Effectiveness of Weighted Searching in an Operational IR Environment

Hans-Peter Frei
UBILAB
Union Bank of Switzerland
CH-8021 Zurich, Switzerland

Yonggang Qiu
Department of Computer Science
Swiss Federal Institute of Technology (ETH)
CH-8092 Zurich, Switzerland

We describe an experiment to compare the effectiveness of Boolean retrieval and weighted retrieval on a commercial data collection that contains millions of documents. The experiment was carried out via a front-end connected to an operational conventional host-oriented IR environment. In contrast to previous experiments where weighted retrieval had to be simulated on the host, our host was equipped with a built-in weighted retrieval algorithm. The results of the experiment clearly show that weighted retrieval performs significantly better than Boolean retrieval. On the other hand, no difference in the performance between manually weighted queries and automatically weighted queries could be detected.

1. Introduction

The area of Information Retrieval (IR) is gaining importance in many contexts. The number of searches carried out in commercially available IR systems has increased dramatically in recent years. In addition, the amount of information stored in commercial databases is increasing steadily. Despite the novel IR methods that have been developed in the past, the old-fashioned Boolean query languages are still predominantly used in commercial systems. There have been numerous attempts to improve Boolean searching. Extended Boolean logic, automatic Boolean query construction, and soft evaluation of Boolean queries yield some improvements. Likewise, the ranking of the output of Boolean queries has been shown to be of some value [Fox 88].

However, the main problem of Boolean searching is not its performance. For many users the main obstacle is being able to use Boolean logic effectively in order to formulate queries in the way a commercial retrieval system requires. Even trained users are unable to conduct effective Boolean searches at times [Mea 81]. Casual users are often lost when they have to connect terms with the operators AND and OR. Also, there is an intuitive way of trying to improve the result of a query by including additional search terms [Qiu 93]. In a Boolean environment, this approach is very often counter-productive.

Weighted retrieval, on the other hand, has the advantage that it can easily be understood by a casual user. In addition, the result of a weighted query usually improves when the number of search terms in the query grows [Qiu 93, Rob 86]. Moreover, it seems a definite advantage for many users to get a sorted list as output as opposed to just a set of items in an arbitrary order. This is why we decided to build ISIR [Qiu 92] as a front-end to a commercially available system.

ISIR uses the commercial database service Data-Star of RadioSuisse to which it connects automatically via Internet. It allows the user to issue queries in the form of either a Boolean expression or weighted search terms. In addition, the user may obtain assistance in finding appropriate search terms by using a thesaurus browser that is integrated in the system. This thesaurus browser supports various thesauri, such as ERIC, INSPEC, and MeSH.

In order to compare the effectiveness of Boolean retrieval and weighted retrieval, we ran an experiment with the students enrolled in an Information Retrieval course.

2. Boolean vs. Weighted Retrieval

Boolean and weighted retrieval are usually considered to be opposites [Fox 88, Fre 91c, Qiu 92, Rad 82, Rad 88, Wal 79]. Many experiments have been carried out in the past in order to compare Boolean and weighted retrieval. Most of them were laboratory tests carried out on relatively small test collections [Kee 92, Rad 82, Won 89] unlike the Cirt experiment [Rob 90] that is particularly relevant to what we describe in this paper.

First of all, Cirt is a front-end system connected to an operational IR host with real users, queries, databases, hosts, networks, etc. The system used for the research described in this paper was designed similarly and even the same host was used, namely Data-Star. Secondly, the results reported from the Cirt experiment were not really enthusiastic. This is in sharp contrast to our experiment that showed a significant improvement in weighted retrieval over Boolean retrieval. Although the authors claim in [Rob 90] that 'Weighted retrieval is capable of achieving results comparable to those obtained with Boolean searching,' the attentive reader notices that the authors actually expected quite a bit more.

The main reason for the mediocre success of Cirt was the inability of the host to execute weighted queries. Rather, weighted queries had to be converted into sequences of Boolean queries that were sent to the host and evaluated as if they were issued by a normal Data-Star user. We experienced the same disappointment when we did a similar experiment a year earlier. It seems to be difficult to simulate weighted queries in a Boolean environment especially when some of the basic data like term frequency is not available.

In addition, it is well-known that one of the strengths of weighted retrieval is caused by the usually large number of search terms employed. When using Cirt, the users used only a few search terms because of the time it took to process queries. Therefore, the actual potential of the weighted technique could never really surface.

