A correction method of word spelling mistake for English text

Guimin Huang, Jingru Chen, Zhenglin Sun
School of Computer Science and Information Security, Guilin University of Electronic Technology, Guilin, GuangXi, 541004, China
stanaugust@163.com, gmhuang@guet.edu.cn

Abstract. Spell correction is already a mature field, the need to combine advantages of different methods for better performance arises in non-word problem. A combined spell correction method is proposed in this paper, which contains Levenshtein distance for comparison between misspelled words and correct spelled words in dictionary, improved Double Metaphone algorithm that includes vowel phoneme rule sets aimed at Chinese English learners, and global vectors(GloVe) for character representation that can generate vectors in order to obtain better suggestion lists for misspelled words. Result shows that the combined approach proposed in this paper is better than phonetic correction or edit distance method only, and a comparison with two generally implemented spell check tools is done for experiment which shows that this approach is better than them in correcting misspelled words, and the success rates of suggestion lists for spelling mistakes hit the spot.

1. Introduction
Almost every computer-based task involves spelling correction, writing in text editing software, typing in search engines, and coding in programming environment, for instance. For the human readers, spelling mistake is not a matter, but for the automated system, misspellings can pose an unsatisfying problem that limits the effectiveness [1]. The most common and effective way for detecting and correcting misspelled words is using Levenshtein distance [2] to quantify the similarity with a dictionary viewed as standard. While there are a lot of related distance measures and their applications, the Levenshtein distance is still the promising way of applying edit operations to compare two strings [3]. But it is one-sided that just to estimate the edit distance between misspelled words and correct words.

According to the studies performed to analyze the types and the trends of spelling errors, such as the most notable one performed by Damerau [4], spelling errors can be generally divided into two types, Typographic errors and Cognitive errors [5]. Typographic errors are mainly caused by mistyping the keyboard, including missing a letter, inserting an extra letter, transposing two adjacent characters and replacing a letter by a wrong character, misspelling result from one of four types of single error above can cover 80 percent of spelling mistakes [4], and composite spelling mistakes result from typos also belong to this category. As stated above, typographic errors can be duly handled by Levenshtein distance with a correct spelled dictionary.

Cognitive errors are caused by the inability to spell words correctly due to unknown words or cognitive confusion. Only function of spelling is to represent phonemes [6]. For an unfamiliar word or just a word with vague memory, whatever a foreign learner or native speaker, one may attempt to recall the pronunciation for reference. As a consequence, misspelling word has the same or similar pronunciation
to the correct word that intended to write down in the case of cognitive errors. Therefore, Double Metaphone [7] algorithm is adopted for correction of the cognitive errors.

Metaphone [8] is one of the major improvements to the Soundex algorithm, it is similar to the classic Soundex in concept, but it is much richer in its approach to phonetic coding. Metaphone can be viewed as a rule set, it can map letter combinations into corresponding consonant classes. Ten years later, on account of the defect in mapping code, Philips L. proposed Double Metaphone algorithm, it makes some modifications to the original consonant classes, add A and remove W and Y, encoding the first phoneme of all words beginning with vowel as A, and different accents are considered, so it can return different codes for a polyphonic word. Double Metaphone has unique advantages on matching spelling mistakes related to pronunciation, and it can return more reliable word phoneme encoding, especially for homophones which is the key point of cognitive errors of misspelling. In this paper, fine tuning on vowels is done for Chinese English learners, and experiment shows a satisfying result according to CLEC [9] corpus.

As for generating suggestion lists for spelling correction, GloVe [10] is adopted for character representations, which can obtain vectors of characters. Adding character vectors up to get the word vectors, so that misspelled word has a vector, which is the reference that used to compare to the suggestions generated from improved Double Metaphone algorithm and Levenshtein distance for sorting the list.

