Connor Hanley

LIN 313

Venkat

August 8, 2022

Assignment 4

1. Spell Correction: “beeds”

   a) Using one deletion, the spell checker can generate: “beds” and “bees”.

   b) Using one substitution, the spell checker can generate: “deeds” and “feeds”.

   c) Using one insertion, the spell checker can generate: “bleeds” and “beekds”.

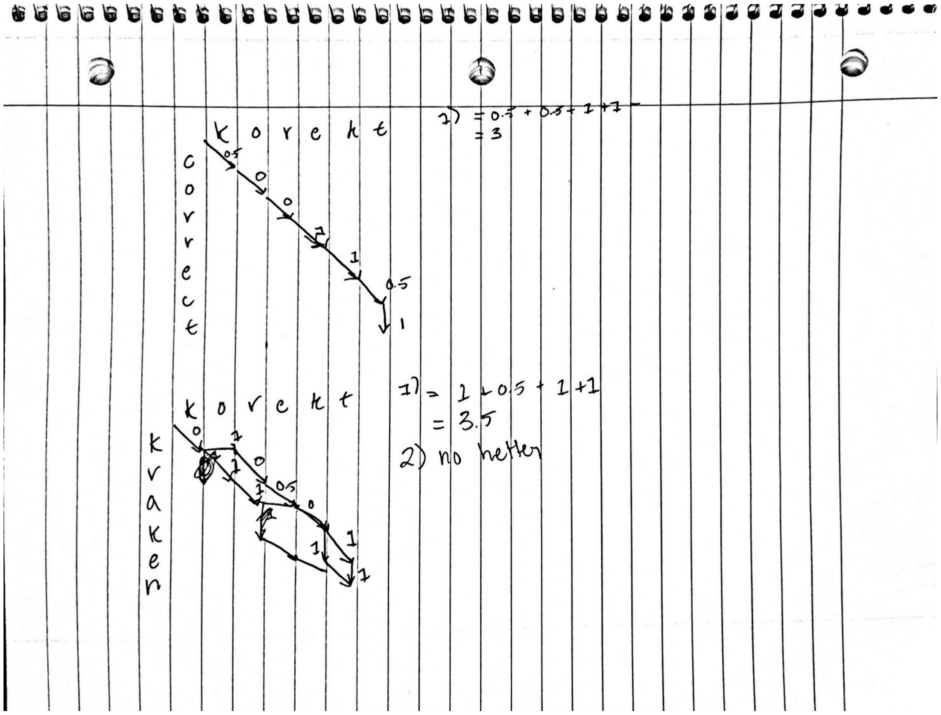
   d) Two candidates of two edit distances would be: “bred” and “beer”.

      - Here’s the code for Norvig’s minimal spelling checker in Rust :)

        https://github.com/hanleyc01/LIN-313/blob/main/spell/src/main.rs

2. The minimum edit distance from the string “korekt” → correct and kraken

   a) insert: 1; delete: 1; substitute: no change: 0, vowel → vowel: 0.5, consonant → consonant: 05, vowel → consonant: 1, consonant → vowel: 1



3. Given: “would you eat them in a box ? would you eat them with a fox ?

   not in a box . not with a fox . not in a house . not with a mouse .

   a) P(“not”) = 4/30 ~= 0.133
   b) P(“box”) = 2/30 ~= 0.067
   c) P(“box” | “a”) = 2/6 ~= 0.333
   d) P(“box” | “in”, “a”) = 2/3 ~= 0.667

4. Bayesian Spell Corrector: Given the sentence, “Probably no legendary sea monster was as horrifying as the great korekt*”, where “*” marks where the spell-checker has detected an error.

   a) Given the typo = “korekt”, and the candidates c = {“correct”, “kraken”, “carrot”}, the ranked probability of the elements of c are as follows