We profited from the fact that a real weighted retrieval algorithm has been built directly into the Data-Star system in the meantime. This algorithm accepts a weighted query q represented by a vector $q = (a_1, a_2, ..., a_m)$. Each document d_j in the database is thought to be represented by a vector $d_j = (b_{1j}, b_{2j}, ..., b_{mj})$. Here, the a_i's and b_{ij}'s imply the weight of the index terms (t_is) in the query q and in the document d_j, respectively. The value m signifies the number of terms in the database. The retrieval function used by the weighted retrieval algorithm is the scalar vector product:

$$RSV(q, d_j) = \sum_{i=1}^{m} a_i \cdot b_{ij}$$

Note that the documents stored in the Data-Star databases are indexed conventionally. Hence, an index term is either assigned to a document or not. In other words, the term weights are either 1 or 0. For the purpose of weighted retrieval, weights are automatically assigned to the document terms. The weighted retrieval algorithm implemented in Data-Star supports two weighting schemes in addition to the original "weights":

1. $$b_{ij} = \begin{cases} 1 & \text{if the term } t_i \text{ is present in the document } d_j \\ 0 & \text{otherwise} \end{cases}$$

2. $$b_{ij} = tf(t_i, d_j)$$

3. $$b_{ij} = \frac{tf(t_i, d_j)}{1 + tf(t_i, d_j)}$$

where, $tf(t_i, d_j)$ is the term frequency of the term t_i in the document d_j.

Note that the weighting scheme 1 represents the original 'Boolean weighting' expressing the presence or absence of the term t_i in the document d_j.

3. The Experiment

The objective of the experiment was to compare the effectiveness of Boolean retrieval, manually weighted retrieval, and automatically weighted retrieval on a very large commercial document collection. The Data-Star INSPEC database – a major source of international information in physics, electrical and electronic engineering, computer and control theory, and related information technology – was chosen as our data collection. There were roughly 4.3 million documents in the INSPEC database at the time when the experiment was performed. This is in sharp contrast to most IR experiments that are performed on standard test collections of only a few thousand documents.

In our experiment, sample documents were handed out to the test subjects, computer science undergraduate students enrolled in an information retrieval course. The task was to find more such documents - similar to the sample documents - by using the IR system ISIR [Qiu 92] briefly described in section 1. In addition, the two retrieval techniques, Boolean and weighted, had to be used. In other words, every test subject formulated a Boolean query and a weighted query with the aim of capturing the content of the sample document in question. Subsequently, the two queries were evaluated by ISIR. With ISIR it is possible to assign the inverse document frequency as a weight for the search terms of the query. Therefore, every information need was expressed by the three different queries:

- a Boolean query,
- a manually weighted query,
- an automatically weighted query.

These three queries were run "simultaneously" against the INSPEC collection. Each query retrieved the same number r of documents, where r was equal to or less than the size of the answer set produced by the Boolean query. The answer set delivered by ISIR contained a mixture of documents retrieved by either the Boolean query or the weighted queries, or by all. The test subject was asked to judge the documents in the answer set by marking them with grades that indicated the degree of relevance with respect to the query. The retrieval methods were then evaluated by means of the usefulness measure [Fre 91a] and the precision values. (Note, that it is impossible to determine the recall values in such an environment.) Let us first describe the different retrieval methods employed.

The test subjects composed Boolean expressions of search terms (or term stems followed by a dollar sign $) by combining them with the operators "and", "or", "not", "xor", "same" (same paragraph), "with" (same sentence) and "adj"(adjacent in an order) as well as by including parentheses where appropriate. ISIR returned the usual retrieval status values for Boolean retrieval, namely:

$$RSV_b(q, d_j) = \begin{cases} 1 & \text{if } d_j \text{ satisfies the Boolean query } q \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

The retrieval function used with the weighted retrieval was the scalar vector product described in section 2.

$$RSV_w(q, d_j) = \sum_{t_i \in q} a_i \cdot b_{ij}$$

A test subject formulated a weighted query by indicating an arbitrary number of search terms t_i. In addition, the test subject specified how well each of the terms t_i described the information need by assigning a weight $w_u(t_i)$. The value of $w_u(t_i)$ was between 0 and 100 and was used as query weight a_i:

$$a_i = w_u(t_i)$$

None of the weighting schemes provided by Data-Star (see section 2) were used as document weights. Instead, we used the well-known 'term frequency-inverse document frequency' combination [Fre 91c]:

$$b_{ij} = \frac{tf(t_i, d_j)}{1 + tf(t_i, d_j)} \cdot idf(t_i)$$

where:

$tf(t_i, d_j)$ is the term frequency of t_i in d_j,

$idf(t_i) = \log(\frac{N}{df(t_i)})$ is the inverse document frequency of t_i, N is the number of documents in the database, $df(t_i)$ is the number of documents containing t_i.

