Océ at TREC 2003

Pascha Iljin, Roel Brand, Samuel Driessen, Jakob Klok

Océ-Technologies B.V.
P.O. Box 101
5900 MA Venlo
The Netherlands
{pi, rkbr, sjdr, klok}@oce.nl

Abstract
This report describes the work done at Océ Research for the TREC 2003. This first participation consists of ad hoc experiments for the Robust track. We used the BM25 model and our new probabilistic model to rank documents. Knowledge Concepts' Content Enabler semantic network was used for stemming and query expansion. Our main goal was to compare the BM25 model and the probabilistic model implemented with and/or without query expansion. The developed generic probabilistic model does not use global statistics of a document collection to rank documents. The relevance of the document to a given query is calculated using term frequencies of the query terms in the document and the length of the document. Furthermore, some theoretical research has been done. We have constructed a model that uses relevance judgements of previous years. However, we did not implement it due to the time constraints.

1 Introduction
This is our first participation in the Text REtrieval Conference. We aimed to compare the models we constructed during the last two years. We decided to participate in the Robust track because it allows to evaluate IR systems given a set of topics and relevance judgements of previous years. That is exactly what we did for an internal research using the CLEF Dutch collection. Furthermore, the Robust track is oriented towards the actual practical situation in information retrieval (i.e. good results are expected for every query). Due to the time restrictions we did not manage to retrain our theoretical model for the TREC's collection of documents and queries in English.

2 Description of runs
The description of the submitted runs is presented in the table below:

Run number	Ranking model	Topic's tags used	Expansion of query terms
1	BM25	Title+Description	yes
2	BM25	Title+Description	no
3	BM25	Description	no
4	probabilistic	Title+Description	yes
5	probabilistic	Title+Description	no

The information about the query construction is presented in Section 3. The models will be described in Section 4.

3 Methods

3.1 Query
A query is constructed automatically from the *title* and *description* (in one of the experiments just the description is used, as required by the track guidelines) by splitting on non-alphanumerical characters to obtain terms. All single characters are removed afterwards. Furthermore, all remaining terms are converted to lower case. For the query expansion, the morphological collapse (dictionary based stemming) of Knowledge Concepts' Content Enabler semantic network is used to obtain root forms of query terms. The root forms are then expanded with the semantic network. The morphological variants of the root form (such as plural form, etc.) are added to the query.

Expansion of query terms
All query terms are morphologically expanded using Knowledge Concepts' Content Enabler semantic network.

Related terms and synonym expansion
Research was done on using related terms and synonyms. We found that Knowledge Concepts' Content Enabler is not good enough to create related terms and synonyms for our models. A measure of 'similarity' between two terms is needed in order to rank the proposed list of related terms and synonyms. Only terms that are very 'similar' in their meaning to a query term should be added to the expanded query.

Query consisting of *topic* + *description* tags
For the experiments with the queries composed of the *topic* and *description* tags, the terms from these two tags have been put together without duplicate removal. We assumed that if a term in the query is present more than one time, it is considered to be a more important term than if it occurs once.

3.2 Indexing
The index was built by splitting documents on non-alphanumerical characters. Single characters were removed from the index. Stop words were left in the index because it is very difficult to construct a universal set of stop words. If such a set is based on the frequencies within a document collection, it is highly probable that the set of stop words will not be the same for two different document collections. In case it is based on human decisions, a number of important terms from the document collection and/or query will be removed. For example, consider the terms 'new' and 'year' as stop words (they are used in this role quite often). After removing these terms from the document collection and from the queries, it becomes difficult to find a set of relevant documents for the query '*A New Year tree*'. In order to show that stop word removal is not always beneficial, consider the query '*Who said "To be or not to be?"*'. In this case *all* terms from the query could be defined as stop words. Nevertheless, the stop words should be treated different than other terms. Therefore, we weight them down. This year the following stop word lists were used:
- Search Engine World (http://www.searchengineworld.com/spy/stopwords.htm)
- Institut interfacultaire d'informatique, University of Neuchatel (http://www.unine.ch/Info/clef/)

4 Ranking models

4.1 BM25 model
The general description of the BM25 model is as follows:
Let q_i be a query term in query q
Let $q_{i,0}, q_{i,1}, \ldots, q_{i,n}$ be the expansion of q_i in which $q_{i,0} = q_i$
Let tf ($q_{i,j}$, d) be the term frequency of expansion term $q_{i,j}$

We now calculate the document and term frequency of q_i as follows:

$$tf(q_i, d) = \sum_j tf(q_{i,j}, d) \qquad (1)$$

$$df(q_i) = \left| \bigcup_j \text{set of documents in which } q_{i,j} \text{ occurs} \right|$$

Then for a document *d*, and query *q*, the score is calculated as:

$$\text{Rel}(d, q) = \sum_{q_i \in q} \frac{\log(N) - \log(df(q_i)) \cdot tf(q_i, d) \cdot (k_1 + 1)}{k_1 \cdot ((1-b) + (b \cdot ndl(d))) + tf(q_i, d)}, \qquad (2)$$

in which ndl(*d*) is the length of the document *d*, divided by the average document length.

