from the most meaningful descriptors. Moreover, the combination of descriptors does not necessarily have to be meaningful. While constructing a query formulation in the following example, our algorithm used a descriptor presumably void of any meaning. That query formulation gave a better search result than the one based on the expert's query formulation.

In one of the search requests on which the algorithm was tested, the user was interested in the design and functioning of computers used on moving objects, such as ships, airplanes, and rockets. The search was performed on the collection of 6323 documents selected from the journal of abstracts *Computer Science* (Moscow, VINITI, 1967). The search request was given by the user in the form of seven pertinent documents. The expert analyzing the given search request constructed the following query formulation:

(ship AND computer) OR (aircraft AND computer)
OR (sputnik AND computer) OR (rocket AND computer)
OR (flying machine AND computer) OR (submarine AND computer)

Each of the combinations (every subrequest) makes sense and was easily found using the system's thesaurus. But the output based on this query formulation contained a lot of noise (nonpertinent documents). For example, among the retrieved documents were many documents that dealt with processing data collected from moving objects, but the computers were located elsewhere.

The search based on the query formulation constructed by the algorithm gave much better results. This query formulation contained subrequests that did not seem too meaningful, and the subrequest that gave the best search result consisted of one descriptor "foundation." The analysis of the results showed that, in their discussion of the computer features they developed for moving objects, the documents' authors often referred to what they considered to be an important achievement—the elimination of the necessity for a special foundation. Further analysis showed that at the beginning of the 1960s, many articles discussed special foundations that were built for computer rooms, and therefore the descriptor "foundation" was included in the thesaurus in *Computer Science*. But in the collection of documents published in 1967, there were no articles discussing the building of foundations. Therefore, any document containing the descriptor "foundation" turned out to be pertinent. This descriptor was also very effective when a search was performed on the documents published in 1968. Later this word disappeared from articles in *Computer Science*. Thus, when constructing a query formulation the algorithm "discovered" a style of writing articles that was prevalent during a specific period of the development of *Computer Science*. Clearly, it is very difficult, if not impossible, for an expert to identify such a subrequest as a "foundation."

Another important aspect of the described algorithm is that in information retrieval systems using this algorithm it is possible to perform an automatic indexing of the documents by simply looking up the terms of the documents in the system's dictionary. This method of automatic indexing was tried in many

systems but it was not generally accepted because this system of indexing produces document profiles consisting of a very large number of descriptors. For example, the document profile of a 200-word abstract will typically contain several dozen descriptors. But when a query formulation is constructed by a human being, our experience shows that most subrequests consist of one, two, or three descriptors, rarely four, and very seldom five or six. In fact, because of the difficulty of constructing subrequests consisting of, say seventeen descriptors, such an approach to automatic indexing was considered unacceptable in systems using Boolean search. The algorithm described in this chapter allows the successful use of automatic indexing of the documents, because it can construct subrequests of practically any length. Moreover, automatic indexing of the documents becomes a desirable feature because it introduces uniformity and stability, which in turn improves the occurrence frequencies used by the suggested algorithm.

In this chapter we considered several assumptions that might be used in developing algorithms for the automatic construction of effective query formulations. On the basis of several pragmatic assumptions, we developed an algorithm for the automatic construction of query formulations in Boolean form. The suggested algorithm not only substitutes a traditional intellectual process for IR systems by an algorithmic process, but it also substantially simplifies the end-user problem. The simplicity of the communication between a user and an information retrieval system was one of the main considerations in designing the algorithm. The user expresses his or her search request by a set of documents pertinent to the user's need. This does not require any specialized knowledge on the part of the user about the system's operation and design. The quality of the search depends only on the algorithm and how completely and precisely the user's POIN is represented in the user's search request.

This algorithm is, without a doubt, not the last in the series of algorithms that could be created for indexing of queries. In the future we will see others that may be more suitable, both for certain users and for certain IR systems. This is true for all algorithms used in the IR system—not only for those involved in query indexing. However, the analyzed algorithm is a successfully functioning version of the design of an IR system's element (such as BIQ), and its use in conjunction with the algorithm for the indexing of documents provides a solution to the problem of the complete automation of indexing (documents and queries) in the Boolean IR system.

7.7 Conclusion

In the framework of developing and operating IR systems, it is traditionally acknowledged that one of the most complex problems is the automatic in-