The query weighted automatically was derived by using the inverse document frequency of t_i to approximate the query weight of t_i instead of the weight assigned manually. Therefore, the following retrieval functions were used for the weighted retrieval:

- manually weighted retrieval:

$$RSV_{mw}(q, d_j) = \sum_{t_i \in q} w_u(t_i) \cdot \frac{tf(t_i, d_j)}{1 + tf(t_i, d_j)} \cdot idf(t_i) \tag{2}$$

- automatically weighted retrieval:

$$RSV_{aw}(q, d_j) = \sum_{t_i \in q} idf(t_i) \cdot \frac{tf(t_i, d_j)}{1 + tf(t_i, d_j)} \cdot idf(t_i) \tag{3}$$

Let us look at the experiment in more detail:

```
BEGIN
  REPEAT
      give next sample document q to a test subject;

      REPEAT
          the test subject formulates/modifies a Boolean query q_b;
          submit q_b to Data-Star and get the size of the answer set  D_ds(q_b);
      UNTIL the size is acceptable AND the test subject is satisfied with the query q_b;
      the test subject formulates a weighted query q_mw;
      generate an automatically weighted query q_aw;

      system asks for r the number of desired documents per retrieval method;
          (*      r is equal to or less than the size of the answer set  D_ds(q_b)    *)

      get r documents from D_ds(q_b) and add them into R_b(D,q,r);
      assign the retrieval status value 1 to these r documents in R_b(D,q,r);

      get the top r ranked documents with respect to the weighted query q_mw;
      and add them into R_mw(D,q,r);
      assign the retrieval status values RSV_mw(q, d_j) to these r documents in R_mw(D,q,r);

      get the top r ranked documents with respect to the weighted query q_aw;
      and add them into R_aw(D,q,r);
      assign the retrieval status values RSV_aw(q, d_j) to these r documents in R_aw(D,q,r);

      R(D,q,r) := R_b(D,q,r) ∪ R_mw(D,q,r) ∪ R_aw(D,q,r);

      for those documents not retrieved by method M, calculate the retrieval status values
      with respect to method M;
          (*      M is Boolean, manually weighted, or automatically weighted          *)

      generate a random order of documents in R(D,q,r);
      deliver the documents of R(D,q,r) to the test subject and get the relevance information;
  UNTIL all queries are evaluated;
END.
```

After the test subject obtained the desired documents from the system, he or she had to assess the documents of the answer set by marking the documents with the grades (6, 5, 4, 3, 2, or 1)*. The grades indicate the degree of relevance of each document d with respect to the query q. The higher the grade, the more relevant the document. The scale for the marking is:

---

* These grades were chosen because the students who served as test subjects are graded with a similar scale and are hence familiar with this scheme. Any other grading scheme, such as A, B, C , ... could have been employed instead.

Meaning	Grade
extremely relevant	6
relevant	5
marginally relevant	4
marginally irrelevant	3
irrelevant	2
not relevant at all	1

4. Results of the Experiment

4.1 Number of Retrieved Documents

Some typical Boolean queries, manually weighted queries, and automatically weighted queries formulated by the test subjects are listed in the Appendix. Although the test subjects were not search professionals, it is apparent that in general the Boolean queries were formulated carefully. The reason is that our computer science students have a solid mathematical background.

As already mentioned, three retrieval methods were used and each method retrieved the same number of documents. The number of documents retrieved by one method is 185 as shown in Table 4-1. Therefore, the total number of documents retrieved by any of the three methods should be at most 3 times 185. Yet, the total number of the retrieved documents is 363, which is only 1.96 times 185. This result indicates that some documents retrieved by one method were also retrieved by some of the other methods. However, the number of documents retrieved by all the three methods is 34, which is only 9.37% of the total number of the retrieved documents.

