Waht kinda typoz do poeple mak?
COMP90049 Project 1 Report

1 Introduction

What kind of typos do people make is a wide question. Some existing solutions, like spelling checker or auto spelling corrector, help people make less misspell. On another hand, these solutions need to know how people make misspells, to improve their performance. For spelling correctors, a suitable algorithm could increase the correctness of their recognization and prediction on the words. This report implements a misspelling predictor based on the basic global edit distance algorithm to correct misspelling, and discuss the performance of those praticular algorithms on those praticular corpuses.

2 Data

There two set of corpuses used for this project: the wiki misspell and the birkbeck misspell. The wiki misspell list is from Wikipedia contributors (nd), there are 4453 tokens that have been identified as common errors madde by Wikipedia editors. Corresponding with misspell list, wiki correct list is a list of the truly intended spellings. Another set of corpuses is from Roger Mitton (nd), which contains 34683 misspellings words, comprising the "Birkbeck spelling error corpus". It has a big difference with the wiki misspell that this list is a machine-readble transcription of hand written by schoolchildren, university students, and adult literacy student. In additional, the dictionary is from (dwyl, nd) which contains approximately 370K English entries. The implementation used it as a referenced correct words during comparision.

3 Methodology

Python is choosen as the language we implemented on, due to the wide range of packages, and there are several avariable algorithms able to complete the task.

3.1 Globla Edit Distance

The global edit distance(GED) is the main algorithm in the implementations. Package editdistance in Python preforms well and fast which is based on the rule of Levenshtein distance. It calculates the "distance" between the misspelling word and the word from the dictionary by the numbers of character edit. In this implementation, these parameters are set to $(0, 1, 1, 1)$ for $match, insert, delete, replace$. The one with the loest distance will be provide as the correct word. The naive version of GED is that only provides one prediction to the user. The predicted word has lowest distance with the word in dictionary. However it is not considered the situation when several words in dictionary have the same lowest distance with the misspelling word. The naive GED simply outputs the one appear earlier in the dictionary. The naive version of GED get improved that rather than storing the possible word as a single candidate, to store them in a list. The improved GED is able to provide multiple possible options in a row. However, multiple responce is hard to get familiar by people. The second algorithm is used to refine the outputs.

3.2 N-gram

By testing the usabilities of the packages on Python, it is found that N-gram has the lowest efficiency. So it is used as a filter to refine the predictions from the GED. Since the less amount of responses from GED led the less amount of input for Ngram, then it could executes in a reasonal time. N-gram splits the tokens to substring with size of N. The number of substrings from two different tokens indicates the similarity of the two tokens. In this implementation, the parameter N is set to 2 and the distance between two tokens is calculated as

$$\frac{Number\ of\ same\ substrings}{Total\ number\ of\ unrepeated\ substrings}$$

4 Evaluation Metrics

Since the output number from different implementations are slight different. There are different evaluation metrics for them. Thoughout the report, accuracy, precision, and recall will be considered separately to evaluate each implementaion system.

- *Accuracy*: For only simgle ouput system, the fraction of correct output that the system responce to.

$$Accuracy = \frac{Number\ of\ correct\ predictions}{Total\ number\ of\ words}$$

- *Precision*: For multiple outputs system, the fraction of correct response among attempted responses.

$$Precision = \frac{Number\ of\ correct\ predictions}{Total\ number\ of\ predictions}$$

- *Recall*: For multiple outputs system, the fraction of word with a correct response.

$$Recall = \frac{Number\ of\ words\ with\ correct\ response}{Total\ number\ of\ words}$$

5 Result

The result are summarised in Table 1.

Method	Accu / Recl	Prec
Single Output GED	0.549	*
Multi Outputs GED	0.797	0.263
GED + Ngram	0.717	0.600

Table 1: Result for Wiki misspell

6 Section

Version one might not suit to all users. A large amount of candidates cannot work for the general users who write or type on dairly used words. However for some long jargon, ???generalling it has less preselelctions???

7 Conclusions

Concluding text.

References

dwyl. n.d. English words. https://github.com/dwyl/english-words.

Oxford Text Archive Roger Mitton. n.d. birkbecks. https://www.dcs.bbk.ac.uk/ ROGER/corpora.html.

Wikipedia contributors. n.d. Wikipedia:Lists of common misspellings. In *Wikipedia, The Free Encyclopedia.* https://en.wikipedia.org/w/index.php?title=Wikipedia:Lists_of_common_misspellings&oldid=813410985.