Spell-Checking Without a Dictionary

Peter Venable
School of Computer Science
Carnegie Mellon University
5000 Forbes Ave., Pittsburgh, PA 15213
pvenable@cs.cmu.edu

December 13, 2000

1 The Problem

The goal of spell-checking to replace incorrectly-spelled words in a document with their correctly-spelled counterparts. An ordinary spell-checker compares each word in a document with a dictionary, prompting the user to select the correct spelling for any unrecognized word, often offering a list of words lexicographically close to the unrecognized word. Simple enough.

When no dictionary is available, however, spell-checking is more difficult. For example, consider a large document containing thousands of technical terms for which no dictionary is readily available, or one in a minority language for which no software has yet been developed.

2 The Solution

In a very large document or group of documents (totaling hundreds of pages) most words which occur at all will occur many times. A random typing mistake (as opposed to a systematic spelling error) will often result in a word that is very rare or even unique (since it is incorrect) and very similar in spelling to the more common word which is the correct spelling. Given a large document and lexicographic distance metric, we can search for words very similar in spelling where one is rare and the other common. If we then replace the misspelled word with the correctly-spelled word, we will have decreased the number of errors in the document. Of course, if we replace a correctly-spelled but rare word with a similar but common word, we will be increasing the number of errors in the document.

3 The Details

Dictionary-free spell-checking can be accomplished as follows:

First, count the frequency of every term in the corpus. Then, for each term t in the document to be checked, attempt (as follows) to find a word which is the correct spelling of that word: Skip numbers and words which are shorter than 4 letters. Then look through the word-frequency list at words w which are not just the same word with different capitalization, which are not just the same word probably made plural by an s at the end, and which are at least 10 times more common than t; that is, $P(w) > 10P(t)$. For each of these words, compute its lexicographic distance from t in characters, $D(t, w)$. Select the word that minimizes this expression:

$$d = min_w D(t, w) \log \frac{1}{P(w)}$$

The lexical distance is combined with the log-likihood of the word (in terms of unigram occurrence) in order to select a word that is both similar in spelling, and relatively common. Then compute a score for this correction, which would replace t with w:

$$s = \frac{\log \frac{P(w)}{P(t)}}{(d+1)^2}$$

4 The Results

To test this technique, we used a 3.9-million-word collection of abstracts from scientific papers. This collection has too many spelling errors to count, which was the motivation for automatic spelling correction. The algorithm described above was used to generate and score a list of spelling corrections, which are misspelled words with their corrected replacements. Then we plotted the change in the number of misspelled words in the document against the minimum score for which we applied corrections. Only the change in the number of errors is known, and not the total number of errors, because no one has time to check 3.9 million words for correctness. Even the twenty thousand or so words which were corrected are too many to check by hand, so we randomly sampled one percent of the corrections for evaluation.

As figure 1 shows, when all spelling corrections scoring 0.25 or higher are applied, about 15,000 more words are spelled correctly than if none are applied. If all corrections are applied regardless of score, those with scores close to zero replace originally-correct words with other words, this driving the word error rate up through the ceiling. A look at figure 2, which is a close-up of the same graph, shows a steady decrease in misspellings as the score threshold decreases to 0.35, and a sharp increase below 0.2.

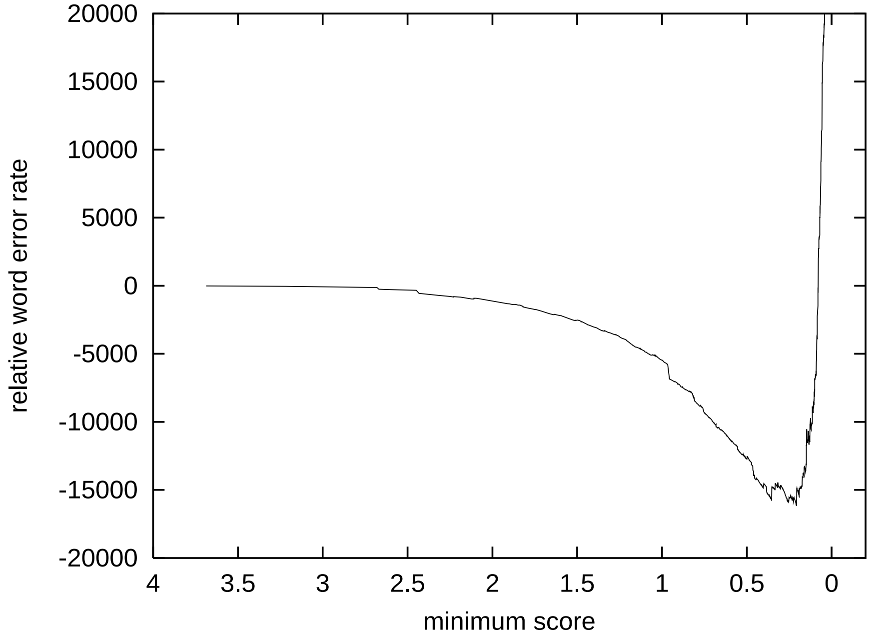

Figure 1: Relative Word Error vs. Minimum Distance Score

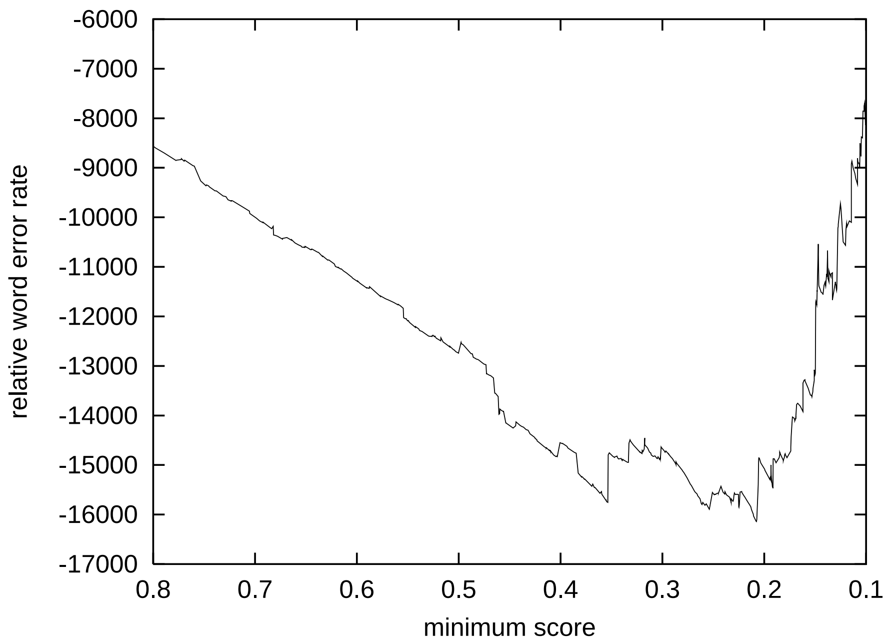

Figure 2: Relative Word Error vs. Minimum Distance Score - Closeup