COMP90049 Project 1 Report: Spelling Correction

1. Introduction

Typographical error includes errors due to mechanical failure or slips of the hand or finger such as “abandonned” and “abandoned”, but excludes errors of ignorance or the flip-flopping of words such as “than” and “then”. Fat finger, which refers to an unwanted secondary action when typing, is also a common reason that causes the typographical errors. For example, “buckled” and “bucked”. Furthermore, similar pronunciation is also a common reason to cause typographical error. Spelling Correction is the process of detecting misspelled words and trying to determine the most likely correctly spelled word which was meant.

The aim of this report is implement a Global Edit Distance (GED), N-gram and phonetic based prediction system to find the most similar word with a misspelled word. This is an attempt to understand the typological error types and how approximate string matching methods deal with those typological errors.

2. Dataset

The dataset involves a list of common misspellings, which contains 4453 tokens occur at least once a year, made by Wikipedia editors and a list of corresponding correct spellings. Another dataset is a list of English dictionaries which contain 37099 tokens is used for approximate string search. In this paper, the dataset of misspelling and correct spellings is referred to as *wiki_misspell* and *wiki_correct*. Also, dictionary dataset is denoted as *dict*!

3. Evaluation Metrics

Throughout this report, the following terms will be used to evaluate each approximate string matching system:

	Correct	Incorrect
Searched	A	B
Non-searched	C	D

Table 1: precision/recall table

- Precision: For systems that predict the token that exactly matches with the corresponding token in *wiki_correct* among all retrieved instances.

$$\mathcal{P}(precision) = \frac{A}{A+B}$$

- Recall: For systems that the words are correctly retrieved among all the correct words. Note that total words in this report are 4453 that will not variate with different systems.

$$\mathcal{P}(recall) = \frac{A}{Total\ \ words}$$

The runtime is also shown, as in different situations may call for more efficient results at cost of recall. However, in this report, evaluation metric takes no account of time since different computers may lead to different time result and it is shown here just in case of real-world deployment.

4. Global Edit Distance

4.1 Basic

The GED here used Levenshtein distance parameters of (m, i, d, r) = (0, 1, 1, 1). The GED code is written in Python and implemented with *editdistance* package. The results are shown in Table 2:

Precision	0.26
Recall	0.79
Avg. Predictions	3.03
Max. Predictions	103
Time/s	1.27

Table 2: Results of GED for 4453 misspelled words.

4.2 Analysis

The Levenshtein distance is effective in situations where typographical errors are caused by slips of hand, i.e. a missing letter, or an extra letter or a wrong letter in the misspelled words. A few explanations are shown in Table 3. However, under this system, misspelled errors, including two missing letters in a word, originally correct spelling word and wrong letter order, cannot be detected or predicted successfully.

Misspelled	Prediction	Correct	Right/Wrong?
withh	witch	With	✗
	with		□
	withe		✗
	withy		✗
phenonmena	phenomena	phenomena	□
unviersity	university	university	□
	unverity		✗
terriory	territory	territory	□
stopry	stoory	story	✗

	story		□
	stoury		✕
congradulations	congradulations	congratulations	□
sophicated	spicated	sophisticated	✕
puting	puting	putting	✕
abondoned	abandoned	abandoned	□
docrines	dourines	doctrines	✕
	doctrines		□

Table 3: Examples of GED

5. N-gram

An N-gram is an N-character slice of a longer string. For example. The word "apple" would be composed of the following n-grams:

bi-gram: _a, ap, pp, pl, le, e_

NLTK library written in python is used for testing N-gram.

5.1 Methodology

In our system, we use N-grams of several different lengths simultaneously. Then, we calculate the jaccard distance by applying n-gram. The system chooses the word that has the smallest jaccard distance as response.

5.1.1 Bigram

Applying the 2-gram system, typographical errors such as flip hands and fat fingers are corrected successfully. A few "examples are shown in table 4 below. In table 4, words like "withh", "abondoned", "congradulations" are flip hand errors with extra letter or wrong letter. However, errors like "phenonmena" which is caused by pronunciation is hard to be successfully predicted by 2-gram.

Misspelled	Prediction	Correct	Right/Wrong?
withh	with	with	□
phenonmena	phenomenona	phenomena	✕
unviersity	enviers	university	✕
	university		□
terriory	territory	territory	□
stopry	stop	story	✕
congradulations	congratulations	congratulations	□

sophicated	sophisticated	sophisticated	□
puting	puting	putting	✕
abondoned	abandoned	abandoned	□
docrines	endocrines	doctrines	✕

Table 4: Examples of 2-gram

5.2 Results

The results of three N-gram are shown below.

