Design and Implementation of a Course Answering System Based on Error Correction and Trie Tree Language Model

Danyang Li, Yahui Zhao, Mengjiang Luo and Rongyi Cui

Intelligent Information Processing Lab., Dept. of Computer Science & Technology, Yanbian University, Yanji, 133002, China
Email: yhzhao@ybu.edu.cn

Abstract. With the rapid growth of network information, the accuracy of input text affects the retrieval results, text proofreading technology emerges as the times require in order to avoid the situation of “wrong answers” caused by wrong questions when searching for information. In the course question answering system, we also need to consider the query speed and efficiency. In order to avoid the trouble of manual proofreading, this paper proposes a system which can automatically correct wrong questions in the professional field of curriculum. Firstly, used the edit distance method for fuzzy matching of error strings; then, used trie tree language model to store data to improve query efficiency. Finally, compared the proofreading effect under different text similarity thresholds, and selected the best value for the experiment. After experimental analysis and comparison, the best result is selected when the text similarity is 0.5, the accuracy rate is 77.91%, the recall rate is 67%, and the F value is 72.04%. Experiments show that the system designed in this paper can effectively correct the wrong text in the field of computer.

1. Introduction
Text proofreading technology can avoid the problem of “wrong answer” caused by wrong input when searching information. In the query system, the storage of trie tree can reduce the query time of the same prefix character and improve the query efficiency.

This design uses edit distance for fuzzy matching and proofreading of wrong questions, uses trie tree model to store dictionaries, and realizes a course Q&A system of intelligent question correction.

2. Text Proofreading
Text proofreading is an important branch of natural language processing, which refers to finding errors in text and giving suggestions for modification. The keywords input affect the search results of Q&A system. Therefore, the research of text proofreading is becoming an important topic.

2.1. Representation of Chinese Text Error
The study of Chinese text errors began in the early 1990s. Referring to the research of English texts [1], domestic scholars divide Chinese text errors into “true word errors” and “non word errors” [2-3]. Chinese characters have their unique characteristics: there will not be any new words in the existing dictionary, but the arrangement and combination of different Chinese characters can represent more than 1 million words. Therefore, from the perspective of words, “true word error” refers to the error caused by the wrong use of a word into another word in the dictionary. “Non word error” is defined as the substitution, insertion or deletion of one or more Chinese characters in a word, which makes the...
word string not a word in the dictionary [4]. For example, “数据哭” (an error string) should be “数据库” (database).

2.2. Research Status of Text Proofreading
In recent years, some scholars have improved the Chinese text proofreading technology with reference to the English text proofreading technology [5-6], and provided some Chinese text proofreading systems [7], such as Black Horse Proofreading System, Radical Cloud Error Checking System and Fangcun Intelligent Proofreading System. But these systems have the following problems [8]: (1) The evaluation indexes of proofreading results need to be improved, such as recall rate, accuracy rate, correction rate, etc. (2) Most proofreading systems are commercial systems with high costs.

In view of the above shortcomings, this paper implements a Q&A system which can automatically correct errors in the field of computer professional courses combined with trie tree language model.

3. Question Correction System
3.1. Question Correction Design
Error correction module mainly includes two parts: problem checking and text proofreading. First of all, we clean the question texts including segmentation and removal of stop words, leaving only professional words as input, and using relevant error checking methods to identify errors. In text proofreading, the system performs fuzzy matching [9] and similarity calculation for error string and selects one as suggestions in dictionary. The framework of text error correction is shown in figure 1.

3.2. Text Similarity Calculation
The number of Chinese characters added, deleted, transposed or replaced by converting Chinese word string $X$ to another word string $Y$ is the edit distance [10], recorded as $\text{dist}(X,Y)$. The minimum editing distance is required to convert the first $i$ characters of string $X$ into the first $j$ characters of string $Y$ by $\text{dist}[i][j]$ of a two-dimensional array. The calculation formula is as follows:

$$\text{dist}[i][j] = \min \{ \text{dist}[i-1][j-1], \text{dist}[i-1][j], \text{dist}[i][j-1] \}, X[i] \neq Y[j]$$  \hspace{1cm} (1)

$$\text{dist}[i][j] = \text{dist}[i-1][j-1], X[i] = Y[j]$$  \hspace{1cm} (2)

$$\text{dist}[i][j] = \text{dist}[i][0] = i, \text{dist}[0][j] = j$$  \hspace{1cm} (3)

After calculating the minimum editing distance between strings, judge whether texts are similar. If the long string length of $X$ and $Y$ is $L_{\text{max}}$, the editing distance is $LD$, and the text similarity is $S$, then:

$$S = 1 - LD \cdot L_{\text{max}}^{-1}$$  \hspace{1cm} (4)

4. Trie Tree Language Model
This design uses the trie to store the dictionary, finds the key words through the trie tree, and return the answer in the system after text proofreading.

4.1. Data Structure of Trie Tree
Trie tree is a kind of variant search tree similar to hash tree [11]. It is often used in statistics, sorting and saving a large number of strings and data. The data structure of the trie tree is shown in figure 2.

