Spelling Correction Method Evaluation

Abstract

This paper investages several spell checking methods. The main goal is to compare and analysis the performance of spelling correction methods on a peculiar data set.

1 Introduction

Spelling correction is a basic task in natural language processing. The method of spelling correction have been very mature. Some neural method also been presented in recent years. In this paper, we investigate some non-neural network method to deal with it. Including soundex, n gram, edit distance and editex.

2 Method

In this section, we simplily introduce the algorithms we evaluated in our paper.

Soundex Soundex, developed by Odell and Russell, and patented in 1918 (Hall and Dowling, 1980). Uses codes based on the sound of each letter to translate a word into an at most 4 character's string. In this paper, we call it Soundex code. Soundex algorithm are given in Figure **??**.

N-Gram N-Gram methods are string distance methods based on n-gram counts, where a n-gram of string s is any substring of s of some fixed length. We get silimarity of words by comparing n-gram distance, which is proposed by (Ukkonen, 1992), defines as:

$$|N_s| + |N_t| - 2|N_s N_t| \qquad (1)$$

where N_s is the set of n-gram in string s.

Edit distance Edit distance calculates by defining single character insertions, deletions (indels) and replacements costs needed to tansform one string into another.

Editex Presented by (Zobel and Dart, 1996), is a method combines phonetic matching and edit distance. In this method, alphabets also be grouped similar with soundex. Then, edit distance will be calculated with the grouping info, the distance between two letter in the same group defines 1 while the different group defins 2. This scheme makes distance between similar phonetic words smaller.

Type	**Words number**
Testset size	716
Dictionary size	393954
Misspelled in Dictionary	175
Correct not in Dictionary	122

Table 1: Dataset

3 Experiment

3.1 Dataset

In our experiment, we use a particular dataset: a number of headwords taken from UrbanDictionary1 that have been automatically identified as being misspelled (Saphra, 2016). We use a sub-sample of actual data posted to UrbanDictionary, this dataset simulates the spelling data in the real world as much as possible.

3.2 Settings

Soundex Calculate the soundex code for every word and then matched with global edit distance.

N-Gram We evaluate the N-Gram algorithm for n in range 1 to 9. For a particular n, we first pad (n-1) $\sharp$ in the front and end of every word. This gurantee the building of n-gram set. For example, 5-gram set for word "he" defined as:

$$\{\sharp\sharp\sharp\sharp h, \sharp\sharp\sharp he, \sharp\sharp he\sharp, \sharp he\sharp\sharp, he\sharp\sharp\sharp, e\sharp\sharp\sharp\sharp\} \qquad (2)$$

Edit Distance There are two kinds of edit distance algorithm, local edit distance and global edit distance. In this paper, we implement both of them and evaluate them. For global edit distance, we have two distance calculate scheme: 1) (+1) for indel and mismatch and nothing to

N	Predicted	Right	Precision	Recall
1	7150	183	2.56	25.56
2	1484	151	10.19	21.09
3	1429	149	10.43	20.81
4	1426	148	10.38	20.67

Table 2: N-gram algorithm results

do for match; 2) (+1) for indel and mismatch and (-1) for match. The different between them will discuss in Section **??**. For local distance algorithm, we use (-1) for indel and mismatch and (+1) for match, and always assign 0 if 0 is better.

Editex We calculate the editex follows (Zobel and Dart, 1996) settings.

4 Results & Evaluation

For all experiments in this paper. We keep all parallel best results as the predicted correction. That is, for example, the soundex code for word "adn" is "a35", while for "attain", "attainabilities", "adamas", "atom" etc. the code are the same. No matter how many word in dictionary have the same code with "adn", all of them seem equally. Although we know this may result in a bad performance on precision if the predicted set is too large.

