COMP90049 Knowledge Technologies Project 1
Lexical Normalisation of Tweets

Anonymous

1 Introduction

Twitter is an online social networking service where users post and interact with messages named as "tweets"[1]. Sometimes tweets are formatted in an informal way with many abbreviations, typos and misspelt words.
This project aims to reformat this kind of words in tweets data by predicting best matches words for each of the input tokens 'misspell.txt' according to a dictionary. The project starts with a basic Global Edit Distance implementation. From the knowledge gained from analysis of the result, some changes are made to the GED. And Soundex is used with GED together to test the variety of the result.

2 Related Work

Compared with formal writing, texts like SMS or "Tweets" contain various of non-standard words (Sproat et al., 2001). Previous work on text normalization includes spelling correction, Noisy Channel Model and Machine Translation (Xue et al., 2011). Aw et al. (2006) established a Phrase-based Statistical Model, which translated SMS language to the English language as a preprocessing step for Machine Translation. Cook and Stevenson (2009) proposed an Unsupervised Noisy Channel Model, which comprised a word model based on word formation process and the frequency of word formation types. Han and Baldwin (2011) indicated most non-standard words in "Tweets" and SMS were based on morphophonemic variation. They proposed using an "OOV" (Out of Vocabulary) whitelist to improve detection of ill-formed tokens.

3 Dataset

The dataset used for this project is from Baldwin et al. (2015) which contains 10322 words from tweets and some of them may be misspelt. Other used datasets are a dictionary named 'dict.txt' which holds 370099 words and a file named 'correct.txt' which holds correct spelt words for analysis.
After analysis of the input tokens with exact string search, it can be divided into three parts (Table 1):

Parts	Numbers of words
In the dictionary	8376
Not in the dictionary	1632
Includes typos	314

Table 1: Three parts of the tokens

The first part includes all words which can be exactly found in the dictionary. In this situation, they can be considered as the best predictions of themselves, due to that they are highly probably words spelt correctly.
The second part includes all words composed of letters but not exist in the dictionary directly. These tokens are probably misspelt and will be processed with approximate string search methods to find the best predictions.
The last part includes all words contains special symbols or numbers due to the reason that there is no symbols or numbers in the dictionary. Tokens in this part are difficult to modify according to the dictionary. They will be treated as the best predictions of themselves temporarily.
Part 1 and Part 3 will be considered as 'direct predictions' in this project.

4 Methodologies

Start with a basic GED algorithm then looking for some changes on GED. Combine with a phonetic matching at the end.

4.1 Global Edit Distance (GED)

In this project, NEEDLEMAN-WUNSH Algorithm is used to calculate GED. It measures

[1] https://en.wikipedia.org/wiki/Twitter

with two strings and returns the minimum number of operations to transform from one string to another. The program begins with parameters $[m, i, d, r] = [+1, -1, -1, -1]$, and returns the word which meets the max number of GED while searching in the dictionary for each word in the queries.

4.2 GED with changes

Two changes including catching more than one result when calculating GED and changing the parameters implement here on the foundation of basic GED.

4.2.1 Return mutiple words

The number of words which reach the max GED in the dictionary may be more than one while traversing the queries. It may increase the recall by returning all these words instead of returning one, due to that these words are all highly correlated to the input.

4.2.2 Change the parameters

Based on multiple words returned, the parameters used in the algorithm can be changed to adjust the weights of different operations. Samples:

$$[m, i, d, r] = [+2, -1, -1, -1]$$
$$[m, i, d, r] = [+1, -2, -1, -1]$$
$$[m, i, d, r] = [+1, -1, -2, -1]$$
$$[m, i, d, r] = [+1, -1, -1, -2]$$

4.3 Soundex

"Tweets" usually contains some words come from euphony, which means some words are different in spelling but same in pronunciations. Phonetic matching methods may help in this situation. Soundex matching method is used here to test (see Table 2).

b, f, p, v	1
c, g, j, k, q, s, x, z	2
d, t	3
l	4
m, n	5
r	6

Table 2: Soundex

5 Evaluation

5.1 Metrics

Two different analysis metrics are applied to evaluate the results.

5.1.1 Accuracy

Accuracy represents the proportion of matched words in all correct words. It is used in the evaluation of the result of basic GED due to that the length of the returned list is corresponding to the input.

$$Accuracy = \frac{Matched\ number}{Total\ number\ of\ input}$$

5.1.2 Precision and Recall

Precision and Recall represent the result when one input word receives multiple results returned. Recall means the proportion of matched words within the relevant results. Precision means the percentage of matched words within all possible predictions.

$$Recall = \frac{Matched\ number}{Total\ number\ of\ relevant\ results}$$

$$Precision = \frac{Matched\ number}{Total\ number\ of\ predictions}$$

5.2 Direct predictions

For the first and the last parts in the tokens, they are treated as the best predictions for themselves at the beginning. The result of matches needs to be verified with a one-to-one comparison with the 'correct.txt'.

