; Gövert, N.; Großjohann, K. (2002). HyREX: Hyper-media Retrieval Engine for XML. In: Järvelin, K.; Beaulieu, M.; Baeza-Yates, R.; Myaeng, S. H. (eds.): *Proceedings of the 25th Annual International Conference on Research and Development in Information Retrieval*, page 449. ACM, New York. Demonstration.

Gövert, N.; Kazai, G. (2003). Overview of the INitiative for the Evaluation of XML retrieval (INEX) 2002. In: Fuhr, N.; Gövert, N.; Kazai, G.; Lalmas, M. (eds.): *INitiative for the Evaluation of XML Retrieval (INEX). Proceedings of the First INEX Workshop. Dagstuhl, Germany, December 8–11, 2002*, ERCIM Workshop Proceedings, pages 1–17. ERCIM, Sophia Antipolis, France. `http://www.ercim.org/publication/ws-proceedings/INEX2002.pdf`.

Robertson, S. E.; Walker, S.; Jones, S.; Hancock-Beaulieu, M. M. (1995). Okapi at TREC-3. In: *Proceedings of the 3rd Text Retrieval Converence (TREC-3)*, pages 109–126. NTIS, Springfield, Virginia, USA.