Number of queries	30
Number of retrieved documents	363
Number of documents retrieved by one method	185
Number of documents retrieved by all the three methods	34
Average number of retrieved documents per query	12.10
Average r (average number of documents retrieved by one method per query)	6.17

Table 4-1: Statistical data

Fig. 4-1 shows the distribution of documents retrieved by the three methods. It can be seen easily that few documents retrieved by the Boolean retrieval method were retrieved by any of the weighted retrieval methods and many documents retrieved by one weighted retrieval method were also retrieved by the other weighted retrieval method.

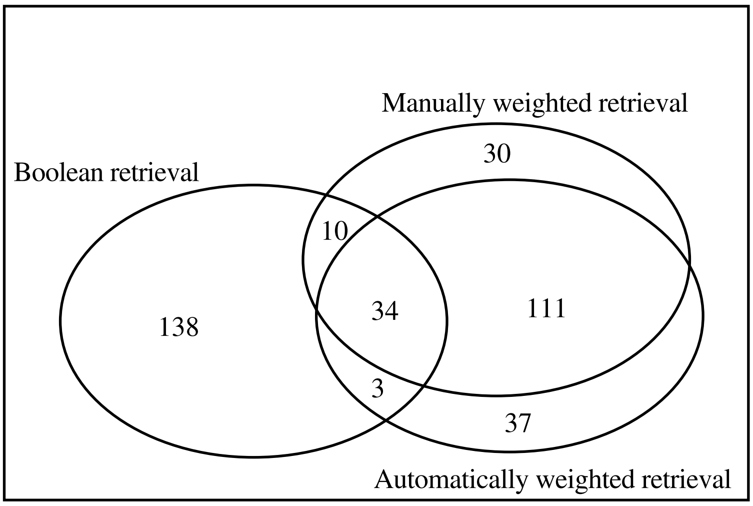

Fig 4-1: Number of documents retrieved by the three methods

Fig 4-2 shows how many relevant documents were retrieved by the three methods. A document is counted as relevant when the test subject assigned a relevance grade of 4, 5, or 6. The distribution of the documents is similar to the one shown in Fig 4-1. In particular, the set of relevant documents retrieved by the Boolean method is nearly disjoint with the sets from the weighted retrieval methods. On the other hand, the two weighted retrieval methods retrieved many common relevant documents.

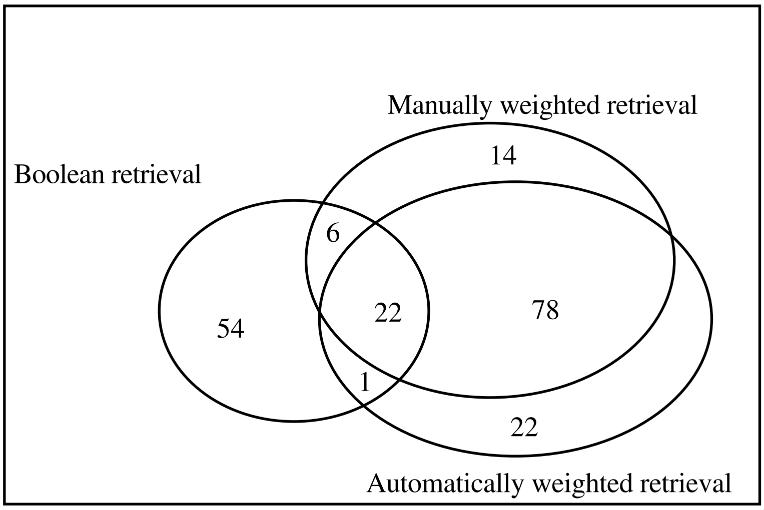

Fig 4-2: Number of relevant documents retrieved by the three methods

4.2 Precision

As mentioned before, the test subjects had to assess the documents in the answer set R(D,q,r) by marking the documents with grades (6, 5, 4, 3, 2, or 1). Documents marked with a grade of 6, 5, or 4 are regarded as relevant, and the others are regarded as non-relevant. With this data, the precision of a query can be calculated. This was done for all the queries and all the three different retrieval methods. The average precision values for the three retrieval methods are shown in Table 4-2.

Method	Boolean retrieval	Manually weighted retrieval	Automatically weighted retrieval
Average precision	0.46	0.67	0.69
Improvement over Boolean		+ 45.65 %	+ 50.0 %

Table 4-2: Average precision values

Table 4-2 shows that the precision values in our experiment were higher for the weighted retrieval methods than for the Boolean retrieval method. It is to be noted that the difference of 2.99% between the two weighted retrieval schemes is very small.