5.2.1 Results

Table 3 represents the evaluations for direct predictions parts.

Parts	Matches	Accuracy
In the dictionary	7805/8376	93.18%
Includes typos	299/314	95.22%
Summary	8104/8690	93.26%

Table 3: Results of direct predictions

Table 4 and 5 are some samples from words which are not matched.

Mis-matched Tokens	Correct
u	you
didnt	didn't
cause	because

Table 4: Samples from parts1

Mis-matched Tokens	Correct
2	to
sum1	someone
str8	straight

Table 5: Samples from parts3

5.2.2 Analysis

Most predictions in this part match whole, while there are still some miss. Due to the result, it can be generalized that some words can be with the right spelling but wrong meaning like 'cause' and 'because' (Table 4). Some abbreviations and euphony cannot be predicted because the dictionary does not include special symbols and numbers (Table 5).

5.3 Basic GED

5.3.1 Results

Table 6 represents results of basic GED with single prediction. And Table 7 shows failed samples from basic GED.

Matched / Total	99/1632
Accuracy	6.07%

Table 6: Results of basic GED

Tokens	Correct	Returned
brah	brother	brach
mesage	message	mesange
cosplaying	cosplaying	complying

Table 7: Samples from basic GED

5.3.2 Analysis

According to the results, the mismatch data constitutes two forms. Some words like 'brah' which is spelt wrong and returned with 'brach' when 'brother' is expected. This should be improved with multiple returns. Others like 'cosplaying' which is correct but returned with 'complying' because 'cosplaying ' is not in the dictionary. This kind of modern words or slang shows the limitation of the dictionary.

5.4 GED with changes

5.4.1 Results

Table 8 represents results with multiple predictions and mutative parameters.

Table 9 and 10 represents some samples.

$[m, i, d, r]$	Match	Precision	Recall
$[+1, -1, -1, -1]$	150	2.03%	9.19%
$[+2, -1, -1, -1]$	143	2.79%	8.76%
$[+1, -2, -1, -1]$	134	1.97%	8.21%
$[+1, -1, -2, -1]$	182	0.93%	11.15%
$[+1, -1, -1, -2]$	145	2.96%	8.88%

Table 8: Results of GED with changes

Tokens	Correct	Returned
mesage	message	mesange, mesnage, message
opolo	opolo	ovolo, polo

Table 9: Samples of GED with multiple returns

Tokens	Correct	[1,-1,-1,-1]	[1,-1,-2,-1]
spirite	spirited	spirited, spiriter	spilite, spirate, spirit, spirited, spiriter, spirity, spirits, sprite

Table 10: Samples of GED with parameters changes

5.4.2 Analysis

Firstly, return with multiple predictions will match more words and increase the Recall. Some sample words show above like 'mesage' in the misspell.txt will just return with 'mesange' with basic GED. But with multiple returns, it will return three results to match the correct word. Though, some words like 'opolo' still cannot be corrected because it do not exist in the dictionary.

Additionally, the Recall will increase a little when the parameters change. It means the weights of different operations are changing and the program will receive more or less returns to evaluate. It may refer to that people are used to use abbreviations while typing online. So that more situations of misspelt are caused by lack of letters rather than wrong letters.

However, these changes still cannot match most of the words because some of them are not in the dictionary.

5.5 GED with Soundex

5.5.1 Results

Table 11 represents results with GED + Soundex.

Matches	Precision	Recall
148/1632	very tiny	9.07%

Table 11: Results of GED with Soundex

5.5.2 Analysis

According to the results, the Recall reduces instead of increasing and the program returns so many lead to the precision becomes very tiny. Combining Soundex with GED does not help the matching, because the limitation is not on the method used to do approximate string search. The limitation is on the dictionary.

6 Conclusion

In conclusion, the best matches for this project should be 8286/10322 where the Recall is 80.28%. Some changes made on the search method may affect the results a little but it is difficult to increase the Recall to a higher level. All of these approximate search methods are based on the given dictionary which does not contain any symbols, euphony and abbreviations. According to that, the limitation of the success rate for this project is on the dictionary rather than the methods. Maybe importing with a dictionary contains slang will increase the matching rate.

References

Timothy Baldwin, Marie-Catherine de Marneffe, Bo Han, Young-Bum Kim, Alan Ritter, and Wei Xu. 2015. Shared tasks of the 2015 workshop on noisy user-generated text: Twitter lexical normalization and named entity recognition. In *Proceedings of the Workshop on Noisy User-generated Text*, pages 126–135, Beijing, China.