cates are removed), which is given to the user. The user evaluates this output—that is, the user indicates which documents are pertinent to his or her information need—and then returns the results to the system. The algorithm begins by checking if there are any pertinent documents in the output. If no pertinent documents are found in the output, then the algorithm stops (because there is no additional information for the selection of the best alternative). In general, this does not mean there is no more interaction between the user and the system. For example, the user can reformulate the request or change the search parameters (such as altering the acceptable number of documents in the output).

If pertinent documents exist in the combined output, the feedback algorithm computes the value r^2/n for each individual output (corresponding to a specific query formulation) containing at least one pertinent document. The output with the largest value r^2/n indicates that the algorithm that constructed the corresponding query formulation is the most appropriate (gives the best search results) for the given search request and a specific user. This algorithm is going to be used in the subsequent iterations of the feedback algorithm.

Denote by A the algorithm that constructed the query formulation (denoted by PQ) with the best output (largest r^2/n). The feedback algorithm constructs a new marked set by combining the marked set (used by A in constructing PQ) with the pertinent documents found by the user in the previous output. Then on the basis of this new marked set the A algorithm constructs a new query formulation, denoted by AQ (auxiliary query formulation). This query formulation is then compared to PQ, and the algorithm constructs a new query formulation, denoted by NQ, by removing from AQ all subrequests that appear in PQ and also those subrequests that contain any subrequests appearing in PQ.

It is possible for NQ to be empty (if all subrequests of AQ were removed). In this case, the algorithm stops because no new documents could be found. If NQ is not empty, then another search is performed and new documents not appearing in previous outputs (if such exist) are given to the user. This is the end of one feedback iteration.

In cases where several outputs have the same largest value r^2/n, that is, there is more than one "best" algorithm, all of these algorithms are used to construct their AQs and then the logical operator OR ($\vee$) is used to join them into one query formulation (AQ). Notice that this AQ is also in a disjunctive normal form. Then NQ is constructed by removing from this AQ all the subrequests that either coincide with subrequests in the PQs constructed by the best algorithms or contain those subrequests as proper subsets (see the previous example). The NQ is used to perform the search and the new documents (not appearing in previous output) are given to the user.

Clearly it is possible to extend this process through several iterations. In this case, the pertinent documents found by the combined query formulation are added to the marked set used in the previous search and all the algorithms in the system are used to construct a new combined query formulation. This process is repeated for the new iteration, and if the evaluation of the alternatives

gives the same result as the evaluation of alternatives from the previous iteration, then the system can be changed into a better state. Clearly, the number of iterations for which the results of evaluation coincide, as well as the level of this coincidence, has to be determined by the designers of the system and could vary substantially from system to system. After the system is changed into a better state it is expedient, for improving the quality of information service, to use adaptive feedback (a possible algorithm for adaptive feedback was described earlier). The expediency of such combined feedback was discussed in Chapter 4.

The algorithm for selective feedback stops in the following cases:

1. No evaluation of the output is provided by the user.
2. No pertinent documents are found (by the user) in the output.
3. No new subrequests are constructed, that is, NQ is empty.
4. No new documents are found by the system.

In Table 9.7 the main steps in the feedback algorithm for the IR system performing optimal search in a static collection of documents are described.

The preceding algorithms may be further developed and modified to satisfy special requirements of the developers as well as to satisfy users of the system. For example, when the collection of documents is very large (say, in a retrospective search) and the size of the output is too large for the user to consider, there are several ways to improve the feedback algorithm.

1. Using only part of the entire collection of documents at the first step. The selection of the sample is done randomly, and only this sample is used in the process of choosing the best algorithm and correcting query formulations. Then, at the user's request, the search with the corrected query formulation may be performed on the entire collection.

Table 9.7
Summary of the Steps in One Iteration of the Selective Feedback Algorithm for a Static Collection of Documents

1. Pertinent documents from the previous output (using the user's evaluation) are selected.
2. If the previous search was based on a combined query formulation, constructed by more than one algorithm, then these algorithms are compared using the criterion $\sqrt{R \cdot P}$ (or more precisely r^2/n) and the best algorithm is selected.
3. The best algorithm (algorithms if there are more than one) is used to construct an auxiliary query formulation (AQ) (combined from several query formulations in the case of more than one algorithm).
4. A new query formulation (NQ) is constructed by removing from AQ all subrequests that appear in PQ and subrequests that include subrequests of PQ.
5. If NQ is not empty, the search is performed and the preliminary output is obtained.
6. A new output is formed by removing from the preliminary output all the documents appearing in previous outputs. Then the new output (if not empty) is given to the user.