The above precision values were obtained when grade 4 was considered the first grade indicating a relevant document. However, this dividing line between relevant and non-relevant is just a matter of definition. The line could be varied so as to obtain different precision values. We define the improvement in the average precision of method B w.r.t. method A as:

$$\frac{\text{average precision of method B} - \text{average precision of method A}}{\text{average precision of method A}} * 100\ \%$$

and compare the Boolean retrieval with the weighted retrieval as a function of the first grade (threshold) that determines a relevant document. The improvement of the manually weighted retrieval w.r.t. the Boolean retrieval (Manu. Weighted vs. Boolean) and the improvement of the automatically weighted retrieval w.r.t. the Boolean retrieval (Auto. Weighted vs. Boolean) for different threshold values are shown in Fig. 4-3.

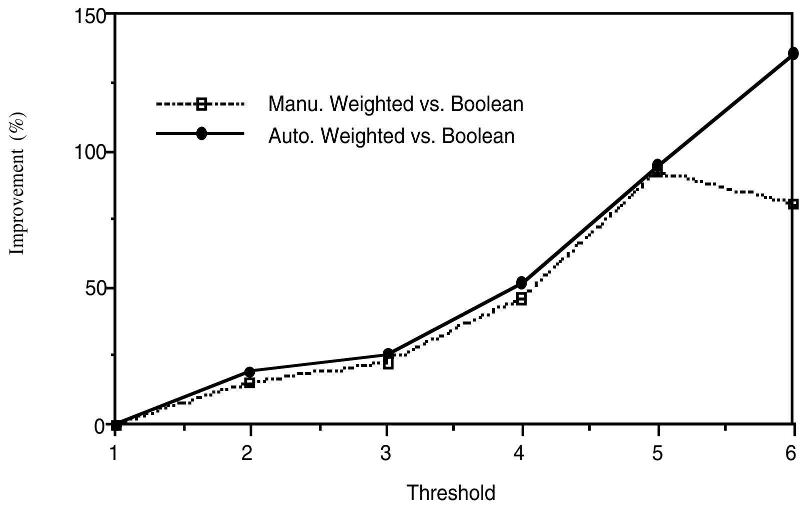

Fig. 4-3: Improvement of weighted retrieval w.r.t. Boolean retrieval

If the threshold value is set to 1, all retrieved documents are regarded as relevant documents, and the weighted retrieval and the Boolean retrieval are equally effective. When the threshold value increases, the improvement generally increases with the exception of the manually weighted retrieval method when the threshold value is set to 6. The curves shown in Fig. 4-3 indicate a surprising result: The higher the required quality of the retrieved documents, the more effective the weighted retrieval methods versus the Boolean retrieval.

4.3 Usefulness Measure

In order to evaluate the three retrieval methods using the usefulness measure [Fre 91a, Fre 91b], we first need to derive the relative relevance assessments from the test subjects' relevance information. As mentioned before, the grades Grade(d,q) indicate the degree of relevance of each document d with respect to the query q. Hence, the preferences π_p derived from relevance information are:

$$\pi_p = \{d_j <_{q_k} d_i \mid \text{Grade}(d_j,q_k) < \text{Grade}(d_i,q_k)\}$$

In the same way, we can derive the preferences π_b determined by the Boolean retrieval method, the π_{mw} determined by the manually weighted retrieval method, and the π_{aw} determined by the automatically weighted retrieval method. With the preferences π_p, π_b, π_{mw}, and π_{aw} the three methods can be evaluated using the usefulness measure. The results are shown in Table 4-3, Table 4-4 and Table 4-5.

u(b,mw)	0.662
u*(b,mw)	0.109
Pk	0.001

Table 4-3: Usefulness of manually weighted w.r.t. Boolean

u(b,aw)	0.779
u*(b,aw)	0.158
Pk	0.000

Table 4-4: Usefulness of automatically weighted w.r.t. Boolean

u(mw,aw)	0.240
u*(mw,aw)	0.017
Pk	0.147

Table 4-5: Usefulness of automatically weighted w.r.t. manually weighted

Table 4-3 and Table 4-4 indicate clearly that the weighted retrieval methods perform consistently better than the Boolean retrieval method. The corresponding error probability values Pk are very small indicating that these usefulness values are reliable. Table 4-5 indicates that the manually weighted retrieval and the automatically weighted retrieval are equally effective. The rather small usefulness values u(mw,aw) and u*(mw,aw) and the large error probability Pk shown in Table 4-5 indicate that no difference in the retrieval effectiveness between the two weighted retrieval methods can be detected for the experiment performed. In other words, the automatically weighted retrieval performs slightly better than the manually weighted retrieval, but not very consistently. A reason might be that the test subjects did not assign weights in a consistent way.

