$f(R_1, R_2)$ computed for all collections of documents in one evaluation cycle. Then inequality (1) has to be checked again for other subrequests using the results of the evaluation cycle. The size of an evaluation cycle depends on the assumptions that the algorithm uses in deciding when to remove a subrequest. For example, if inequality (1) is satisfied at least four times in five collections of documents, then the subrequest could be removed and five could be taken as the number of collections of documents in one evaluation cycle.

In designing an algorithm to correct a query formulation with the intent of increasing the recall level of a search, the idea is to include additional subrequests in the query formulation, which would increase the recall level. This algorithm is based on the following assumptions:

2. Additional subrequests should be constructed automatically (using an algorithm for automatic indexing of search requests) on the basis of the additional information about POIN.
3. For some users not all new subrequests should be added but only those that result in substantially different output.
   3.1. Whether the difference in the outputs is sufficient for the inclusion of a new subrequest is determined for each user; the determination depends on the sizes of the outputs obtained from the original query formulation and from the new subrequest.

We introduce the following notation.

C—the set of documents contained in the output from the original query formulation
D—the set of documents contained in the output from the additional subrequest
$\alpha(x)$—the number of elements in set x

The following formula represents a possible way to use assumptions 3 and 3.1, that is, the higher ratio of $\alpha(D) - \alpha(C \cap D)$ to $\alpha(D)$ indicates more substantial difference in the outputs from the query formulation and from the new subrequest:

$$[\alpha(D) - \alpha(C \cap D)]/\alpha(D) > f(\alpha(C),\alpha(D)). \qquad (2)$$

Assumption 3.1 is indicated on the right-hand side because f depends on the cardinality of C and D. If $f(\alpha(C),\alpha(D)) = 0$ for some user, then all of the additional subrequests will be added in the new query formulation.

In constructing the algorithm, another assumption (analogous to assumption 1.2) is being used.

   3.2. The satisfaction of inequality (2) on one collection of documents could be accidental, and before making a decision about correcting a query formulation it is necessary to make sure that satisfaction of inequality (2) was not accidental.

Now we describe an algorithm that is based on assumptions 2, 3, 3.1, and 3.2.

On the basis of additional information received from the user the query formulation is constructed (by the algorithm for the automatic indexing of search requests). If this query formulation contains a new subrequest it is tested through an evaluation cycle. This is done as follows. The search is performed using the original query formulation and the new subrequest. Then the values of $\alpha(D)$ and $\alpha(C \cap D)$ are computed and inequality (2) is checked. The information about the satisfaction (or the lack of satisfaction) of this inequality is recorded in a corresponding cycle. This is done until enough statistical data are accumulated to allow the algorithm to make a decision with respect to including the subrequest in the final version of the corrected query formulation. If more than one subrequest is tested, then the subrequest to be added to the original query formulation is the one with the highest average value of

$$[\alpha(D) - \alpha(C \cap D)]/\alpha(D) - f(\alpha(C),\alpha(D))$$

computed for all collections of documents in one evaluation cycle. Then inequality (2) has to be checked again for other subrequests using the results of the same evaluation cycle. The query formulation used in this evaluation is a quasi-corrected (by other subrequest) query formulation. This is done for all remaining subrequests to be tested.

The size of an evaluation cycle depends on the assumptions that the algorithm uses in deciding when to add a subrequest. The computation is similar for the removal of subrequests. After adding new subrequests, the algorithm of "noise removal" (the first algorithm) is invoked to correct the obtained query formulation. After the first algorithm is finished, the final query formulation is considered constructed.

The described algorithm of adaptive feedback was tested in 1972 on the IR system Informatica. This system's collection of documents covered all areas of information science. The results clearly demonstrated an improvement in the precision and recall levels of search for every search request used in the experiments.

We mentioned earlier that in only a few papers were concrete methods proposed for automatic feedback in Boolean IR systems, and in describing automatic feedback some authors used only algorithms for the automatic indexing of search requests. Moreover, in the majority of these papers, the feedback process was only considered for static collection of documents, although it was not mentioned explicitly. As an example, we can mention the approach used by Salton and coauthors (Salton, Fox, & Voorhees, 1985) in realizing feedback in a static collection of documents. The authors proposed that researchers consider descriptors from the pertinent documents obtained from the first iteration (from the original query formulation) as well as from subsequent iterations (in the process of one session in an on-line search), but they assigned different weights