COMP90049 Project 1 Report: what wired spellings! Are people crazy?

1 Introduction

The goal of this report is to analyse and evaluate spelling correction based on Approximate String Search algorithms. In this report, the performance of Global Edit Distance (short GED), Local Edit Distance (short LED), N-Gram Distance and Soundex will be detailly evaluated and analysed firstly. After that, an improved algorithm will be given and discussed. Finally, the report will give further optimisation methods.

2 Dataset

There are 716 misspelling words and 716 corresponding correct words, which are fetched from UrbanDictionary that are recognised as being misspelled automatically. Those lists are provided by Saphra and Lopez (2016). The dictionary is compiled from multiple sources, comprising 393954 tokens of English language.

3 Evaluation Metrics

The following definitions will be evaluated in each algorithm throughout this report (Jeremy, Justin, Karin and Rao 2018, p.149-154):

- Accuracy: The proportion that the misspelling words are corrected by each algorithm, comparing to the correct word list.
- Recall: The proportion with a correct response.
- Precision: The proportion of correct predictions among all attempted responses
- Time: The running time of an algorithm.

4 Methodology

4.1 Global Edit Distance

The Global Edit Distance uses Levenshtein parameter as the scoring scheme. The source code is fetched from (Jeremy, Justin, Karin and Rao 2018, p.62) and the raw code has been slightly modified. The running result is shown on Table 1.

Recall	42.46%
Accuracy	14.67%
Precision	4.87%
Time	164s

Table 1: Result of GED for 716 misspelling words

4.2 Local Edit Distance

The second approach is Local Edit Distance, which is similar as the first one. The library is sourced from Debatty (2017) and the result is displayed in Table 2.

Recall	28.91%
Accuracy	14.94%
Precision	7.54%
Time	196s

Table 2: Result of LED for 716 misspelling words

4.3 Two-Gram Distance

We choose 2-Gram as the third approach which scores substrings. The testing result is shown in Table 3.

Recall	25.14%
Accuracy	13.83%
Precision	7.98%
Time	170s

Table 3: Result of Two-Gram for 716 misspelling words

4.4 Phonetic Algorithm

The above algorithms are based on string matching, while Soundex matches similarity of phonetics. This approach is referenced a library from Apache Software (2017) and the running result is listed in Table 4.

Recall	74.44%
Accuracy	0.42%
Precision	0.48%
Time	100s

Table 4: Result of Soundex for 716 misspelling words

5 Evaluation

The figure below illustrates three aspects of performance of those four approaches. It is clear that Soundex performs the best on recall, which is nearly doubled as GED – the second highest recall. However, the precision and accuracy are relatively low. This may because large proportion of misspelling words have similar pronunciation. The overall performance GED, LED and Two-Gram are very close, though GED has higher recall and smaller precision.

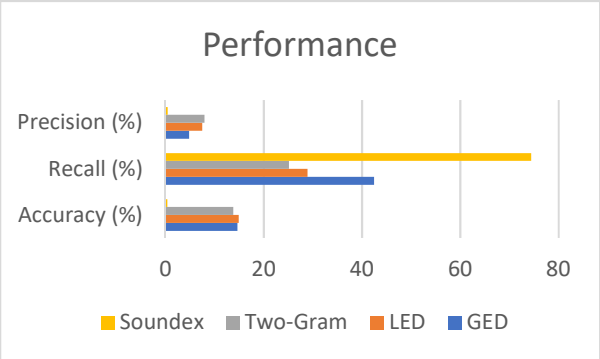

Figure 1: Algorithm Performance Comparison

6 Analysis

The following paragraphs describe what affect the performance.

6.1 Words not in the dictionary

If the word in correct list contains no the same word in the dictionary, it is considered as not in the dictionary. From the statistics of the program, there are about 17.04% words of correct list which are not shown in the dictionary. This means some of the word will be always wrong regardless of approach and the algorithm will find a most similar word appearing in the dictionary. There are some typical cases that the words are not in the dictionary.

6.1.1 Dictionary list is not recorded

For example, the word 'mamacita' has meaning in the real world, but the dictionary does not contain this word.

6.1.2 Acronym

Acronyms are used frequently in case of shorting long words and text chatting. For example, the word 'dece' may be an acronym of Digital Entertainment Content Ecosystem, but it is not listed in the dictionary.