The usefulness values shown above are obtained by using multilevel relevance information. That is, the preferences used for calculating the usefulness are derived from the six level relevance information of the test subjects. Since a Boolean retrieval system divides the document collection into two sets for a query, the set of retrieved documents (answer set) and the set of not retrieved documents, it is argued that it might be inappropriate to evaluate Boolean retrieval by using multilevel relevance information. In what follows, we investigate which method performs better than the other when binary relevance information is used. In other words, only the preferences between the relevant documents and the non-relevant documents are taken into account. Documents with a grade not less than the threshold value are considered relevant, others are regarded as non-relevant. Hence, we can have the preferences π_p' derived from the binary relevance information as follows:

$$\pi_p' = \{d_j <_{q_k} d_i \mid d_i \text{ is relevant to } q_k \text{ and } d_j \text{ is non-relevant to } q_k\}$$

The usefulness values for the preferences π_p', π_b, π_{mw}, and π_{aw} with the threshold value set to 4 are shown in Table 4-6 and Table 4-7.

u(b,mw)	0.633
u*(b,mw)	0.123
Pk	0.003

Table 4-6: Usefulness of manually weighted w.r.t. Boolean by using binary relevance information

u(b,aw)	0.819
u*(b,aw)	0.217
Pk	0.000

Table 4-7: Usefulness of automatically weighted w.r.t. Boolean by using binary relevance information

The results shown in Table 4-6 and Table 4-7 are consistent to the ones shown in Table 4-3 and Table 4-4 where multilevel relevance information is used. The usefulness values shown here are even larger than the ones in Table 4-3 and Table 4-4. They indicate that the weighted retrieval methods, especially the automatically weighted retrieval method, are yet more effective than the Boolean retrieval method when the binary relevance information is used.

5. Conclusion

The experiment we describe was carried out via a front-end to a commercially available system and used INSPEC with 4.3 million of documents. The results show that the effectiveness of weighted retrieval is significantly higher than that of Boolean retrieval in this specific test environment. In addition, the experiment favored the Boolean retrieval because a limited iterative search (see section 3) was possible in contrast to the weighted retrieval. The test environment also favored the Boolean retrieval insofar as the test subjects were computer science students with a broad mathematical background; indeed, they issued rather sophisticated Boolean queries. Despite all this, the weighted retrieval came out better. The main results of the experiment are summarized:

- The weighted retrieval methods are more effective than the Boolean retrieval method when measured by both usefulness and precision.

- The better the quality of the retrieved documents need be, the more superior the weighted retrieval methods perform over the Boolean retrieval method.

- The manually weighted retrieval and the automatically weighted retrieval are equally effective.

- Only a few (relevant) documents retrieved by the Boolean retrieval were also retrieved by any one of the weighted retrieval methods, but many (relevant) documents retrieved by one weighted retrieval method were also retrieved by the other weighted retrieval method.

A follow-up experiment with real information retrieval intermediaries is under way. These search specialists are used to searching in the Data-Star chemical databases. In addition, they will formulate real queries for real users rather than test queries. Such queries are usually very complex. We hope to be able to show that weighted retrieval is just as effective as Boolean retrieval even when the Boolean queries are composed by search specialists. Considering the ease of use of the weighted retrieval, this would probably mean that many end users could use commercial IR systems without the assistance of search specialists.

However, the overlap between the answer sets produced by the Boolean method and those produced by the weighted methods is not very large. Therefore, a sensible combination of Boolean and weighted retrieval [Pai 84, Rad 82, Wal 79] may be the solution for many real IR problems.