Precision	0.061
Recall	0.696
Avg. Predictions	11.34
Max. Predictions	92
Time/s	2.23

Table 5: Results of bigram

6. Phonetic Algorithm

A phonetic algorithm is an algorithm for indexing of words by their pronunciation. Some typographical errors are caused by similar pronunciation words such as “phenomena” and “phenomenon”. The aim, in this paper, was to prove the pronunciation caused typographical errors.

6.1. Soundex

The Soundex algorithm is used to group similar sounding letters together and assign each group a numerical number. The numerical indexing for encoding text results in retrieving a list of words that are pronounced similarly with very little variation in their homophones. In this report, soundex in scikit-learn machine learning library written in Python is used for analysis.

6.1.1 Results

The results are shown below in the table 6.

Precision	0.012
Recall	0.79
Avg. Predictions	21
Max. Predictions	323
Time/s	3.03

Table 6: results of soundex

6.1.2 Analysis

This algorithm fixes typographical errors that edit distance and n-gram can't especially for the errors caused by similar pronunciations. A few examples are shown in table 7. For example, 'puting' cannot be predicted successfully by GED and 2-gram systems, but Soundex successfully find the correct word 'putting'. 'puting' is a typographical error caused by similar pronunciation or flip hand error(missing a letter). However, flip hand errors that lead the word has different pronunciation than the correct word cannot be predicted successfully by Soundex. The recall is higher rather than n-gram as we can see from table 6. In brief, Soundex proves the existence of typographical errors, caused by pronunciation, and could predict correct word successfully.

Misspelled	Prediction	Correct	Right/Wrong?
withh	wad	With	✗
	whit		✗
	with		□
	wot		✗
	…		…
phenonmena	pannonian	phenomena	✗
	peine		✗
	phenomena		□
	piewoman		✗
	…		…
unviersity	unapproached	university	✗
	universite		✗
	university		□
	unverity		✗
	…		…
terriory	taur	territory	✗
	twere		✗
	…		…
stopry	stuffer	story	✗
	…		…
congradulations	congratulating	congratulations	✗
	congradulations		□
sophicated	spicated	sophisticated	✗

puting	puting	putting	✕
	putting		□
abondoned	abandoned	abandoned	□
docrines	dourines	doctrines	✕
	decorums		✕

Table 7: Examples of Soundex

7. Conclusion

Overall, through careful analysis of the effects of GED, N-gram and Soundex prediction algorithm, as well as the typographical errors that each prediction system could deal with, GED and Soundex have higher recall rather than 2-gram. Though GED and Soundex both have 0.79 recall, GED has 24.8% higher precision and 19 fewer average attempted responses than Soundex. Therefore, in the aspect of precision, GED is much more accurate and efficient than Soundex. For the typographical errors caused by slip of hands or fat finger(a missing letter, an extra letter or a wrong letter in a given word), GED is good at predicting the correct word. For the typographical errors that caused by pronunciation, Soundex would be a better choice to find the most similar word.

Reference

Bhatti, Z., Waqas, A., & Ismaili, I. A. (2014, February). Phonetic based SoundEx & ShapeEx algorithm for Sindhi Spell Checker System. *n.d.*

contributors, W. (n.d.). *Wikipedia: Phonetic algorithm*. Retrieved from Wikipedia, The Free Encyclopedia: https://en.wikipedia.org/wiki/Phonetic_algorithm

contributors, W. (n.d.). *Wikipedia: Typographical error*. Retrieved from Wikipedia, The Free Encyclopedia: https://en.wikipedia.org/wiki/Typographical_error

contributors, W. (n.d.). *Wikipedia:Lists of common misspellings*. Retrieved from Wikipedia, The Free Encyclopedia: https://en.wikipedia.org/wiki/Wikipedia:Lists_of_common_misspellings

Goitte, C., & Gaussier, E. (2005). *A Probabilistic Interpretation of Precision, Recall and F-Score, with Implication for Evaluation.* Texas: Springer.

Hollman*, J., Kann*, V., & Domeij*, R. (1998, May 7th). Implementation aspects and applications of a spelling correction algorithm. *Royal Institute of Technology*.

Navarro, G. (2011). A Guided Tour to Approximate String Matching. *n.d.*

Peterson, J. L. (26 May 2005). *Computer programs for spelling correction: An experiment in program design.* Springer.

Wordnet. (2007, November 12). Wordnet definition. *Princeton University*.

Zobel, J., & Dart, P. (n.d.). Phonetic String Matching: Lessons from Information Retrieval. *n.d.*