2. Related Work

There have been proposed several attempt at combination of spelling correction methods. Kilicoglu H et al. [1] presented an approach mixed by Levenshtein distance, phonetic correction using Jazzy and word2vec with CBOW for calculating the contextual similarity score of the suggestions, to solve the misspelling problems caused by the consumers of answering health questions system. They weighted the combination methods by adjusting coefficients, and yielded F1-score of 0.61 for best weighted combination. The limitation of their method is not suitable for a wide field, and it is an uncertain factor that tuning the weights.

Li J et al. [11] combined Metaphone, Levenshtein distance and 2-gram model using a 3-layer neuron network, which they called L.M.T algorithms and WLR neural network. In general, this method is a 12-3-6 three layers’ structure neural network model, called word list neural network ranking (WLR) model, two predictions for each L.M.T algorithms are generated based on misspelling word, every prediction is decided by two features named Jaro-Winkler distance and word frequency. And finally, the system selects only one word through six suggestions produced by L.M.T algorithms for the last recommendation, that based on Jaro-Winkler matric and word frequency normalized by Normal Probability Density function. Their result shows that the system gained the highest hitting rate for 74.69% of first hitting rate, just a little higher than Levenshtein algorithm. As critical thinking, a such huge system established for a little improvement in spelling correction, in the belief of simple is good, is useless for a general field.

Pande H. [12] put forward a search space reduction approach for spelling correction, which is distance measure agnostic, and can be used before distance measure methods of correcting misspelled words. The goal of the approach is to reduce the search scope in spell correction dictionary for misspelling words to reduce time complexity, and search space is just a fraction of the whole dictionary. In fact, it is a filter before spell checking algorithms. The main idea is combining word2vec with fast k-NN using a Ball Tree to implement the clustering of words in dictionary, misspelled words are classified to a cluster, and correcting suggestions are generated from this cluster. Experiment result shows a pretty high performance, for equal or less than 2 Damerau-Levenshtein distance, the success rate can reach 99.59%, means that the cluster is nearly perfect. The core of splitting words into single training units for word2vec is that it remains the phonetic information, in the way that let each sequence contain the longest contiguous consonants or vowels but not both of a word, it is impressive and effective, but it isn’t used in the spell correction algorithm. Thus the phonetic information-retentive manner is used for reference in the approach of this paper, for verifying reliability of the method and applying it to the suggestion generation system.
3. Approach
As mentioned above, Levenshtein distance and Double Metaphone is combined to generate the
suggestions of correcting the misspelled words, and GloVe is adopted for character representations to
sort the suggestion list. This part is set forth the several segment respectively.

\[
\text{cell value} = \min\left(\text{diagonal cell} + \text{ALTCST}, \text{upside cell} + \text{DELCST}, \text{left cell} + \text{INSCST}\right)
\]

Levenshtein distance is already a mature algorithm in a wide application, as a simple description, a
brief introduction to the principle is presentation here. Operating distance is calculated by the cost in the
cell of calculation chart, assume that cost of one operation is 1, thus the alternative cost marked as
ALTCST, cost of deletion marked as DELCST, and cost of insertion marked as INSCST are 1, the cell
value can be calculated as (1).

| p | r | i | n | c | i | p | l | e |
|---|---|---|---|---|---|---|---|---|
| p | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| e | 1 | 0 | 1 | 2 | 3 | 4 | 5 | 5 | 6 | 7 |
| n | 2 | 1 | 1 | 2 | 3 | 4 | 5 | 6 | 6 | 6 |
| c | 3 | 2 | 2 | 2 | 3 | 4 | 5 | 6 | 6 | 6 |
| i | 4 | 3 | 3 | 3 | 3 | 2 | 3 | 4 | 5 | 6 |
| l | 5 | 4 | 4 | 3 | 4 | 3 | 2 | 3 | 4 | 5 |
| e | 6 | 5 | 5 | 4 | 4 | 3 | 3 | 3 | 3 | 4 |
| d | 7 | 6 | 6 | 5 | 5 | 4 | 4 | 4 | 3 | 4 | 4 | 4 | 3 |

For instance, there are two strings “principle” and “penciled”, the calculation chart is shown in the
Table 1. The value in the bottom right corner of the table, according to Levenshtein distance, is the
shortest distance of the two strings.

