Title: An algorithm for suffix stripping

Problem Statement:

An algorithm for suffix stripping is trying solve a problem about removing suffixes from words in English. The precision and recall of the input aren't very different in the earlier suffix removal algorithm and present suffix removal algorithm.

Assumption in prior work:

The algorithms and programs implemented prior are small, fast and reasonable simple. The techniques were not effective at handling irregular forms or exception to the rules because most works focused on simple manipulation techniques, such as removing a fixed set of suffixes or truncating words to a certain length. The addition of more rules to increase the performance of one area of the vocabulary causes and equal degradation of performance somewhere.

Idea:

To get around the limitations of previous approaches, more comprehensive and rule-based approach is taken to suffix stripping by using a set of well-defined rules and heuristics that are designed to handle a wide range of cases, including irregular forms and exceptions to the rules. It works by applying a series of steps to identify and remove suffixes from words, starting with the longest and most specific suffixes and working down to the shorter and more general suffixes.

Technique:

An algorithm for suffix stripping consists of five phases that are applied to words in sequence:

Phase 1: Remove common suffixes from the end of words using a set of rules that identify and remove suffixes such as "sses", "-ed", "-ing", "-ly", and "-es".

Phase 2: Remove additional suffixes from the end of words using a set of rules that identify and remove suffixes such as "-ment", "-ness", "-ize", and "-ate".

Phase 3: Check for certain exception words that do not follow the regular patterns and apply special rules to handle them.

Phase 4: Check for certain double consonants at the end of words and apply a set of rules to handle them.

Phase 5: Remove any remaining suffixes from the end of the word using a set of rules that identify and remove suffixes such as "-ant", "-ence", and "-able".

Evaluation:

To assess the correctness of the algorithm, the output is compared to a manually stemmed version of a sample of words. The manually stemmed version was created by a group of human annotators who were asked to reduce each word to its base form, or stem, using a set of rules and heuristics. It was found that the majority of the words in the sample was stem correctly, with an overall stemmer accuracy of about 95%. The stemmer was most effective at stemming regular forms of words, with an accuracy of about 98%, and was less effective at stemming irregular forms, with an accuracy of about 80%.

Implications:

The introduction of a more comprehensive and rule-based approach to suffix stripping that was designed to handle a wide range of words and to be robust in the face of changes to the language. This made the stemmer more effective at handling irregular forms and exceptions and more suitable for use in a variety of natural language processing tasks. Also, the concept of using a set of rules and heuristics to stem words in a specific order, starting with the longest and most specific suffixes and working down to the shorter and more general suffixes. This approach has been widely adopted in other stemmers and has become a standard approach to suffix stripping.