CS5012 - Language and Computation

Practical 1: Touchscreen correction

Student number: 130018883 Words: 320

The purpose of this practical was to design a text correction model that took a word, proposed corrections based on mistyping a character, calculated the probability of each possible correction using letter n-grams, and proposed the most likely correction. My model has two main functionalities: it can take a single word as input from the user and propose a correction, and perform k-fold cross validation.

Performing 10-fold cross validation on the stripped down Brown corpus has a success rate of ~0.92; i.e. no correction is proposed around 92% of the time. I used a value of $p_l = p_r = 0.2$, and $p_r' = p_l' = 0.3$. I believe the high success rate is partly due to this small probability of a mistype, and indeed when I changed the model such that $p_r = p_l = (1-p_r-p_l) = ⅓$ the success rate dropped to ~0.75.

Since I store bigram probabilities in a dict, whenever I came across a previously unseen bigram my program would encounter a Key Value error. In order to prevent this, I used Laplace smoothing in my training function, and added one instance of all possible bigrams so that the occurrence would be at least one.

One thing that this model did not take into account was mistyped words, so I added a function that would randomly change the last character by one key based on p_l and p_r described above, and testing if the proposed correction was the same as the original word. Expectedly the accuracy of the model fell, but it maintained a success rate of ~0.82, which I was still very happy with.

As a final extension I tried to generalised the model to take any value for N and use an N-gram model. I found refining some of the parts of my code a challenge, and was unable to complete it to my satisfaction. I've included my work so far in the ngram-model.py file.