Appendix. Query Texts

Some typical Boolean queries, manually weighted queries and automatically weighted queries formulated by the test subjects are shown in the following table.

```
Boolean query : information adj retrieval and logic and
                semantics
Manually weighted query:
 {( INFORMATION ,   20 ),
  ( RETRIEVAL ,     20 ),
  ( LOGIC ,         30 ),
  ( SEMANTICS ,     30 ) }
Automatically weighted query:
 {( INFORMATION ,   2.466788 ),
  ( RETRIEVAL ,     4.842672 ),
  ( LOGIC ,         3.920782 ),
  ( SEMANTICS ,     5.792548 ) }


Boolean query : ((replicated adj database) or (distributed adj
              database)) and serialization
Manually weighted query:
 {( REPLICATED ,    70 ),
  ( DATABASE ,      90 ),
  ( DISTRIBUTED ,   80 ),
  ( SERIALIZATION,  100 ) }
Automatically weighted query:
 {( REPLICATED ,    7.583377 ),
  ( DATABASE ,      4.081138 ),
  ( DISTRIBUTED ,   3.785400 ),
  ( SERIALIZATION,  9.895773 ) }


Boolean query : (image adj compression) and (real adj time) and
              (compression adj algorithm$)
Manually weighted query:
 {( IMAGE ,         80 ),
  ( COMPRESSION ,   80 ),
  ( REAL ,          70 ),
  ( TIME ,          70 ),
  ( ALGORITHM$ ,    20 ) }
```

```
Automatically weighted query:
 {( IMAGE ,         3.441395 ),
  ( COMPRESSION ,   4.858546 ),
  ( REAL ,          3.481345 ),
  ( TIME ,          1.911407 ),
  ( ALGORITHM$ ,    2.844967 ) }
```

Table A: Query texts

References

[Fox 88] Fox, E.A., Koll, M.B.: Practical Enhanced Boolean Retrieval: Experiences with the SMART and SIRE Systems. *Information Processing & Management,* 24(3): 257-267, 1988.

[Fre 91a] Frei, H.P., Schäuble, P.: Determining the Effectiveness of Retrieval Algorithms. *Information Processing & Management,* 27(2-3): 153-164, 1991.

[Fre 91b] Frei, H.P., Meienberg, S., Schäuble, P.: The Perils of Interpreting Recall and Precision Values. *Informatik Fachberichte*, 289: 1-10, Springer-Verlag, Berlin, 1991.

[Fre 91c] Frei, H.P., Meienberg, S.: Evaluating Weighted Search Terms as Boolean Queries. *Informatik Fachberichte*, 289: 11-22, Springer-Verlag, Berlin, 1991.

[Kee 92] Keen, E.M.: Some Aspects of Proximity Searching in Text Retrieval Systems. *J. of Information Science*, 18(2): 89-98, 1992.

[Mea 81] Meadow, C.T., Cochrane, P.A.: *Basics of Online Searching*. Wiley, New York, 1981.

[Pai 84] Paice, C.D.: Soft Evaluation of Boolean Search Queries in Information Retrieval Systems. *Information Technology*, 3(1): 33-41, 1984.

[Qiu 92] Qiu, Y.: ISIR: an Integrated System for Information Retrieval. *Proc. 14th IR Colloquium, British Computer Society*, Lancaster, 1992.

[Qiu 93] Qiu, Y., Frei, H.P.: Concept Based Query Expansion. To appear in: *Proc. of the 16th Int. ACM SIGIR Conf. on R&D in Information Retrieval*, Pittsburgh, SIGIR Forum, ACM Press, June 1993.

[Rad 82] Radecki, T.: Reducing the Perils of Merging Boolean and Weighted Retrieval Systems. *J. of Documentation*, 38(3): 207-11, 1982.

[Rad 88] Radecki, T.: Trends in Research on Information Retrieval - The Potential for Improvements in Conventional Boolean Retrieval Systems. *Information Processing & Management*, 24(3): 219-227, 1988.

[Rob 86] Robertson, S.E., Thompson, C.L., Macaskill, M.J.: Weighting, Ranking and Relevance Feedback in a Front-end System. *J. of Information Science*, 12: 71-75, 1986.

[Rob 90] Robertson, S.E., Thompson, C.L.: Weighted Searching: The Cirt Experiment. *Prospects for Intelligent Retrieval,* Informatics 10: 153-165, Aslib, London, 1990.

[Wal 79] Waller, W.G., Kraft, D.H.: A Mathematical Model of a Weighted Boolean Retrieval System. *Information Processing & Management,* 15(5): 235-245, 1979.

[Won 89] Wong, S.K.M., Ziarko, W., Raghavan, V.V., Wong, P.C.N.: Extended Boolean Query Processing in the Generalized Vector Space Model. *Information Systems*, 14(1): 47-63, 1989.