```
#!/usr/bin/perl

$letternumber = "[A-Za-z0-9]";
$notletter = "[^A-Za-z0-9]";
$alwayssep = "[\\?!()\";/\\|`]";
$clitic = "('|:|-|'S|'D|'M|'LL|'RE|'VE|N'T|'s|'d|'m|'ll|'re|'ve|n't)";

$abbr{"Co."} = 1; $abbr{"Dr."} = 1; $abbr{"Jan."} = 1; $abbr{"Feb."} = 1;

while ($line = <>){ # read the next line from standard input

    # put whitespace around unambiguous separators
    $line =~ s/$alwayssep/ $& /g;

    # put whitespace around commas that aren't inside numbers
    $line =~ s/([^0-9]),/$1 , /g;
    $line =~ s/,([^0-9])/ , $1/g;

    # distinguish singlequotes from apostrophes by
    # segmenting off single quotes not preceded by letter
    $line =~ s/^'/$& /g;
    $line =~ s/($notletter)'/$1 '/g;

    # segment off unambiguous word-final clitics and punctuation
    $line =~ s/$clitic$/ $&/g;
    $line =~ s/$clitic($notletter)/ $1 $2/g;

   # now deal with periods. For each possible word
   @possiblewords=split(/\s+/,$line);
   foreach $word (@possiblewords) { #
      if it ends in a period, if
      (($word =~ /$letternumber\./)
              && !($abbr{$word}) # and isn't on the abbreviation list
                 # and isn't a sequence of letters and periods (U.S.)
                 # and doesn't resemble an abbreviation (no vowels: Inc.)
             && !($word =~
                  /^([A-Za-z]\.([A-Za-z]\.)+|[A-Z][bcdfghj-nptvxz]+\.)$/))
          { # then segment off the period
          $word =~ s/\.$/ \./;
      }
      # expand clitics
      $word =~s/'ve/have/;
      $word =~s/'m/am/;
      print $word," ";
   } print
 "\n";
}
```

Figure 3.22 A sample English tokenization script, adapted from Grefenstette (1999) and Palmer (2000). A real script would have a longer abbreviation dictionary.

D. Jurafsky, James H. Martin (2008) Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition, Prentice Hall, Upper Saddle River, N.J. , 2008.