How to write a Spelling Corrector
Language Technology

Miquel Puig i Mena

September 2019

1 Objective

Analysis of a 80-90 % accuracy Spelling Corrector.

2 Motivation

Modern day-casual writing has evolved to a bidirectional interaction between the writer and a machine agent.

Logically, it's the human form that starts the dialogue and defines text's topic but we've achieved a point where is the intelligent-agent that will drive the author through the writing process by finding misspelled words, proposing entire phrases according to the context or even defining a more suitable skeleton for the document among others.

Given such a broad set of paradigms, this report focuses only on misspelled words and how to provide a feasible correction.

In the online-article [Nor] we can see a simplistic spelling corrector approach that aims to provide a 80-90 % accuracy. Even though it's for educational purposes, it will be the reference of study for the rest of the document.

3 Implementation

Speller corrector provides a simple entry point where you submit a word w and you receive the most likely spelling correction for w (See Eq. 1)

$$Highest\_Confidence\_Candidate = correction(w) \tag{1}$$

Internally, correction function tries to generate a set of candidates derived from the inputted word. For each individual 'c' of generated set, probability P that c was intended when w was inputted is calculated.

Finally, the candidate with highest confidence (**selection mechanism**) is returned as a predicted correction (See Eq. 2). To calculate P(c if w) we have to consider both the probability of c (P(c)) and the probability of the change from c to w (P(w if c)) as independent factors. Using Bayes' Theorem [Wika] the expression can be factorized. (See Eq. 3)

$$Argmax_{c \epsilon candidates}\ P(c|w) \tag{2}$$

$$Argmax_{c \epsilon candidates}\ P(c) * P(w|c) \tag{3}$$

3.1 Language Model P(c)

In an ideal world, P(c) is the probability that c appears as a word of English text[Nor]. Representation of English language has to be defined as a limited and tangible set of words: **dataset**. Note that a dependency has been created since depending on the dataset's quality, the performance of the system will vary.

3.1.1 Dataset

Norvig's experiments used a corpus extracted from a concatenation of public domain book excerpts from Project Gutenberg[al] and lists of most frequent words from Wiktionary[Wikb] and the British National Corpus[Cor].

Probability of candidate among all English language can be determined by counting the number of times each word appears in the dataset.

3.2 Error Model: P(w if c)

The probability that w would be typed in a text when the author meant c[Nor]. A good spelling error model can be characterised but, for Norvig's purposes, a simplistic approach can lead to the aimed accuracy. A list of priorities is defined where each group has infinite preference against it's lower groups; all components within a priority have homogeneous probability. Listed below the segmentation of preferences:

(1) The original word, if it is known; otherwise

(2) The list of known words at edit distance one away, if there are any; otherwise

(3) The list of known words at edit distance two away, if there are any; otherwise

(4) The original word, even though it is not known.

Where a known word is restricted to each different word appeared in the dataset corpus. Finally, subject word to correct has to be derived in every direction such as letter replacement, letter missing, etc. To achieve so, edit1(word) generates a set of different combinations originated in word. Note that generated set has to be filtered later on with combinations that end up in known words.

3.3 Recursive N edits extension

Given such powerfull function as edit1, is straight forward to create combinations to N distance as seen in Norvig's extended edit function "editN".

```
def editsN(word, deep):
    "All edits that are N edits away from `word`."
    if deep == 0:
        return word
    return (eN for e1 in edits1(word) for eN in editsN(e1, deep-1))
```

References

[al] Michael Hart et al. *Project gutenberg.* URL: `https://www.gutenberg.org/`. (accessed: 12.09.2019).

[Cor] British National Corpus. *British National Corpus.* URL: `http://www.natcorp.ox.ac.uk/`. (accessed: 12.09.2019).

[Nor] Peter Norvig. *How to write a Spelling Corrector.* URL: `http://norvig.com/spell-correct.html`. (accessed: 12.09.2019).

[Wika] Wikipedia. *Bayes' theorem.* URL: `https://en.wikipedia.org/wiki/Bayes'_theorem`. (accessed: 12.09.2019).

[Wikb] Wiktionary. *Wiktionary.* URL: `https://www.wiktionary.org/`. (accessed: 12.09.2019).