The root node of the trie tree does not store any data, and other node only stores one character. Each retrieval starts from the root node to the leaf node, and all the characters on the link are connected to form a word. Therefore, the trie tree can share common prefixes, which saves storage space and improve the retrieval speed. For example, in figure 2, “Yanji” and “Yanbian University” share the “Yanbian” character, and “Yanbian” and “Yanbian University” share the “Yanbian” string.
Trie tree reduces the traversal process of characters and improves the retrieval speed by strengthening the mapping between strings.

Figure 1. Error correction processing framework.  

Figure 2. Trie tree.

4.2. Trie Tree Operations

Add a string to the trie tree. The algorithm is as follows:

Step 1. The Input the string key to be inserted, root node of trie tree is RootNode, and the character retrieval position charIndex is initialized to 0;

Step 2. If RootNode is empty, create a new character node;

Step 3. Name current node CurrentNode, make CurrentNode start from RootNode;

Step 4. Note that Compare is the D-value between the current tree node and the key to be inserted in the charIndex position;

Step 5. (1) When Compare > 0(or < 0), if the right (left) node of CurrentNode is empty, create a new character node, traverse down to step 4;

(2) When compare = 0, charIndex++. If the lower node of CurrentNode is empty, create a new character node. Continue to traverse down to step 4; If all matches have been completed, add new character information and go to step6;

Step 6. End.

Find the string in the trie tree. The algorithm is as follows:

Step 1. Enter the string key to be queried and start to query the location Offset. The character retrieval position charIndex is initialized to Offset, and the string word is initialized to null;

Step 2. If the RootNode is empty or the key is empty, return word and go to Step 7;

Step 3. Make the CurrentNode start from the RootNode;

Step 4. If CurrentNode is empty, return word and go to Step7;

Step 5. Note that compare is the D-value between the current tree node and the key to be queried in the charIndex position;

Step 6. When Compare > 0 (or < 0). Continue to traverse to the left(right), go to step 4;

When Compare = 0, charIndex++. If the match is not finished, the candidate longest match word; if charIndex is equal to the length of key, return the word and go to step7. Continue to traverse down to step 4;

Step 7. End. Return the string word.

5. Design and Implementation of Course Question Answering System

5.1. System Design

In order to facilitate the students searching in the field of professional courses, and avoid the download and installation process, this system use B/S structure to develop. The project is deployed on Tomcat server, and implement with MySQL as database.
This design uses Computer Professional Dictionary which contained 10299 professional words. We need to collect answers of the words on Baidu Encyclopedia, which is the largest Chinese encyclopedia in the world. And store them in the database in advance. The system architecture is shown in figure 3, the flow chart of Q&A system is shown in figure 4.

5.2. System Implementation
This system designs two interfaces: search interface and response interface. The user can input course questions in the search interface and click the search button to enter the response interface. After text proofreading, the answer interface will prompt the user with “Are you looking for...?” and retrieve relevant answer from database and display it in the text box. The search interface is shown in figure 5.

Enter “什么是数据哭” (an error string) in the search box, and the correct question should be “什么是数据库” (What is database). The response interface is shown in figure 6.

6. Analysis of Test Results
6.1. Evaluation Criteria
In this system, machine learning evaluation standard is used to measure the error checking module, accuracy rate and correct rate are used to measure the error correction module.

(1) Error checking part

\[
\text{Precission} = \frac{\text{The number of error questions found correctly}}{\text{The number of error questions found}} \times 100\%
\]  

(5)
\[ \text{Recall} = \frac{\text{The number of error questions found correctly}}{\text{Total number of error questions in text}} \times 100\% \] (6)

\[ F = \frac{2 \times \text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}} \times 100\% \] (7)

(2) Error correction part

\[ \text{Accuracy} = \frac{\text{The number of correct proofreading}}{\text{The number of error questions found}} \times 100\% \] (8)

\[ \text{Correct} = \frac{\text{The number of correct proofreading}}{\text{Total number of error questions in text}} \times 100\% \] (9)

As long as the wrong text is changed, it is considered that the wrong question has been found. If the modified vocabulary is consistent with the correct text, it is considered that the correction is correct.

6.2. Text Similarity Threshold Comparison

We use 100 course error question texts to test the system’s error checking and correction ability. The selection of the most suitable error correction words depends on the similarity threshold. This system uses 0.4, 0.5 and 0.6 as threshold. The test results are shown in table 1 and figure 7:

| Threshold | Accuracy | Recall | F value | Accuracy rate | Correction rate |
|-----------|----------|--------|---------|---------------|----------------|
| 0.4       | 62.50%   | 55%    | 58.51%  | 90.91%        | 50%            |
| 0.5       | 77.91%   | 67%    | 72.04%  | 79.10%        | 53%            |
| 0.6       | 74.12%   | 63%    | 68.11%  | 22.22%        | 14%            |

Figure 7. Line chart of error correction data rate.

