- Place putative sentence boundaries after all occurrences of . ? ! (and maybe ; : —)
- Move the boundary after following quotation marks, if any.
- Disqualify a period boundary in the following circumstances:
  - If it is preceded by a known abbreviation of a sort that does not normally occur sentence finally, but is commonly followed by a capitalized proper name, such as *Prof.* or *vs.*
  - If it is preceded by a known abbreviation and not followed by an uppercase word. This will deal correctly with most usages of abbreviations like *etc.* or *Jr.* which can occur sentence medially or finally.
- Disqualify a boundary with a ? or ! if:
  - It is followed by a lowercase letter (or a known name).
- Regard other putative sentence boundaries as sentence boundaries.

Figure 4.1 Heuristic sentence boundary detection algorithm.

have used heuristic algorithms of this sort. With enough effort in their development, they can work very well, at least within the textual domain for which they were built. But any such solution suffers from the same problems of heuristic processes in other parts of the tokenization process. They require a lot of hand-coding and domain knowledge on the part of the person constructing the tokenizer, and tend to be brittle and domain-specific.

There has been increasing research recently on more principled methods of sentence boundary detection. Riley (1989) used statistical classification trees to determine sentence boundaries. The features for the classification trees include the case and length of the words preceding and following a period, and the a priori probability of different words to occur before and after a sentence boundary (the computation of which requires a large quantity of labeled training data). Palmer and Hearst (1994; 1997) avoid the need for acquiring such data by simply using the part of speech distribution of the preceding and following words, and using a neural network to predict sentence boundaries. This yields a