Minimum Edit Distance Performance for Auto-spell Correction*

Hamid Fazli Khojir
University of Windsor
fazlikh@uwindsor.ca

1 INTRODUCTION

Finding the similarity between two string has many applications in NLP, genetics, and etc. Since string are not like real numbers and we cannot find the similarity between two string just by simple additions and subtraction, we need to define a new term, called minimum edit distance. Minimum edit distance is defined as the minimum number of modifications (add, remove, replace) that is required to convert one string to another string. Levenshtein distance, Longest common subsequence, and Hamming distance are famous edit distances. In NLP, corpus is a collection of texts that will be used as a base for NLP tasks. Birkbeck is a relatively old corpus that can be used to extract a pair of wrong and correct words, and is publicly available. Wordnet is a public English dictionary with almost 150,000 words that is available in many python libraries like nltk and PyDictionary. Considering those three elemnts, we want to use an statistical approach to find the correct spelling of a misspelled word.

2 MOTIVATION

Writing is one of the main ways of communication. However, we may write a misspelled word because of many different reasons. The main purpose of our work is to reduce these kinds of mistakes by suggesting the correct spelling of a misspelled word.

3 PROBLEM DEFINITION

Given a dictionary $\mathcal{D}$, a corpus of a pair of misspelled and correct tokens C, a pair of token $(tMisspelled, tCorrect) \in C$, top-$k$ least distance of token $tMisspelled$ is desired. Next, based on the $tCorrect$ and $x \in$ top-k, average of $s@k$ needs to be calculated for $k \in \{1, 5, 10\}$

3.1 Example

For misspelled word 'prepearing', top-10 similar words are ('rehearing', 2), ('repeating', 2), ('pampering', 3), ('preceding', 3), ('propelling', 3), ('prospering', 3), ('rearing', 3), ('releasing', 3), ('repelling', 3), ('revealing', 3). The number in parentheses is the minimum edit distance according to Levenshtein distance.

4 EXPERIMENT

4.1 Datasets

Birkbeck dataset is consisted of many different files for different purposes. After considering all of those files, we have decided to use EXAMSDAT as our corpus because of two main reasons. Firstly, this file has more misspelled words compared to other files. Secondly, we need to know the correct spelling of the misspelled word to evaluate s@k. EXAMSDAT has 12,231 pairs of words, and 6,817 pairs of unique words. We have removed duplicated words for our experiment.

4.2 Results

As Figure 1 shows, for k equal to 1, 5, and 10, s@k is 0.285, 0.473, 0.517, respectively. Calculating the edit distance between each misspelled word and all words of dictionary is time consuming. In order to handle this problem, we have used 200 different processes. Therefore, the whole process of finding similar words and calculating s@k takes almost 10 minutes when using Cedar cluster of Sharcnet. Figure 2 shows the life time of each process.

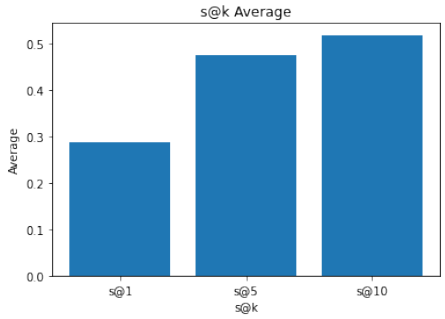

Figure 1: Success at K

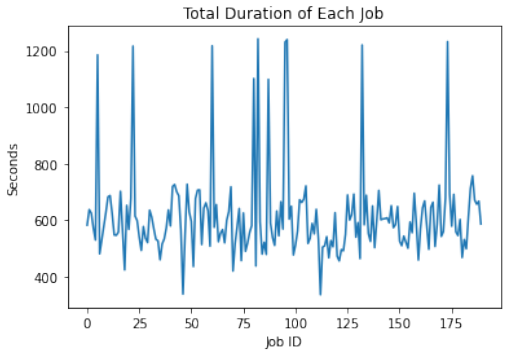

Figure 2: Solving a maze by using the wall follower algorithm

5 CONCLUSION

In this work, we successfully calculated average s@k. As we expected, the probability of success will increase by increasing k. Additionally, the difference between s@1 and s@5 is much greater than s@5 and s@10, which shows that the relation between success and k is not linear.

*https://github.com/feknall/spell-correction-minimum-edit-distance