# Lexical normalization by spelling correction methods

**Anonymised**

1. **Introduction**

lexical normalization is a translation from token with a document to canonical form. This report analyses the performance of three kinds of different means of correction methods which are (N-gram, LED, GED)[4][6]. In this experiment, we take the word from the “Misspell” with the methods mentioned above to compare with the words in dictionary to get one or more return words which is computed to be the closest lexical And then all the return word will be compared with the “correct” word to calculate the “precision” and the “recall.[5] After that Calculating the average number of the “recall” and “precision” to evaluate this method. There are 10322 words in this experiment that come from Twitter named “misspell” of which approximately 90% of them are same as “correct”, there are 370099 words in the dictionary to compare which is quite a large number.[1]

Under this circumstances，there are a lot of things we can try to see if we can improve the performance of our methods. And the performance of our method cannot be only regarded as the percentage of the “correct”, “precision” or the software running time as (complexity). In fact, we should analyze how the data result come from and what should we do to increase the performance of the data, for example, like change the parameter when using the GED and so on.

**2. Methodology**

**2.1 Global edit distance**

Global edit distance (GED) also known as Levenshtein distance is a method that to calculate the distance of two string by test the smallest changes from one word to another one which depends on the 4 parameters that “insert”, “delete”, “match”, “replace”. Give the different weight to the four parameter will have totally different answer.[2] In principle，the more weight is given to the “insert” it is more likely to return the neighbor of “delete” and “replace” and vice versa. Which set is the best to this method can be different toward different target data. But there are so many words that it is impossible to change the parameter according to the actual situation. Therefore, the only way to find the better parameter for GED is the try and get the return the percentage of the “correct”, “precision” to compare. Because a large number of data the software runs take a lot of time, there are four sets of data that we test that [match, delete, insert, replace] are [1,-1,-1,-1] ,[2,-2,-2,-1], [2,-1,--1,2],[1,-1,-1,-2] and the result are shown in the picture as follow.

**2.2 Local edit distance**

Local edit distance(LED) is quite similar to the (GED) that both compare how the two strings similar according to a pre-designed algorithm. The only difference is that the LED consider the substrings that match. That is to say, if the two objects are of similar length this algorithm is just similar to the GED, but once the length of two strings is quite different then there is a possibility that the shorter one matches the substring of the longer one to get a high score then return. Therefore it depends on if the correct answer matches the shorter one or not. In the case we can see, there is “u” in “misspell” that the correct answer is “you” which can be part of the substring match (line 8). However, there are also “masage” correspond to the “massage”(line 118)in which the substring may not get a high score because the substring only matches half of it, in this case, the GED performs better. Due to the volume of data we can not analyze the potential circumstances one by one, So there is just one set of the data that come from the configuration of Match (+1), Insert/Delete/Replace (-1) as follow.

**2.3 N-Gram distance**

N-Gram distance can be regarded as a simpler variant of Edit Distance. But according to the calculation executed by the algorithm, the more character matches in two string the higher possibility it will return.[3] That is to say, the N-Gram do not care about the order of the two strings match, but how similar they are. For example, the “emma” (line1060) when using N-Gram that can be divided in to “em”，“mm”，“ma” would match “anthema”(in dict line 14323) but this two words are not similar at all and also would not match in other means. following are the data by this method.

**3. evaluation**

In this section, we just compare all the result of the methods mentioned above separately by its accuracy, precision and running time based on the actual case.

**3.1 Case analyses**

By the actual case, we can find that most of these tweets from the social media platform Twitter are true. However, the rest tweets need lexical normalization can be divided into three groups that 1.using abbreviation like “u” represents “you”, 2. Typing mistakes like writing “message” into “masage”, 3.Using “ ’ ” lead to the misspell like using “im” to represent “I’m”. Based on these characters mentioned above[1]. There are quite some efficient ways that can actually improve the performance of spelling correction methods which will be discussed later.

**3.2 Result**

 There are 6 teams of data that get from this experiment, to get a More intuitive understanding of these data, we put it into the graphs below.

As we can see from the graph the recall of all spelling correction methods in these report performance is similar. But the precision is quite different， the GED using [2,-1,-1,2] have the best performance contrast to the GED using [1,-1,-1,-1] has the worst data. The N-gram and LED come to the middle.

AS for the running time, it depends on lots of factories. Such as Computer performance, the choice of programming language, the different method. At first, there is some attempt via python which will cost about 12 hours per process-quite a long time, so all the attempt in this report based on Java. Secondly, According to the time consuming, the GED perform best and N-Gram come close to it, however, LED perform worst cost about 3 hours.

**3.3 Result analyses**

The GED3 has the best performance due to it give high weight to the “replace” contrast to the low weight to “delete” and “insert”. It suggests that missing typing and wrong typing more letters which need “delete” and “insert” are more like to be true than typing the wrong word that needs to be replaced. It is obvious that by using the same computer with the same programming language, the running time only depends on the complexity of the algorithm so the GED and N-gram have a low complexity over the LED.

**4. Improvement**

Potential improvements considered include:

1. More attempt of all approaches to find the parameter with a higher rate of precision and recall. As well as to analyze the differences between the misspell and correct to change the parameter according to the difference and test the hypothesis to see if it is a better mean.

2. It would be better to write a function to derived the separate the word with the distance of 0 which is actually on the dictionary with the words with distance more than one. By this, it will save a lot of time comparing and also reduce the complexity of the algorithm a lot.

3. There are still some methods of spelling correction methods like the Phonetics which based on the translation in Soundex and Damerau–Levenshtein distance which is a Deformation of GED add a new method of exchange.

1. **Conclusions**

This report attempts three spelling correction methods. Find the conclusion that all of these methods are not perfect and has its own advantage and disadvantage, and can be improved by change the parameter or algorithm based on the blind test of data or according to the further analysis of the data.

**Reference**

[1]Baldwin, Timothy, Marie Catherine de Marneffe, Bo Han, Young-Bum Kim, Alan Ritter, and Wei Xu (2015) Shared Tasks of the 2015 Workshop on Noisy User-generated Text: Twitter Lexical Normalization and Named Entity Recognition. In Proceedings of the ACL 2015 Workshop on Noisy User-generated Text, Beijing, China, pp. 126–135.

[2] Heeringa, W. Measuring Dialect Pronunciation Differences using Levenshtein Distance.

[3] Kumar, P., & &rarr;, V. (2019). An Introduction to N-grams: What Are They and Why Do We Need Them? - XRDS. Retrieved from https://blog.xrds.acm.org/2017/10/introduction-n-grams-need/

[4] Soundex Converter & Calculator at Ancestor Search. (2019). Retrieved from http://www.searchforancestors.com/utility/soundex.html

[5] Beyond Accuracy: Precision and Recall. (2019). Retrieved from <https://towardsdatascience.com/beyond-accuracy-precision-and-recall-3da06bea9f6c>

[6] 6.2. Example: Edit Distance. (2019). Retrieved from https://secweb.cs.odu.edu/%7Ezeil/cs361/web/website/Lectures/styles/pages/editdistance.html