In this paper, a suitable dictionary contained about 60,000 words for spelling correction is adopted,
which is prepared for CET 4 and 6, each typed words is compared to words in this correct spelled
dictionary using Levenshtein distance. As a consequence, several suggestions are generated from this
method.

As for cognitive spelling errors, Double Metaphone is adopted for phonetic correction, which is
embedded in Jazzy\(^1\) and modified for application. In this version, 15 consonant classes are used to map
letter combinations, that represented by the following characters: “A B X S K J T F H L M N P R 0”.

A lot of sets of rules are used to generate the phonetic codes of words represented by the consonant
classes, but vowels are not considered in it except initial vowels in words represented by “A”, so vowel
encoding rules is put forward in the research based on CLEC [9]. In the research of spelling mistakes in
100 essays from ST3 in CLEC, the result shows that nearly 32.8% misspelling words are caused by
misusing vowels, such as “minite” and “socity”. But misspelling like “ouful” which indicates “awful” in
the essay, cannot be corrected by Double Metaphone.

So the rule sets for vowel phonemes are established by the combination of vowels and consonants in
correct spelled words with the same pronunciation, for example, “a’a’a’aw’al’or’ou’oor’our’” are
pronounced as “[ə:]”, when the word misspelled as “ouful”, the correct spelling form “awful” can be
easily found through the rule sets for vowels. In this research, 19 vowel phonemes are all established in

\(^1\) http://jazzy.sourceforge.net.
the rule sets for vowels, details can be found in Table 2. The vowel phoneme “[æ]” is not set, because only “a” can pronounce like this, it is meaningless to add it in the confuse sets.

### Table 2 Rule Sets of 19 Vowel Phonemes

| Phoneme | Rule Sets |
|---------|-----------|
| [i:]    | e'ea\ e'e\ i'e |
| [i]     | i'ei\ y'ay |
| [ə:]    | e'al\ ur\ ear\ or |
| [ə]     | ou'ure\ e'ar\ or'a\ e'o\ u |
| [ə:]    | a'ar\ au\ aw\ al\ or\ ou\ oor\ our |
| [ə]     | o\ a |
| [u:]    | o'oo\ u\ u |
| [u]     | o'oo\ u\ ou |
| [a:]    | a'al\ ar\ au\ ear |
| [A]     | o'u\ oo\ ou |
| [e]     | e'ea |
| [ei]    | a'ai\ ay\ e'a\ ey |
| [ai]    | i'y\ i'e\ uy |
| [oi]    | o'i\ oy |
| [au]    | ou'i\ ow |
| [ou]    | o'oa\ ow |
| [ia]    | ear\ eer\ ere\ ea\ easier |
| [eo]    | air\ ear\ are |
| [ua]    | oor\ our\ ure |

The rules established for vowels is an imitation of Double Metaphone rules, as a result, the application is viewed as an improvement of Double Metaphone, and the suggestions generated from it can gain a better performance for the English text written by Chinese learners.

At the final part of this approach, the principle of bag of characters, which is mentioned above, is adopted for sorting the suggestions of correcting misspelled words. In this part, GloVe is used for generating vectors to represent the combination of characters, these vectors are added up to generate n-dimensional vectors for representing the words from dictionary for suggestions and misspelled words, and cosine similarity is adopted for comparing the similarity between misspelled words and suggestions.