6.2 Words in the dictionary

For the remaining 82.96% words which are in the dictionary, it is not guaranteed that the word will be found correctly in the correct list. There are 4 frequent cases that affect the performance.

6.2.1 Recorded Misspelling

The most common case is wrong spelling but very close to the correct word. For example, the corresponding correct word of 'amazimg' and 'amazong' is 'amazing', which only needs few edits to be corrected. Normally, the best approach is approximate string matching instead of phonetic method, though phonetic method will perform good on recall in this case.

6.2.2 Correct in dictionary

This is to say, the word is correct in the dictionary, but it is not the right one in the correct list. For example, 'aeroplane' is a correct word in the dictionary and it will also get best marks on either algorithm because they are equivalent. However, the correct word, 'airplane', has worse score. Therefore, in this case, neither approach will find the correct one.

6.2.3 Similar Pronunciation

Some of words are largely different on spelling, but the pronunciation is very similar. For an instance, 'aye' and 'eh' have only character 'e' in common according to the spelling, but the pronunciation is similar. In this case, phonetic algorithms will have more chance to find the right answer.

6.2.4 Short Form or Variant Spelling

There are many short form or variant spelling in the 'misspelling' list, these words are commonly used on the Internet chatting, such as 'b4' (before), 'b01' (boy) and 'gr8t' (great). This is also more likely to be corrected by using phonetic methods because the spelling of words is largely different, while the

pronunciation is similar. However, all of algorithms above has not set up parameter of numbers, which causes those words are difficult to be recognised.

7 Modified Method

From the analysis above, there are many optimisations which could be applied.

Firstly, the new algorithm can select a list words using Soundex first, because Soundex has best performance in recall. The high recall means the correct word is likely shown in the Soundex attempting list. Then a combination of Local Edit Distance and Global Edit Distance can be used to find a best match string from that list because of higher accuracy.

Secondly, we can assume that the words in the 'misspelling' list are always incorrect. This means if the algorithms found a word in the dictionary that is identical to the word in the misspelling list, the word will be skipped, therefore the algorithm will turn to compute the next word.

Thirdly, the dictionary can be optimised by adding missed words from correct list. As the report mentioned before, there are about 17.04% of words in correct list are not shown in the dictionary, so those missing words can be added directly into dictionary to increase accuracy and recall.

Table 5 shows the performance of the optimised algorithm, which implements those three recommended optimisations. As can be seen from the table, all aspects of the performance are improved.

Recall	40.01%
Accuracy	28.07%
Precision	19.96%
Time	72s

Table 5: Result of Optimised Method for 716 misspelling words

To further optimise the correcting algorithm, a new converting dictionary can be built. In this dictionary, frequently used tokens and its full spelling should be both referenced. For an instance, the number '4' can be converted as 'four', 'for' or 'fore' and those conversions have been added in to the new dictionary. Assuming we are looking for correcting form of 'b4', the algorithm will try each combination of 'b' and '4' then use optimised method implemented before to find the best matching score.

Furthermore, the following optimisation on converting dictionary could be considered:

- Adding proper noun
- Adding short form

8 Conclusion

This report evaluates the performance of Global Edit Distance, Local Edit Distance, N-Gram and Soundex in correcting misspelling words and analyses reasons that may affect the performance. Then three modifications have been made on optimised method to improve the performance according to the analysis and the feasibility has been proved.

References

Naomi Saphra and Adam Lopez. 2016. *Evaluating Informal-Domain Word Representations with UrbanDictionary.* In *Proceedings of the 1st Workshop on Evaluating Vector-Space Representations for NLP*, Berlin, Germany. pp. 94–98.

Zobel, Justin and Philip Dart. 1996. *Phonetic String Matching: Lessons from Information Retrieval*. In *Proceedings of the Eighteenth International ACM SIGIR Conference on Research and Development in Information Retrieval*. Zurich, Switzerland. pp. 166–173.

Jeremy Nicholson, Justin Zobel, Karin Verspoor and Rao Kotagiri. 2018. *Approximate String Search and Matching.* Available from: http://lms.unimelb.edu.au. [04 April 2018].

Apache Software. 2016. *Apache Lucene*. Available from: https://lucene.apache.org/core/. [04 April 2018].

Apache Software. 2017. *Apache Commons*. Available from: https://commons.apache.org. [04 April 2018].

Thibault Debatty. 2017. *Java String Similarity*. Available from: https://github.com/tdebatty/java-string-similarity. [04 April 2018].