ods for distinguishing these meanings, many researchers accept the quite reasonable assumption that there is a certain group of words that will explain the meaning of each homonym. Furthermore, they reason that when a homonym is encountered surrounded by one group of words, it adopts one meaning, and when the homonym is surrounded by another group of words, it has another meaning. This encourages the developers to try to isolate an explaining group of words for each homonym and to divide it into subgroups corresponding to the homonym's meanings (the number of subgroups is equal to the number of meanings the homonym can adopt). These explaining groups allow us to develop rather simple distinguishing algorithms. The essence of these algorithms is the following. First, a homonym that coincides with one from the descriptor dictionary is found in the text (such words are specially marked in the dictionary). Then the surrounding words are compared with the explaining words listed to determine the concrete meaning of the concrete homonym. Normally, the very first explaining word encountered determines the meaning of the given homonym. If the surrounding words do not include explaining words, the homonym is not distinguished. It is clear that when implementing this method one must determine the size of the surroundings (normally no more than several words) and the sequential order in which they should be examined. For instance, some researchers suggest that the following examination order is best: (1) the first word to the left, (2) the first word to the right, and (3) the second word to the right. In addition, in practice one must deal with rather large lists of explaining words. For example, in the description of one of the existing automatic indexing algorithms (the IR system in the electrical engineering area), a list of 631 explaining words has been compiled to recognize meanings of a single homonym. Of these, 411 words are used to help the IR system recognize one meaning and 220 words help it to recognize the other meaning. The experimental verification of distinguishing the meanings of this homonym has shown approximately 86% correct recognitions, 11 to 12% nondistinguishing responses, and 2 to 3% errors in the set of 1000 homonyms (Fedorow, 1973).

One can easily see that this method is sufficiently simple and effective. However, its effectiveness depends on the quality of the distinguishing word lists. These lists are compiled not on the basis of strict methods and scientific recommendations but on the basis of the developer's expertise. In addition, the need for hundreds of explaining words for each homonym makes the method rather awkward and reduces the performance of the entire indexing process. Given all the factors mentioned above and the small number of homonyms contained in the descriptor dictionary (this number, however, will depend on the natural language used), the lexical homonyms in many systems are not distinguished.

Methods for recognizing word combinations are used far more widely in practice. Normally, the word combination is understood as a rather stable sequence of two or more words (within the sentence) unified according to grammar and meaning. For instance, in computer science word combinations such as "artificial intelligence," "central processing unit," and "systems analysis" are universally accepted. When developing algorithms for recognition of these word combinations, researchers often divide them into two types. The first type includes word combinations that can be regarded as a single word: "binary system" is a good example. Word combinations of the second type are encountered in document texts similar to those of the first type, but they can also be encountered in a more separated form. For example, the document may contain the expression "primary and secondary memory," whereas the descriptor dictionary contains the word combinations "primary memory" and "secondary memory." (Obviously, in the course of automatic indexing only the word combinations included in the descriptor dictionary are recognized.)

The majority of the existing algorithms are based on the assumption that in sentences all words of the word combination can be encountered in an arbitrary order and are not separated by punctuation marks. Obviously, the same word can be contained in several word combinations with various words, for example, "binary system," "binary tree," "binary logic," and "binary code." Such a word (in this case "binary") is considered a main element (word) of the word combination. All such words from all word combinations contained in the dictionary are included in a "special" list. The following algorithm is a representative example.

An arbitrary current word from the sentence being analyzed is compared with main words. If the comparison result is negative, the system examines the next word of the sentence. Otherwise, it checks words to the right of the examined word (if that is possible) to see if one of them is not a main word. If successful, the system assumes that it has found a possible two-word word combination. It then compares this word combination with two-word word combinations included in the dictionary. If the same word combination exists in the dictionary, it is included in the document profile. Otherwise, the system then seeks the next, third word of the word combination among nonmain words. If it finds such a word, the system assumes that a possible three-word word combination is found, and so forth. Normally, systems use word combinations consisting of no more than four words. If the verification of words to the right of the word under examination (the main word) does not provide word combinations, then the system examines the group of words to the left. (A number of algorithms include some additional rules. For instance, it is supposed that words of a word combination should not be separated by more than three words.) If the word combination is not found after all the described operations, the system examines the next word of the text by the same procedures. As a rule, the verification of a word combination is continued until the end of the sentence or a comma. The remaining part of the sentence (after removal of the word combination found and commas, if present) is arranged into a sentence for further indexing.