According to the corpus, which is a dictionary, co-occurrence matrix M is built for bag of characters. Bag of characters means that words are divided into combination of characters based on longest continuous vowels or consonants, for example, “approach” can be divided into “a ppr ao ch”. And then, the approximation relation between character vectors and co-occurrence matrix M based on equation 2.

$$v_i^T v_j + b_i + b_j = \log(M_{ij})$$  \(2\)

In the equation, is the times that representing character combination \(j\) occurred in the context of character combination \(i\), \(v_i^T\) and \(v_j\) are the vectors needed in this part, \(b_i\) and \(b_j\) are bias terms of two vectors. The loss function is showed in equation 3. \(N\) is the size of character combinations, \(f\) is a weighting function to control the affection of the co-occurrence frequency, and its upper limit value is 1.
The main equations of theory utilized here of GloVe are (2) and (3). The preprocessing of this part can be simply described as follow. First, prepare the character combinations for training using GloVe model, each word in the correct spelled dictionary is divided into sequences. Second, build the co-occurrence matrix for the prepared sequences, and minimize the loss function to train the model with layer size as \( n \), which means that the dimensional of vectors is \( n \), and window size as \( m \), and min frequency as \( t \), occurrence of character combinations lower than \( t \) are ignored. Then, a matrix of vectors is constructed for representing character combinations.

When the spell correction is done by Levenshtein distance and Double Metaphone, and after suggestions are given, the sort system is ready for them by the things prepared above. Misspelled words and its suggestions are divided into sequences as the method for training GloVe model, and search these sequences in the matrix of character combination vectors, and add these vectors up to gain the vectors for misspelled words and its suggestions, compare each suggestion with the misspelled words through cosine similarity, and the final suggestion list is sort by the similarity.

4. Experiment
The approach is implemented in Java on Eclipse. Jazzy, an open source spell checker, is adopted for Double Metaphone part, and improved through the method mentioned above. For training bag of characters, DL4J\(^2\), a Deep Learning library, is used to generating vectors. Whole experiment is executed on Windows 10 Professional version with Intel(R) Core(TM) i5-8500 CPU @ 3.00GHz with 16.0GB of RAM.

In this paper, a combined dictionary used to train the GloVe model is constructed by four dictionaries on the four websites\(^3\), and finally it contains 293,920 words. As for the dictionary for generating suggestions, in this experiment, a vocabulary list is adopted for CET 4 and 6, which contains about 60,000 words. In this part, 200 essays are chosen from CLEC for testing, which contains 100 essays from ST3 for CET4 and 100 essays from ST4 for CET6, and total spelling errors are 655.

After several attempts to train the GloVe model, the dimension of vectors \( n \) is set to 100, the window size of training \( m \) is set to 4, and min frequency of character combinations \( t \) is set to 5, and finally a matrix of vectors for character combinations is generated, by which the misspelled words and their suggestions can gain the word vectors. Cosine similarity is used to produce the suggestion rank.

In the contrast test, the rate of precision, the rate of recall, and F1-score are adopted for measuring the performance as metrics, their definitions are shown in (4), (5) and (6).

\[
\text{Precision} = \frac{S_i}{S_i + S_f} \times 100\% \quad (4)
\]

\[
\text{Recall} = \frac{S_i}{N_m} \times 100\% \quad (5)
\]

\[
F_1 = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \times 100\% \quad (6)
\]

\(^2\) http://deeplearning4j.org.
\(^3\) http://www-01.sil.org/linguistics/wordlists/english/wordlist/wordsEn.txt, http://www-personal.umich.edu/~jlawler/wordlist, http://rosettacode.org/mw/index.php?title=Textonyms/wordlist, http://www.miegelstronk.com/corncob_lowercase.txt.
In the equations of metrics, \( S_t \) refers to suggestions in the top 3 of suggestion lists that truly correct misspelled words, \( S_f \) refers to suggestions fail to correct misspelled words, \( N_m \) refers to the total spelling errors. As a logical consequence, the F1-score can be calculated as (6).

As a contrast, a Java version of Aspell named JaSpell4 and a JNA based Java implementation of Hunspell called HunspellJNA5 are compared with the approach proposed in this paper. The result of spelling correction evaluation is shown in Table 3. Otherwise, the strict assessment criterion is applied, which means that correct spelling words only in the top 3 of suggestion lists are viewed as correcting misspelled words successfully.

