NLP spelling detection and correction

1. **Spelling errors**
2. **Non-word Errors**: These are the most common type of errors

**Ex**: typing langage instead of language

1. **Real Word Errors**: instead of typing non-word , you end up typing a real word but not the one u meant to type

Ex: typing where instead of were

1. **Datasets:**
2. **British national corpus.txt consists of** 1,115,504 words
3. **Birkbeck spelling error corpus.txt that we will use later for evaluation**
4. **Language models:**
5. **Unigram frequencies:** a language model learns frequencies of n-grams from large text corpora of a language, like English. Once trained, it can be used to evaluate the validity of an n-gram from that language,

Given an input unigram (n-gram with n = 1, a single word) w, we generate a set of candidate unigrams,

Image for post

Image for post

Image for post

This means pick the most frequent word among the set of candidates.

And this could lead us to an unintended error ex:

Image for post

According to unigram frequencies “the” is the right word because it’s the most common word in English , but perhaps the intended word was “thew” and we assumed it was “the “ because the language model said so.

This leads us to another language model called

1. The noisy channel model (the error model):

Itmodels user mistakes and expected user input (based on the data). Mistakes are modeled by weighted edit distances and expected input by a character language model.

Ex:

Image for post

Because you are more likely to type “W” after “e” instead of typing “x” after “e”

(qwerty keyboard)

Building the error model mathematically:

Image for post

the unigram probability, P(c), (given by the language model), multiplied by the probability that w is a spelling error of c, (given by the error model).

The four parts of this expression are:

1. **Selection Mechanism**: argmax  
   We choose the candidate with the highest combined probability.
2. **Candidate Model**: *c ∈ candidates*  
   This tells us which candidate corrections, *c*, to consider.
3. **Language Model**: P(*c*)  
   The probability that *c* appears as a word of English text. For example, occurrences of "the" make up about 7% of English text
4. **Error Model**: P(*w*|*c*)  
   The probability that *w* would be typed in a text when intended to type  *c*. For example, P(*teh*|*the*) is relatively high, but P(*theeexyz*|*the*) would be very low.