      - P(c1 | korekt) = P(korekt | c1) * P(c1) / P(korekt) = P(korekt | c1) / P(korekt | c1) * P(c1) + P(korekt | ~ c1) * P(~c1) = (0.25)*(126/18933) / (0.25)*(126/18933) + (0.975)*(18933-126/18933) = 0.000171
      - P(c2 | korekt) = P(korekt | c2) * P(c2) / P(korekt | c2)*P(c2) + P(korekt | ~c2)*P(~c2) = 0.0000853
      - P(c3 | korekt) = P(korekt | c3) * P(c3) / P(korekt | c3)*P(c3) + P(korekt | ~c2) * P(~c2) = 0.00004652
      - So, the ranking of P(candidate=x | type=“korekt”) is correct, kraken, carrot (as it was given).

   b) For the bigram language model of P(candidate = x | typo = k, previous_word = g), where x ∈ c = {correct, kraken, carrot}, k = kraken, and g = great

      - P(c1 | k,g) = P(k | c1)*P(c1,g) / P(k | c1)*P(c1,g) + P(k | ~c1)*P(~c1, g)
        P(k | c1) = .025
        P(k | ~c1) = .975
        P(c1, g) = 3/1013 = 0.0029615
        P(~c1, g) = 1 – P(c1,g) = 0.9970385
        So, P(c1 | k,g) = (0.25*0.0029615)/((.025*0.0029615)+(.975*0.9970385))
        Therefore, P(c1 | k,g) = 0.000761556
      - P(c2 | k,g) = P(k | c2)*P(c2,g) / P(k | c2)*P(c2, g) + P(k | ~c2)*P(~c2,g)
        P(k | c2) = 0.008
        P(k | ~c2) = 0.992
        P(c2, g) = 19/1013 = 0.01875617

$P(\sim c2, g) = 0.98124383$

So, $P(c2 \mid k,g) = (0.008*0.01875617)/((0.008*0.01875617)+(0.992*0.98124383))$

Therefore, $P(c2 \mid k,g) = 0.000154127$

- $P(c3 \mid k,g) = P(k \mid c3)*P(c3, g) / P(k \mid c3)*P(c3,g) + P(k \mid \sim c3)*P(\sim c3,g)$

  $P(k \mid c3) = 0.34$

  $P(k \mid \sim c3) = 0.66$

  $P(c3, g) = 2/1013 = 0.001974334$

  $P(\sim c3, g) = 0.998025666$

  So, $P(c3 \mid k,g) = (0.34*0.001974334)/(((0.34*0.001974334)+(0.66*0.998025666)$

  Therefore, $P(c3 \mid k,g) = 0.001018056$
- We can conclude that, given the typo “korekt”, preceded by the word “great”, the most likely word to appear are as follows: carrot, correct, kraken

5. Automatic spell checking, precision vs. recall. Search engines use automatic spelling correction on web pages *before* indexing, so that people are able to search for terms even if the terms are misspelled, compared to a phone which preforms spell-checking *after* the message has been typed.

   To begin determining whether precision or recall is more important to emphasize with these two methods of spell-checking, it would first be beneficial to lay out what we mean by precision and recall (and what are the true positives, etc. in these two cases).

   Precision is the ratio of true positives over both true and false positives, meaning that it deals in measuring the relevance of positively classified items. Recall, on the other hand, deals in the ratio between true positives over true positives and false negatives, meaning that it measures to what extent the items classified positively are *actually* relevant. To put it in other terms, precision works with both true and false positives, whereas recall works with true positives and false negatives.

   In the case of search engines beforehand spell-correction, I think that we can best identify the positives as words that are identified as misspelled, and then have a correction applied, whereas negatives are those words which were not identified as misspelled. In this case, I think that precision is much more important. Searching for terms just is the task of looking for something, and in cases were potentially something has been incorrectly identified as misspelled and thereby “corrected”, you add the potential of certain search terms being eliminated from pages, making it harder for people to search for things.

Phone spell-correction, I think, must rely much more on recall, as it is dealing directly with words typed by a user. So, if you misspell a word when sending a text message, and can’t for the life of you remember the exact way of spelling it, but the spell-checker does not even recognize that the word has been misspelled, I think this leads to a greater user dissatisfaction. For example, say I can’t remember how to spell “beautiful”, but I know that it is not “beutiful”. If “beutiful” is incorrectly identified as being spelled correctly, then I would have to spend the (probably somewhat minimal) effort to research how it’s actually spelled.