COMP90049 Project 1 Report: Waht kinda typoz do poeple mak?

Anonymous

1 Introduction

Spelling errors are not uncommon in everyday life. This report proposes several hypotheses as to how typos are made and tries to find supporting evidence by implementing a prediction system predicated on basic Global Edit Distance algorithm (GED).

There are three datasets being used in this project, namely “wiki_misspell.txt” [1], “wiki_correct.txt” [1], and “dict.txt” [2]. The first two sets refer to a collection of commonly misspelled English words and actually intended corresponding entries respectively. The last dataset servers as the dictionary for approximate string match in the prediction system. However, those datasets have been slightly modified.

Generally, typos occur when people type in excrescent letters, drop a certain letter from intended word, or simply mis-hit neighboring keys by chance. The prediction system is designed by following the concepts as well as theories with respect to approximate sting matching presented in *An Introduction to Information Retrieval* [3].

2 Hypotheses and Methodology

2.1 Hypotheses

Hypothetically, the most common ways of making typos are probably adjacent transpositions within a word. Besides, it is also assumable that people tend to make typos by mis-hitting neighboring keys by mistake.

2.2 Edit Distance

Word prediction programs have been designed based on dynamic-programming global edit distance algorithm. It is slightly different in this system since the parameters for basic operations (m, i, d, r) are (+1,-1,-1,-1) and it calculates the score for transforming a string to another. Thus, a higher score implies higher probability that the corresponding word is truly intended. However, it is more likely to get more than one candidates with same scores. In this case, the system will simply return the first candidate as the prediction word. To test the hypotheses presented above, modifications have been applied to the fundamental edit distance algorithm.

2.3 Methodology

- Keyboard effects — mis-hitting surrounding keys on standard English keyboard. Instead of simply replacing the incorrect letter within a English word and decrease the score by -1, a modified algorithm will first check all the surrounding keys of the misspelled letter on keyboard and then decide whether a penalty will be introduced accordingly. Thus, in our case, the weight of replacement can be either 1 or -1 as it will consider the neighboring incorrect letter being mis-hit by chance. The weights for matching, insertion, deletion and replacement are +1, -1, -1, -1|+1 respectively.

- Transposition of two adjacent letters. This idea derives from Damerau - Levenshtein distance [5]. Apart from the basic operations used in global edit distance, in this case, a new operation called "transposition" has been added into the system. During dynamic-programming calculation, when a single letter mismatch is detected, the system will check the current and its preceding one letter of origin and target strings and decide whether it can be corrected by swapping. If so, the mismatch will not attract any decrease in score as though it is actually a perfect match. The parameters (m, t, i, d, r) are (+1, +1, -1, -1, -1). Regardless of other three operations, parameter "m" represents match and "t" refers to transposition. They weights the same because a perfect match and a swap-to-match are expected equally in our case.

3 Evaluation

3.1 Accuracy

The system aims to output a single result as predicted best match for every single misspelled word. Thus, accuracy is adopted as the only evaluation metric. For each method mentioned above, we will calculate the fraction of correct predictions as its accuracy. Comparing the results with that of the pure edit distance algorithm, an increase in accuracy of prediction is expected so that hypotheses proposed is reasonable to some extent.

3.2 Result

The results are presented as following:

Method	Number of Misspelled	Correct Prediction	Accuracy
Levenshtein distance (0,+1,+1,+1)	4453	2444	54.88%
Modified edit distance (+1,-1,-1,-1)	4453	2901	65.15%
Keyboard Effects (+1,-1,-1,1|-1)	4453	3009	67.57%
Adjacencies transposition (+1,+1,-1,-1,-1)	4453	3174	71.28%

Table 1 Prediction accuracy

3.2.1 Levenshtein distance and Modified edit distance

According to the results above, it is noticeable that Levenshtein distance with (0,+1,+1,+1) gives the poorest prediction, with only 54.88% correct predictions. Whereas the modified global edit distance algorithm with (+1,-1,-1,-1) [5] shows greater performance, with accuracy of 65.15%. It is probably because Levenshtein distance is more focused on the number of operations needed to transform a string to another one, while the modified global edit distance expects matches more. In that case, the modified algorithm is able to correct misspelled words with more matches where Levenshtein algorithm failed. Therefore, further modifications are based on the modified GED algorithm for more accurate outputs. Typical examples are listed in the table below:

Misspelled	Correct	Levenshtein GED	Modified GED
ackward	awkward	☑ awkward	backward
ackward	bawkward	awkward	☑ backward
critized	criticized	critize	☑ criticized
beacuse	because	aeacus	☑ because
dimesnional	dimensional	digestional	☑ dimensional

Table 2 Comparison between Levenshtein GED and Modified GED

3.2.2 Keyboard-related GED

The accuracy of the system has increased to 67.57% from fundamental 65.15%, which is not significant though. The plausible hypothesis that people usually make typos by hitting neighboring wrong keys is probably not feasible with the common misspelled words derived from wikipedia. One possible reason is that the misspelled words listed in the "wiki_misspell.txt" is not primitive enough, which means that mis-hitting is still more likely to be one of the most common types of typos, but the mis-hit characters have already been corrected manually before wikipedia analyze and collect those typos since people tend to be sensitive to misspelled words phonetically.

3.2.3 Adjacencies transposition GED

Referring to table 1, the prediction system saw a relatively significant growth in accuracy by taking adjacent words relationships into account. The proportion of correct prediction words rises from 2901 (by modified GED) to 3174 out of 4453 misspelled entries. Particularly, the correction rate is considerably higher than that of Levenshtein distance algorithm, roughly increased by 30%. Some examples of swap-to-match have been listed out in the table below:

Misspelled	Correct	Modified GED	Adjacencies-transposition GED
almsot	almost	aleshot	☑ almost
archaoelogy	archaeology	archaeogeology	☑ archaeology
baout	about	backout	☑ about
bcak	back	beak	☑ back
becuase	because	bechase	☑ because

Table 3 Comparison between Adjacencies-transposition GED and Modified GED

Actually, this is not an exhaustive table of examples, many other examples can be found in the corresponding output document. The test result suggests that people usually make typos by swapping two adjacent letters.

4 Conclusion

In conclusion, test results indicate that adjacencies transposition typos are commonly made by people. The corresponding GED algorithm achieved an accuracy of 71.28% which is considerably high in practice. However, the datasets used in the test are problematic by and large. For instance, some so-called misspelled

words is actually a real word but not intended such as “hinderance”, which brings down the accuracy somehow. Besides, it is also noticeable that there are some non-word entries being considered as valid and being listed in the dictionary used in this project, which also brings down the accuracy. The algorithm for handling tied candidate corrections is not well-designed as it simply returns the first candidate as the best match.

Reference

[1] En.wikipedia.org. (2018). *Lists of common misspellings*. [online] Available at: https://en.wikipedia.org/w/index.php?title=Wikipedia: Lists_of_common_misspellings&oldid=813410985 [Accessed 27 Aug. 2018].

[2] GitHub. (2018). *dwyl/english-words*. [online] Available at: https://github.com/dwyl/english-words [Accessed 27 Aug. 2018].

[3] Manning, C., Raghavan, P. and Schütze, H. (2008). *An Introduction to Information Retrieval*. New York: Cambridge University Press, pp.56 - 65.

[4] J. Nicholson, "Approximate String Search and Matching", the University of Melbourne, Vic, 2018.

[5] F. Damerau, "A technique for computer detection and correction of spelling errors", *Communications of the ACM*, vol. 7, no. 3, pp. 171-176, 1964.