According to figure 7, when the similarity threshold is 0.5, the comprehensive result is the best. When the threshold of text similarity is 0.4, the accuracy of error correction is the highest, which is more than 90%. This shows that when the threshold of similarity is low, it is easier to find error correction suggestions in dictionaries. On the contrary, when the similarity threshold is 0.6, the accuracy of error correction decreases significantly, only 22.22%, but its error checking ability is still good, which shows that too high threshold value of similarity is not conducive to text error correction.

To sum up, this design uses the case of text similarity of 0.5 to analyze the experimental data.

6.3. Experimental Data Analysis

Under the condition that the threshold of text similarity is 0.5, some experimental data are shown in table 4, and wrong words and corrected words are underlined.
Table 2. Error question text and error correction examples.

| No. | Error question text / Chinese Meaning | Discover | Discovery correct | Error correction suggestion /Chinese Meaning | Suggestion correct |
|-----|--------------------------------------|----------|-------------------|---------------------------------------------|-------------------|
| 1   | 什么是数据哭/An error string          | Y        | Y                 | 什么是数据库/What is database              | Y                 |
| 2   | 怎样才养/An error string              | N        | -                 | -                                           | -                 |
| 3   | 欧式距离怎么求/An error string        | Y        | Y                 | 欧氏距离怎么求/How do we figure out the Euclidean distance | N                 |
| 4   | 次品统计怎么做/How to do defect statistics | Y       | N                 | 次品统计图怎么做/How to make the defective statistic chart | Y                 |
| 5   | 传书延迟会引起什么/An error string   | Y        | Y                 | 传输延迟会引起什么/What does the transmission delay cause | Y                 |
| 6   | 汉编码是什么/What is Han encoding     | Y        | Y                 | 汉熵编码是什么/What is Chinese entropy coding | N                 |

From the ability to identify errors, Some of the wrong questions are not found, such as sentence 2. “怎样才养” (An error string) is divided into “怎样才养” (An error string) at the process of word segmentation. The segmentation parts are all stop words and will not be corrected. Some mistakes are miscalculation, such as the sentence 4. The correct result should be “词频统计怎么做” (How to count word frequency). Not only did the system fail to detect “词频” (word frequency), but also changed “统计” (statistics) into “统计图” (statistical chart). The main reason is that the similarity between “次品” (substandard product) and “词频” (word frequency) is not enough, while the similarity between “统计” (statistics) and “统计图” (statistical chart) is higher.

From the perspective of error correction ability, the system has a small number of error results, such as the sentence 6, the correct question should be “汉字编码是什么” (What is the Chinese character code). The result of word segmentation is “汉/编码/是什么” (An error string), the error correction suggestion of “编码” (coding) in professional dictionaries is “熵编码” (Entropy coding). It leads to error correction as “汉熵编码是什么” (What is Chinese entropy coding).

In addition to some omissions, misjudgments and correction errors, other errors were successfully found and corrected accurately, such as sentence 1, 3, and 5.

7. Conclusion
This paper designs and implements a course answering system based on trie tree language model, which can automatically proofread the question text.

The system is based on B/S mode, and can correct the error under the optimal text similarity threshold, returns the correct answer from the database according to the error correction suggestions. The trie tree language model of this system improves the query efficiency.

Acknowledgments
This research was financially supported by State Language Commission of China under Grant No. YB135-76, scientific research project for building world top discipline of Foreign Languages and Literatures of Yanbian University under Grant No. 18YLPY13 and 18YLPY14.
References

[1] Golding A R, Dan R and Mooney J A 1999 Winnow-based approach to context-sensitive spelling correction Machine Learning 107-130.

[2] Wang L 2018 Research on Chinese Text Real-Word Error Automatic Detection and Correction Algorithm (Zhejiang: Zhejiang Gongshang University).

[3] Tao Y-C, Hai Z-Y, Shi L and Wei L 2018 Study of Chinese word collocation feature extraction and text proofreading Journal of Chinese Computer Systems 39 (11) 2485-2490.

[4] Hai C-Y 2019 Research on Text Semantic Feature Detection and Proofreading (Henan: Zhengzhou University).

[5] Karen K 1992 Techniques for automatically correcting words in text ACM Computing Surveys (CSUR) 24 (4) 377-439.

[6] Peterson J L 1980 Computer programs for detecting and correcting spelling errors Communication of the ACM 23 (12) 676-687.

[7] Yuan Z-W 2018 Research on Key Technologies of Textual Proofreading on Government Websites (Sichuan: University of electronic Science and Technology of China)

[8] Qu Q 2018 Design and Implementation of Intelligent Course Question Answering System in MOOC Environment (Jilin: Yanbian University)

[9] Wang T-T 2012 Discussion on the approximate string matching Modern Computer (001) 12-15.

[10] Ou X-C 2019 A minimum edit distance optimization algorithm based on automatic error correction Network Security Technology & Application (12) 44-48.

[11] Sun F-Y 2016 The Design and Implementation of Site Search Engine Based on the Inverted Index and the Trie (Harbin: Harbin Institute of Technology).