This model was used for the CLEF 2002 runs and has been described in [1]. Last year we observed that the performance of the BM25 ranking algorithm depends greatly on the choice of the values of the parameters *k1* and *b*.

The estimation of those values for the optimal performance is only possible when the document collection, the set of queries and the set of relevance judgements are all available beforehand. Hence, the Robust track for the old queries is a suitable training set.

4.2 Probabilistic model

The probabilistic model has been selected as the result of theoretical research conducted in 2002 [2]. It contains some innovations with respect to the standard probabilistic approach. The urn model (i.e. balls in an urn = terms in a document) was selected as a basis for the probabilistic model.

We calculate the degree of relevancy without using collection statistics (e.g. document frequency). The sparse data problem is commonly solved using the linear interpolation method or other smoothing techniques that are based on collection statistics. Robertson showed that "relevance of a document to a request should not depend on the other documents in the collection" in order to guarantee "optimality of ranking by the probability of relevance" [3]. Therefore, the selection of a complete document collection as a smoothing element is not strongly motivated and not even supposed to exist according to the basic principle of the probabilistic approach in information retrieval. We found experimentally that under certain distributions of terms over documents in the document collection, the linear interpolation approach will give illogical ranking results. A standard solution to the sparse data problem is to assign non-zero values for query terms that do not exist in a document. The most natural and easy way to solve the sparse data problem is to assign a constant positive value α to the terms that do not exist in the document. We named this **'the α-method'**.

For the query without term expansion:

$$\text{Rel (d, Q)} = \prod_{q_i \in Q} \left[\frac{1}{2} \cdot \left(\frac{tf(q_i, d)}{L_d} + \alpha \right) \right], \qquad (3)$$

where L_d is the length (not normalised) of the document *d*,
α should be less than [the length of the longest document in the document collection]$^{-1}$. This guarantees coordination level ranking.

4.3 Statistical model (theoretical results)

In 2002 we aimed to implement a set of clues (that we defined) in a '*mathematically correct*' model, i.e. a model without internal contradictions or violations of axioms. Examples of clues are:

- presence of terms in the document that are synonyms of the terms in the query;
- importance of a topic's tag;
- part of speech of the query terms;
- query terms of certain document frequency;
- presence of proper nouns in the query;
- length of a document.

We found that a set of defined clues could not be entirely incorporated in the currently known information retrieval models while maintaining mathematical correctness. However, we have succeeded to construct a statistical approach that allows incorporation of these clues. For each clue, a value expressing its expected '*significance*' is calculated. *Significance* values are based on relevance judgements from previous years for (document, topic) pairs.

For every clue we test whether its incorporation makes a statistically significant contribution to the overall performance of an information retrieval system.

Let us select a *clue*[1] to investigate its contribution to the improvement of the performance. The following procedure is carried out for the whole set of queries. Let us consider a query *q*:

- From *q* we determine those components that can be tested for contribution of the *clue* with respect to the total performance of an information retrieval system.

[1] Taking two or more clues simultaneously is very complex.

Let us denote by $Comp_c(q, clue)$ the c^{th} component in the query q that is tested, where $c = \overline{1, C(q, clue)}$, and $C(q, clue)$ is the total number of components from the query q that can be tested on the *clue*.

Example 1
In case the *clue* is a 'presence of query terms in a document', all query terms are components.
Example 2
In case the *clue* is a 'noun', the components from the query 'Crocodiles living in the lake' are 'crocodiles', and 'lake'.

- The following notation will be used:

$|R(J,q)|$ - the number of documents from document collection (*Dc*) that have got values *relevant* from the relevance judgements for the query q.

$|I(J,q)|$ - the number of documents from *Dc* that have got values *irrelevant* from the relevance judgements for the query q.

$|R_{Comp_c(q,clue)}|$ - the number of documents from *Dc* that have got values *relevant* from the relevance judgements for the query q and that contain $Comp_c(q, clue)$.

$|I_{Comp_c(q,clue)}|$ - the number of documents from *Dc* that have got values *irrelevant* from the relevance judgements for the query q and that contain $Comp_c(q, clue)$.