### Table 3: Spell Correction Evaluation

| Method          | Precision | Recall | F1   |
|-----------------|-----------|--------|------|
| JaSpell         | 0.69      | 0.65   | 0.67 |
| HunspellJNA     | 0.77      | 0.73   | 0.75 |
| This approach   | 0.86      | 0.82   | 0.84 |

According to the result of this experiment, the approach proposed in this paper gains a higher precision than the other two spell checkers, with about 16.86% higher than JaSpell and about 8.61% higher than HunspellJNA, and the F1 measure of this approach obtains a better score. Thus this approach has a better performance than the others.

For testing the effectiveness of ranking system that using the theory of bag of characters, hitting rates of suggestion lists is evaluated in four sections, which can be called as top 1, top 3, top 5, and top 10. The result is shown in Fig. 1.

![Figure 1. Evaluation of the Hitting rates](image)

As the hitting rates shown, the top 1 suggestion can gain a twenty percent higher performance than the others, this is vital for a spell checking system, which means that users can get the most reliable suggestion at once. And the approach put forward here can give better spell correction suggestions in any section.

### 5. Conclusion and Future Work

In this paper, a fresh combined spell correction approach is proposed for Chinese English learners, which contains Levenshtein algorithm, improved Double Metaphone algorithm with vowel rule sets, and trained GloVe model for generating vectors of character combinations to sort the suggestion lists. The approach gains a better performance, but there are still some details need to be promoted, the Levenshtein algorithm

---

4 http://jaspell.sourceforge.net.
5 Website: https://github.com/dren-dk/HunspellJNA.
can be instead of Damerau-Levenshtein distance, and the rule sets of Double Metaphone can move forward since the vowel rules put forward in this paper are suitable for Chinese learners but not verified in a wide field.

Think of it in another way, spell correction can also be done using n-gram embeddings or finite-state, and word frequency can be used by adding a weight coefficient, if these methods are organized effectively, the performance may scale new heights. Therefore, a further research need to be done for creating a more satisfying spell correction system.

Acknowledgment
This work was supported by the National Natural Science Foundation of China (No. 61662012, No.61662015).

References
[1] H. Kilicoglu, M. Fiszman, K. Roberts, and D. Demner-Fushman, “An ensemble method for spelling correction in consumer health questions,” in AMIA Annual Symposium Proceedings, San Francisco, 2015, pp. 727.
[2] V. I. Levenshtein, “Binary codes capable of correcting deletions, insertions, and reversals,” in Soviet physics doklady, Soviet Union, 1966, vol. 10, pp. 707-710.
[3] L. Yujian and L. Bo, “A normalized Levenshtein distance metric,” IEEE transactions on pattern analysis and machine intelligence, vol. 29, pp. 1091-1095, June 2007.
[4] F. J. Damerau, “A technique for computer detection and correction of spelling errors,” Communications of the ACM, vol. 7, pp. 171-176, March 1964.
[5] N. Gupta and P. Mathur, “Spell checking techniques in NLP: a survey,” International Journal of Advanced Research in Computer Science and Software Engineering, vol. 2, pp. 217-221, December 2012.
[6] E. Carney, A Survey of English Spelling. New York, CA: Routledge, 2012.
[7] L. Philips, “The double metaphone search algorithm,” C/C++ users journal, vol. 18, pp. 38-43, June 2000.
[8] L. Philips, “Hanging on the metaphone,” Computer Language, vol. 7, pp. 39-44, December 1990.
[9] G. Shichun and Y. Huizhong, Chinese Learner English Corpus. Shanghai, CA: Shanghai Foreign Language Education Press, 2003.
[10] J. Pennington, R. Socher, C. Manning, “Glove: Global vectors for word representation,” in proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), Doha, 2014, pp. 1532-1543.
[11] J. Li, K. Ouazzane, H. Kazemian, Y. Jing, and R. Boyd, “A neural network based solution for automatic typing errors correction,” Neural Computing and Applications, vol. 20, pp. 889-896, September 2011.
[12] H. Pande, “Effective search space reduction for spell correction using character neural embeddings,” in proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, Valencia, 2017, pp. 170-174.