COMP90049 Project 1 Report: waht wierd spelings! aer peaple carzy? (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 GED), 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.

1. 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.

1. Evaluation Metrics

The following definitions will be evaluated in each algorithm throughout this report:

* Accuracy: The proportion that the misspelling words are corrected by each algorithm, comparing to the correct word list.
* Time: The running time of an algorithm.
* Average Prediction: The number of times that each word predicts the correct word.
* Precision: Proportion of correct predictions among all attempts

Few assumptions have applied in this testing:

* All testings are using both the same misspelling word list dictionary.
* The running environment (time) is based on Java.

1. Methodology
   1. Global Edit Distance

The Global Edit Distance uses Levenshtein parameter with the score that match worth 0 mark and delete/replace/add worth 1 mark. The source code is fetched from Apache Software (2016) and the raw code has been slightly modified. The running result is shown on Table 1.

|  |  |
| --- | --- |
| Average Predictions | 7.03 |
| Accuracy | 14.67% |
| Precision | 1.7% |
| Time | 47s |

Table 1: Result of GED for 716 misspelling words

* 1. Local Edit Distance

The second approach is Local Edit Distance, which matches the best substrings. The library is sourced from Debatty (2017) and the result is displayed in Table 2.

|  |  |
| --- | --- |
| Average Predictions | 217813 |
| Accuracy | 15.5% |
| Precision | % |
| Time | 83s |

Table 2: Result of LED for 716 misspelling words

* 1. N-Gram Distance

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

|  |  |
| --- | --- |
| Average Predictions | 365048 |
| Accuracy | 14.66% |
| Precision | % |
| Time | 126s |

Table 3: Result of 2-Gram for 716 misspelling words

* 1. 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.

|  |  |
| --- | --- |
| Average Predictions | 393954 |
| Accuracy | 0.42% |
| Precision | % |
| Time | 72s |

Table 4: Result of Soundex for 716 misspelling words

1. Evaluation

The Global Edit Distance has best performance in both average prediction and running time, while Soundex computes every word in dictionary in each prediction. Though the Local Edit Distance has highest accuracy, the accuracy of Global Edit Distance, Local Edit Distance and N-Gram (2-Gram) Distance are very close. However, Soundex algorithm perform worst in term of accuracy. This may because many words have similar pronunciation, which may have better score than the correct one, some of situations will be discussed detailly later.

1. Analysis
   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:

1. Misspelling

Some of the words are simply misspelled, such as ‘ballsack’ and ‘mansex’.

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.

1. Acronym

Acronyms are used frequently in case of shorting long words. For example, the word ‘dece’ does not has actual meaning itself, which is an acronym of Digital Entertainment Content Ecosystem. But it is not recorded in the dictionary.

* 1. 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.

1. Misspelling

The most common case is wrong spelling but very close to the correct word. This only needs few edits to be corrected, so the best approach is approximate string matching rather than phonetic method.

1. 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 (zero) on either algorithm simply because they are equivalent. However, the real word listed in the ‘correct’ is ‘airplane’, which has worse score.

1. Similar Pronunciation

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

1. 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, Soundex algorithm does not set up parameter of numbers, which causes those words are hard to be recognised.

1. Modified Method

It is obviously that Local Edit Distance has best performance in correcting words among two approximate string matching methods in terms of accuracy, average prediction and run time, while Soundex performs better on looking for similar pronunciation as the Section 6 described.

Firstly, the optimised method can be a combination of both Local Edit Distance and Soundex algorithms. Specifically, the new algorithm will firstly use Local Edit Distance to find a best matching word with a score, if the score is lower than a pre-set threshold, then use this word directly as target word, otherwise the algorithm will turn to use Soundex.

Secondly, we can assume that the words in the ‘misspelling’ list are 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 missing words. 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.

1. Improvement

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 firstly try each combination of ‘b’ and ‘4’, which uses optimised method in Section 7 to find the best matching score. This would avoid phonetic guessing thus the accuracy will be increased significantly.

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

* Adding proper noun
* Adding short form

1. Conclusions

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.

Apache Software. 2016. Apache lucene. <https://lucene.apache.org/core/>.

Apache Software. 2017. Apache commons. <https://commons.apache.org>.

Thibault Debatty. 2017. <https://github.com/tdebatty/java-string-similarity>.