- Calculate for every component $Comp_c(q, clue)$:

$$R_c(clue,q) = \frac{|R_{Comp_c(q,clue)}|}{|R(J,q)|} \quad (4)$$

$$I_c(clue,q) = \frac{|I_{Comp_c(q,clue)}|}{|I(J,q)|} \quad (5)$$

The pair $(R_c(clue,q), I_c(clue,q))$ indicates how often a component $Comp_c(q, clue)$ occurs in relevant and irrelevant documents respectively. In case $R_c(clue,q) > I_c(clue,q)$, the component $Comp_c(q, clue)$ occurs more often in relevant documents than in irrelevant ones.

After $(R_c(clue,q), I_c(clue,q))$ is calculated for each component c of each query q, a set of pairs $\{(R_1(clue), I_1(clue)), (R_2(clue), I_2(clue)),\ldots, (R_t(clue), I_t(clue))\}$ is obtained, where $t = \sum_{q=1}^{|Q|} C(q, clue)$ is the number of all components for *clue* from all $|Q|$ queries in the test collection.

In case $\sum_{\{R_i(clue)>I_i(clue)\}}^{t} 1 > \sum_{\{R_i(clue)<I_i(clue)\}}^{t} 1$, one can state that after incorporating the *clue*, the components of q appear more often in relevant documents than they appear in irrelevant ones. This statement implies that the incorporated clue is expected to improve the performance of the information retrieval system.

In order to decide if a clue may improve the performance of the system, the set of pairs $\{(R_1(clue), I_1(clue)), (R_2(clue), I_2(clue)),\ldots, (R_t(clue), I_t(clue))\}$ should be statistically investigated. The statistical method called the Sign Test is used in order to compare two sets of pairs. It is the only method that can be used for our purpose.

The Sign Test is used to test the hypothesis that there is "no difference" between the two probability distributions (in our case, R(*clue*) and I(*clue*)). For the statistical model it tests whether the presence of the *clue* has influence on the distribution of the query components in relevant and irrelevant documents.

The theory of the Sign Test requires:
1. The pairs to be mutually independent.
2. Both R_i(*clue*) and I_i(*clue*) should have continuous probability distributions.

Because of the assumed mutual independence between queries, mutual independence between query terms, and mutual independence between terms in documents, pairs (R_1(*clue*), I_1(*clue*)) are mutually independent (point 1). A continuous distribution is defined as a distribution for which the variables may take on a continuous range of values. In the considered case, the values of both R_i(*clue*) and I_i(*clue*) take any value from the closed interval [0,1], and so their distributions are continuous (point 2). Hence, the necessary conditions for the Sign Test hold.

The hypothesis implies that given a pair of measurements (R_i(*clue*), I_i(*clue*)), both R_i(*clue*) and I_i(*clue*) are equally likely to be larger than the other. The zero hypothesis H_0: $P[R_i(clue) > I_i(clue)] = P[R_i(clue) < I_i(clue)] = 0.5$ is tested for every $i=\overline{1,t}$. Applying the one-sided Sign Test means that rejecting H_0, we accept the alternative hypothesis H_1: $P[R_i(clue) > I_i(clue)] > 0.5$. A one-sided 95% confidence interval is taken to test the H_0 hypothesis. If H_0 is rejected, the incorporation of the *clue* is expected to improve the performance of the information retrieval system.

> Remark
> Using the Sign Test described for a certain clue, we conclude whether its incorporation into an information retrieval system can improve the performance. This conclusion is based on theoretical expectations only.

Two criteria are defined to estimate the possible contribution of a clue to a system from a practical point of view.

In case there are *t* components for all the queries, $\forall i = \overline{1,t}$ calculate for *clue*

i) #(R(*clue*)) – the number of components for which $R_i(clue) > I_i(clue)$
ii) #(I(*clue*)) – the number of components for which $R_i(clue) < I_i(clue)$

According to the theoretical issues of the Sign Test, one has to ignore the statistics of the components for which $R_i(clue) = I_i(clue)$. Thus, when a component of a certain clue is found in both relevant and irrelevant documents, and the relative frequency of $R_i(clue) = I_i(clue)$, this is neither good nor bad. Such an observation should not influence the total statistics.

However, the other theoretical issue will not be taken into account. According to the theory of the Sign Test, when one observes more than one component with the same values of R_i(*clue*) and I_i(*clue*), all but one component should be ignored too. However, this claim cannot be valid in the area of linguistics due to the following reasons:

1. The influence of each component on the clue has to be calculated. Even in the case the same statistics are obtained for different terms, all terms will make a contribution to the performance of the system. So, every component will be an extra observation for a clue.

2. If a term is used in more than one query, it has multiple influences on the performance. For each query different statistics should be obtained. Hence, each component should be considered separately for every query.

