well. Here is one simple but sufficiently effective method to find new words that should be included in the descriptor dictionary (or the list of stop words). If a certain word is not included either in the stop list or in the descriptor dictionary, but is encountered in each M documents out of a total of N, then this word is regarded as a new term and is provided to IR system experts responsible for the maintenance (correction) of all vocabularies and lists of the system. These experts then decide on which dictionary list this word should be included. For example, if a certain word, W, is not included on any list or in vocabularies but is encountered in an average of 5 out of every 100 documents, then this word is given to the system experts. Implementing such a simple algorithm does not affect the performance of the automatic indexing and, at the same time, is useful for the overall operation of the system. Thus, the use of stop lists in the automatic indexing can be judged reasonable from various points of view.

We have considered the main methods of automatic text analysis used in the development of vocabulary-assisted automatic indexing algorithms. Now, we can consider one such algorithm.

6.5 Algorithm of the Automatic Indexing of Documents

Algorithms of automatic indexing, used in IR systems providing services to the users, appeared in the late 1960s. Some time later, several companies proposed various information retrieval software packages including automatic indexing procedures. The document program system (DPS) software package developed by IBM is one example. It is worth noting that during the past 20 years, no essential changes in the ideas of vocabulary-assisted automatic indexing algorithms, that is, algorithms for constructing document profiles from descriptors that are most characteristic for the given topic, have been made. Therefore, the algorithm described next can be regarded as rather typical for the entire family of algorithms utilizing the Boolean search (Frants & Voiskunskii, 1971).

This algorithm begins its work with the first sentence of the text that is input. Note that the title of the document is considered its opening sentence. The sentence separation is performed in the following manner. The algorithm considers any text standing before the first period or between two periods as a sentence. Additionally, all expressions standing in brackets within the sentence are also considered sentences. Words in quotations are rejected from consideration (from the sentence). At the next stage, all unimportant words contained in the stop list are removed from the sentence. In other words, each word of the separated sentence is compared with each word of the stop list. In case of coincidence (match), the word is removed from the sentence being analyzed. Then the algorithm begins to search for the sentence parts that can contain word combinations. These fragments are the group of words standing between two punc-

tuation marks, between the beginning of the text and the punctuation mark, or between the punctuation mark and the end of the sentence. If no punctuation marks are contained within the sentence, all words of the sentence are considered to be fragments. In the framework of the algorithm under consideration, the following rules are used to recognize word combinations:

1. Three or four words of natural language make up a word combination and are always translated by the same descriptor if they are not separated by punctuation marks in the document text and if they are included in the word combination dictionary as a combination.
2. Two words of natural language make up a word combination and are always translated by the same descriptor if they are not separated by punctuation marks in the document text, if they are not separated by more than three words, and if they are included in the word combination dictionary as a combination.

First, the algorithm attempts to find a four-word word combination. It checks each fragment of the sentence to see whether its words contain a four-word combination; that is, each sequence of four words is compared with the vocabulary of four-word word combinations. If such a word combination is found, these words are removed from the fragment, and the corresponding descriptor is included in the document profile, providing that this descriptor has not been included earlier. The search is continued for the next four-word word combination until all four-word word combinations are recognized or until the algorithm finds that no four-word word combinations are included in the sentence. Then the algorithm begins to search for triple word combinations. The method of search is similar to that described for four-word combinations. In this case, word combinations also may or may not be found. If they are found, the algorithm removes such word combinations from the sentence, and the corresponding descriptors are included in the document profile, providing that these descriptors were not included earlier.

After the search for triple word combinations is completed, the algorithm begins to search for double word combinations. This procedure is implemented in two stages. At the first stage, the algorithm analyzes double word combinations in the same manner as it does for triple word and four-word word combinations; thus, all procedures performed with these word combinations are similar. But after the first stage is completed, the algorithm begins the second stage; that is, the search for double word combinations is continued in the following manner. The algorithm fixes the first word of the fragment and unifies it not with the word next to it on the right (this was done at the first stage) but with the following word to the right (of course, this is necessary only if there is such a word). (Note that the word to which other words are connected is called the base word.) The combination obtained is compared with word combinations from the dictionary of double word combinations. If the combination coincides,