request T in Q such that all descriptors in T also appear in S (i.e., $S \cap T = S$). This NQ will become the final query formulation and will be used in the search of the new collection if there are no nonpertinent documents in the previous output. If, on the other hand, there are nonpertinent documents, then the feedback algorithm proceeds with the precision step.

The precision step starts by identifying all subrequests in PQ that were used to obtain nonpertinent documents. These subrequests are the candidates for removal from NQ. They will be removed from NQ if they did not appear in AQ, because they were used to retrieve noise and had low weight in the new marked set. However, they will be left as part of the final query formulation if they appear in AQ because their weight in the new marked set exceeded a certain threshold and possibly they will perform better in the new collection. After applying this procedure to all subrequests appearing in PQ and removing the appropriate subrequests from NQ, the final query formulation is obtained and it will be used for the search in the new collection of documents.

It is possible that the previous query formulation will be used for the search in the new collection of documents. This can occur in the following three cases:

1. There is no user's evaluation of the system's output.
2. The output from the previous collection does not contain any pertinent documents.
3. After the application of the feedback algorithm, the new query formulation coincides with the previous query formulation (i.e., no subrequests were removed from or added to the previous query formulation).

It is necessary to mention one more technical point about the feedback algorithm. From the previous discussion it is clear that when we construct a marked set at each iteration (for both static and dynamic collections) we combine pertinent documents from the last output with pertinent documents from previous outputs. This marked set may become very large and we may want to control its size by either reducing the number of iterations included (for example, to five or six iterations) or by putting a limit on the number of documents in the marked set and removing the documents from the earlier iterations. This consideration is more relevant for the dynamic collection of documents. Table 9.2 summarizes the steps involved in one iteration of the feedback algorithm for the dynamic collection of documents.

It is sometimes advantageous to combine two feedback algorithms (for static and dynamic collections). This would be appropriate, for example, if a user were interested in SDI using on-line systems. In this example, the user's interaction with the system during one session is similar to the case of the static collection, and the corresponding feedback algorithm can be applied. At the end of the session, the system stores the set of all pertinent documents used to construct the final query formulation. A new session (with the new collection) begins with the construction of a new query formulation based on the marked set stored from the previous session and using the descriptor frequencies from the new collection (as in the feedback algorithm for the dynamic collection). After the search based on this query formulation has been completed and the user has evaluated the system's output, the feedback algorithm for the static collection is applied again. It should be pointed out that combining the two algorithms seems the most expedient because in this case the system takes into account two attributes (2 and 4) of POIN as it adapts to the user's need.

Table 9.2
Summary of the Steps in One Iteration of the Feedback Algorithm for Dynamic Collection

1. Pertinent documents from the previous output (using the user's evaluation) are selected. If there are no pertinent documents (or there is no user's evaluation), then the previous query formulation is used for the search in the new collection of documents (Step 7). If there are pertinent documents, then the system proceeds to Step 2.
2. A new marked set is constructed by combining the pertinent documents obtained in Step 1 and the pertinent documents used for the construction of the previous query formulation.
3. An auxiliary query formulation (AQ) is constructed using as input the marked set obtained in Step 2 and new descriptor frequencies from the new collection.
4. A modified query formulation (MQ) is constructed by adding new subrequests to AQ (using the recall step described earlier).
5. Nonpertinent documents are selected from the previous output. If there are no such documents, the modified query formulation from Step 4 becomes the final query formulation, and the search is performed (Step 7). If there are nonpertinent documents, then the system continues on to Step 6.
6. The subrequests from PQ that were used in finding nonpertinent documents are selected. These subrequests are removed from MQ if they did not appear in AQ (i.e., the precision step is applied). The resulting query formulation is the final query formulation for this iteration.
7. The final output is obtained as the result of the search in the new collection based on the final query formulation.

9.6 Evaluation

The feedback algorithms described in the chapter were tested experimentally on the IR system AIRS (adaptive IR system) developed at Fordham University. We will now describe the most important results of these experiments.

The collection consisted of 2504 documents in computer science, selected from the journal *Computer Abstracts,* and 28 queries with relevance judgments. The collection statistics are summarized in Table 9.3.