3. In case the same component is used more than one time in a query, it is considered multiple times (according to the assumption described in 'Query consisting of *topic* + *description* tags', see Section 3.1).

To estimate the significance of a certain clue, the ratio $\frac{\#(R(clue))}{\#(I(clue))}$ is calculated. The larger this ratio, the higher the significance is. After calculating these ratios for all the clues, they can be ranked in a decreasing order, where the top value will correspond with the most significant clue.

- *Not all clues have the same contribution to the ranking function.*

The contribution of a certain clue depends on the level of improvement to the performance of an information retrieval system.

- *Not all clues should be implemented in the statistical model.*

A clue is implemented into a model if the ratio $\frac{\#(R(clue))}{\#(I(clue))}$ has a value higher than one. Only in this case one can expect that the selected clue can improve the performance of the system.

Experiments with the statistical model

We have done a number of experiments with the statistical model for the CLEF Dutch document collection, the set of queries and the relevance judgements for 2001 and 2002. Depending on their degree of significance, different statistics have been chosen to obtain better performance for two different sets of queries (using the same document collection). The proper choice of features and their 'gain' values lead to better results. We conclude that this model is strongly dependent on the data collection, queries and relevance judgements. Hence, the results for a set of new documents, new queries and new relevance judgements are difficult to predict. Due to time restrictions we did not retrain the model for the TREC Robust track. Therefore we did not submit the statistical model.

5 Numerical results

The following numerical results were obtained for the runs submitted by Océ at TREC 2003.

Old topics:

Run	Number of retrieved relevant documents	Average precision	R-precision	Number of topics with no relevant in top 10 (in %)	Area underneath MAP(X) vs. X curve for worst 12 topics
1 (BM25,TD,Exp)	2005 out of 4416	0.1245	0.1763	12.0	0.0117
2 (BM25,TD,noExp)	1903 out of 4416	0.1205	0.1714	14.0	0.0101
3 (BM25,D,noExp)	1570 out of 4416	0.0923	0.1470	24.0	0.0027
4 (Prob,TD,Exp)	1425 out of 4416	0.0749	0.1312	20.0	0.0041
5 (Prob,TD,noExp)	1418 out of 4416	0.0859	0.1363	20.0	0.0038

New topics:

Run	Number of retrieved relevant documents	Average precision	R-precision	Number of topics with no relevant in top 10 (in %)	Area underneath MAP(X) vs. X curve for worst 12 topics
1 (BM25,TD,Exp)	1419 out of 1658	0.3646	0.3567	10.0	0.0352
2 (BM25,TD,noExp)	1428 out of 1658	0.3379	0.3423	6.0	0.0406
3 (BM25,D,noExp)	1318 out of 1658	0.3049	0.3159	16.0	0.0134
4 (Prob,TD,Exp)	1241 out of 1658	0.2921	0.3066	12.0	0.0145
5 (Prob,TD,noExp)	1255 out of 1658	0.2846	0.3167	10.0	0.0180

All topics together:

Run	Number of retrieved relevant documents	Average precision	R-precision	Number of topics with no relevant in top 10 (in %)	Area underneath MAP(X) vs. X curve for worst 25 topics
1 (BM25,TD,Exp)	3424 out of 6074	0.2446	0.2665	11.0	0.0163
2 (BM25,TD,noExp)	3331 out of 6074	0.2292	0.2568	10.0	0.0168
3 (BM25,D,noExp)	2888 out of 6074	0.1986	0.2315	20.0	0.0055
4 (Prob,TD,Exp)	2666 out of 6074	0.1835	0.2189	16.0	0.0063
5 (Prob,TD,noExp)	2673 out of 6074	0.1852	0.2265	15.0	0.0066

6 Conclusions

We have compared the BM25 and our probabilistic model on the basis of mono-lingual runs for English. The BM25 model systematically outperforms the probabilistic one. This indicates that striving for mathematical correctness does not imply better retrieval performance. At the same time we have observed that the developed probabilistic model performs satisfactorily. Furthermore, we conclude that the query expansion using the Knowledge Concepts' Content Enabler semantic network does not improve the performance of the IR systems we constructed. The performance of the IR engine using the query consisting of the *description* tag only, is worse than using the *topic* and *description* tags.

7 References

[1] Roel Brand, Marvin Brünner: Océ at CLEF 2002. Lecture Notes on Computer Science, Springer-Verlag Heidelberg, 2003.

[2] Pascha Iljin: Modeling Document Relevancy Clues in Information Retrieval Systems. SAI, to appear in 2004.

[3] Djoerd Hiemstra: Using Language Models for Information Retrieval. Ph.D. Thesis, Centre for Telematics and Information Technology, University of Twente, 2001.