COMP90049 Project 1 –

Waht kinda typoz do poeple mak?

# 1 Introduction

Approximate string search algorithms are broadly applied as spelling correction methods to resolve typographical errors in languages which are omnipresent. In this project, we are working on lists of misspelt words to get them automatically corrected using approximate string search approaches.

The report discusses several implementations of these methods along with the evaluation towards them according to related experiments. The main datasets involved in the project include:

***wiki\_misspell.txt*** and ***wiki\_correct.txt***

Originally from Wikipedia: Lists of common misspellings [1], both consisting of 4453 words in identical alphabetical sequences. (Hereinafter referred to as Wiki Data)

***birkbeck\_misspell.txt*** and ***birkbeck\_correct.txt***

Originally from Birkbeck spelling error corpus in University of Oxford Text Archive [6], containing 34683 entries. (Hereinafter referred to as Birkbeck Data)

***dict.txt***

A dictionary of the study comprising 370099 English words in alphabetical order, each existing only once. (Hereinafter referred to as Dictionary)

# 2 Overview of Methods

There are numerous methodologies developed regarding to approximate string search. Edit distance (EDIT, also known as Damerau-Levenshtein distance) counts the minimum number of edits required to transform one string into the other. It can be further classified as *global* (GED) and *Local* (LED) [2] The calculation of N-Gram Distance also aims to choose among the similarities between strings for the optimal result of correction. [3] There are also Phonetic String Matching algorithms (e.g. Soundex) which can detect phonetic spelling mistakes caused by similar pronunciations between two words. [4]

## 2.1 Global edit distance

In this project we choose Levenshtein distance as the implementation, which is a most known global edit distance. The standard scores specified for Levenshtein distance are: Match (0), Insert / Delete / Replace (+1). For each misspelt character in the list**,** we calculate its Levenshtein distances with every word in the ***Dictionary*** and choose the entry with minimum distance as the optimal match.



## 2.2 N-gram distance

Bigram distance is applied in this project, which means substrings are of length 2. Same as Section 2.1, we calculate bigram distances between a misspelt word with every entry in ***Dictionary***, and finally pick the one with minimum bigram distance.



## 2.3 Soundex

Particularly implemented to test Birkbeck Data, which contains phonetic typographical errors. Firstly, we translate each entry in misspelt list and dictionary respectively into the 4-digit code defined by Soundex. Then for each word in misspelt list, pick entries as a list from the translated dictionary with the identical code. Therefore, further select words with the minimal global edit distance to the misspelt word to narrow the range of candidates.



Phonetic errors are comparably hard to identify, mainly because some misspelt words might be legal and can be sought from the dictionary, which means that particular word will definitely be picked as the optimal candidate.

Moreover, our datasets are simple so that no contextual information are provided for implication.

# 3 Hypotheses

According to the initial investigations over the typos in Wiki Data, the typographical errors can be classified into basically four types, which are:

* Insertion, with unnecessary character(s) inserted or duplicated. E.g. abandonned (abandoned)
* Omission, with necessary character(s) omitted. E.g. abilties (abilities)
* Transposition, with two characters swapped unexpectedly. E.g. abritrary (arbitrary)
* Substitution, with some character(s) wrongly replaced by other character(s). E.g. acadamy (academy)

The observation to Birkbeck Data indicates that it contains a considerable portion of phonetic error apart from the four types above.

* Phonetic errors, usually with words whose pronunciations are quite similar. E.g. board (bored)

The experiments are designed based on the hypothetic errors above.

As the testing data are solely single words without a linguistic context, the entire prediction is based on the similarities between two words with no additional information. Therefore, the result is not as satisfactory as those in terms of paragraphs or articles.

Occasionally there are situations where multiple predictions with same distances. As there are no extra information provided by Dictionary (frequency, context, etc.), here we just keep them as a list of candidates to be evaluated by the precision metrics.

# 4 Experiments and Results

Based on the characteristics of the test data, we apply Levenshtein distance and N-Gram distance to Wiki Data because of the absence of phonetic errors, whereas we bring Soundex to Birkbeck Data to evaluate the performance of phonetic matching techniques.

Concerning there can be multiple words in the dictionary with identical distances to one misspelt word, the project uses **precision** and **recall** in terms of evaluation metrics of the approximate string search algorithms.

Wiki

====== Levenshtein Result ======

Precision 0.3984

Recall 0.8739

Wiki

====== N-Gram Result ======

Precision 0.5733

Recall 0.7748

Birkbeck



====== Levenshtein Result ======

Precision 0.0902

Recall 0.4591

====== Soundex Result ======

Precision 0.1571

Recall 0.4296

# 5 Critical Analysis

A discussion of how the results provide evidence supporting the presence/absence of theoretical

types of typographical errors;

# 6 Conclusion

The results largely depend on the quality of the test data and dictionary.

# 7 References

[1] 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

[6] Mitton, Roger (1980) Birkbeck spelling error corpus. In University of Oxford Text Archive,

http://ota.ox.ac.uk/headers/0643.xml

[2] Levenshtein, Vladimir I. (1966). “Binary codes capable of correcting deletions, insertions, and reversals”. Soviet Physics Doklady 10 (8): 707710.

[3] Kondrak, Grzegorz (2005). “N-Gram Similarity and Distance”. In Proceedings of the 12th international conference on String Processing and Information Retrieval (SPIRE’05), pp. 115-126, Buenos Aires, Argentina.

[4] Zobel, Justin and Dart, Philip (1996). “Phonetic String Matching: Lessons from Information Retrieval”. In Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR’96), pp. 166-172, New York, USA.

[5] B. John Oommen,

String alignment with substitution, insertion, deletion, squashing, and expansion operations, Information Sciences, Volume 83, Issues 1–2, 1995, Pages 89-107, ISSN 0020-0255