then the algorithm implements the procedures described earlier. If the combination does not coincide, the base word is unified with the next, more remote word (again, if there is such a word in the fragment) and the procedure is repeated. The most remote word connected to the base word is one separated from the base word by three words (see rule 2). If no words to the right of the base word form word combinations with it, the algorithm examines words to the left of the base word. If such words exist (which are, again, separated from the base word by three words at the most), they are connected to the base word from the proper side (i.e., the base word is always the first word of the word combination), and all procedures are repeated. Clearly, when the base word is the first word of the fragment, the left part of the fragment is absent. Having examined all the permitted words to the left of the base word, the algorithm selects the next word of the fragment (if there is one) as a base word and performs all the preceding procedures with it. Thus, the procedure is repeated until the last word of the last fragment is analyzed. Then the second stage of finding double sentence combinations is completed together with the whole process of finding sentence combinations in the sentence.

Then the algorithm removes all punctuation marks, stop words, and words in quotations from the sentence. It then compares the remaining part of the sentence with the descriptor dictionary to make certain decisions. The first word of the sentence is compared with all words (word forms) contained in the conditional equivalence classes of the descriptor dictionary. In case of coincidence, the corresponding descriptor is included in the document profile, providing that it was not included earlier, and the algorithm compares the next word of the sentence. If the analyzed word is not contained in the descriptor dictionary, it is included in the list of undefined words, providing that it was not introduced in this list earlier. Then the algorithm analyzes the next word of the sentence repeating the procedures described. This continues until the last word of the sentence is analyzed. Then the algorithm begins to analyze the next sentence of the document, and all the preceding procedures are repeated. After the last sentence has been analyzed, indexing of the given document is considered complete. The document profile constructed in the course of indexing is written into a special file. The "side product" of indexing, namely, the list of stop words, is used to update the special stop list of the system. Lists are arranged in a way that allows the IR system to account for the occurrence frequency of stop words in a given number of documents (e.g., in every hundred). If the list of the document contains a word that is not included in the system list, this word is added to the system list. Otherwise, its occurrence frequency is updated (enlarged by a unit). When the occurrence frequency of a certain word exceeds an established threshold, the system experts analyze it and decide whether this word should be included in either the descriptor dictionary or the stop list.

The general steps for constructing a document profile are given in Figure 6.4.

1. Separation of the sentence from the document text for further indexing.
2. Removal of stop words from the sentence.
3. Recognition and indexing of word combinations contained in the sentence.
4. Indexing of single words in the sentence and final creation of the document profile.

Figure 6.4
Steps for constructing a document profile.

The automatic indexing algorithm just presented is quite similar to that developed by the authors in 1970. The source code in the LISP programming language for this algorithm was developed and compiled by one of the authors. The main difference is that the algorithm of 1970 automatically separated stems. However, for a number of the reasons stated, we consider it more advantageous not to use stemming procedures.

Thus, we have considered the algorithm for selecting descriptors for document profile. Descriptors selected are those judged most important from the point of view of the topic of the collection. As mentioned previously, algorithms of this kind are most widely used in functioning IR systems. Nevertheless, let us briefly consider other methods used to select descriptors for the document profile that seek to select descriptors that more fully represent the meanings of the document being indexed. First, such indexing can be used in systems utilizing Boolean search and, second, we believe that familiarizing readers with the main ideas of this investigative line will be useful.

6.6 Statistical Indexing Methods

When considering the main directions in automatic indexing, we have already mentioned the problem of determining the most important (for the meaning) terms of documents. One would say that certain linguistic methods (algorithms) should be used for this purpose first. However, linguists lack any successful algorithms. That is why researchers use mainly statistical algorithms that calculate the measure of meaning in a term of the document. This seems very convenient for the methods incorporating descriptor weights, because the measure of meaning calculated can be considered as a weight.

In 1958 Luhn suggested that the frequency of word occurrence in an article could furnish a useful measurement of word significance in the article: "The justification of measuring word significance by use-frequency is based on