We use this evaluation method because the purpose of this paper is illustrate the characteristics of different string matching alrothrims by spelling correction experiments on a particular dataset, rather than find a best algorithm with parameters to correct the misspelled datasets.[1]

4.1 N-Gram Algorithm Evaluation

The results of n-gram algorithm shows in Table 2, in which we see that when $N = 1$, the number of predicted words is large and recall is also the highest. This illustrates that around 25.56% of the misspelled words are character shifted without deletion and insertion, because the best matches for $N = 1$ means two words have the same consist of characters but maybe different order. However, the precision of $N = 1$ is really low so it is not a good parameter for this algorithm. $N = 2$ and $N = 3$ gives more balanced and signicicant resuls. For $N >= 4$, the results are all the same. The reason might be that padding $n - 1\sharp$ strategy for n-gram result in the related similarity tend to be table when n is large.

[1]The dataset we used contains only 716 words, even find best algorithm with parametrs, it might be some kind of overfitting of the particular datasets.

Scheme	Predicted	Right	Precision	Recall
GED-1	5528	253	4.57	35.34
GED-2	2497	204	8.16	28.49
LED	727774	133	0.02	18.58

Table 3: Edit distance algorithm results

4.2 Edit Distance Algorithm Evaluation

Table 3 shows the resuls of edit distance algorithm. We see that even both the first line and second line are results of global edit distance, different distance calculate strategy result in widely divergent results. Both strategy with (+1) for indel and mismatch. The difference is the first do nothing for match while the second (-1). Which always makes the first strategy matchs more result. For example, calculate the edit distance between "aginst" with "against" and "agist". This first strategy's result is the same 1 because "against" add one character and "agist" delete one character. But the second strategy calculate -5 for "against" and -4 for "agist", which means "against" is more similar with "against" because they have more similar elements (characters). Both strategy have its characteristic and we can not say which one is better, it depends on the application scenarious.

As for results of local edit distance algorithm. The predicted words is explosive large but both precision and recall are not satisfactory. Which indicates that this algorithm is not suitable for the task. The reason might be that misspelled words are not tend to be a part of the corresponing correction or in reverse. Nevertheless, we can not say local edit distance is a bad algorithm. It is very powerful in other tasks such as gene alignments.

4.3 Comprehensive Evaluation

Table 4 shows the two words' demostrate output of the best match for each algorithm and Table 5 shows the results of all evaluation.

From Table 5 we knows that toth soundex and local edit distance predicted too much best matches. The reason for soundex is that for words with a fixed first letter, the number of different soundex code is only 1000, 3 digitals from 0-9. This will lead to plenty of words have the same soundex code.

Although the precision of soundex is extremely low, it's result seems much better than local edit distance for both precision and recall. It is not hard to think of computing edis distance based on the soundex results. This is a

Method	Misspelled	Correct	Matched set
Soundex	accually	actually	akal, axile, azalea, asylees, auxiliar, ...
	ahain	again	awin, annoy, aani, aoyama, anne, ayme, anay, ...
Local Edit Distance	accually	actually	actually, tactually, unactually, contactually, ...
	ahain	again	disenchain, rechain, toolchain, toolchains, ...
Global Edit Distance	accually	actually	actually
	ahain	again	chain, amain, arain, again, ghain, alain, hain
N-Gram (N=2)	accually	actually	actually, ally
	ahain	again	ain
Editex	actually	actually	usually, actually, annually, casually, chally, ...
	ahain	again	amain, attain, arain, again, alain, hain

Table 4: Demostrate of different algorithm's spelling correction result.

Method	Predicted	Right	Precision	Recall
Soundex	495146	436	0.09	60.89
Local Edit Distance	727774	133	0.02	18.58
Global Edit Distance	2497	204	8.16	28.49
N-Gram (N=2)	1484	151	10.18	21.09
Editex	2830	230	8.13	32.12

Table 5: All method results

naive combination of two algorithm which is exceeded by editex.

In Table 5, we find that

5 Conclusions

References

Patrick A. V. Hall and Geoff R. Dowling. 1980. Approximate string matching. *ACM Comput. Surv.*, 12:381–402.

Naomi Saphra. 2016. Evaluating informal-domain word representations with urbandictionary. In *RepEval@ACL*.

Esko Ukkonen. 1992. Approximate string matching with q-grams and maximal matches. *Theor. Comput. Sci.*, 92:191–211.

Justin Zobel and Philip W. Dart. 1996. Phonetic string matching: Lessons from information retrieval. In *SIGIR*.