Hindawi
Mobile Information Systems
Volume 2022, Article ID 4634903, 7 pages
https://doi.org/10.1155/2022/4634903

Research Article

Design of Exercise Grading System Based on Text Similarity Computing

Miao Chen[1] **and Yazhou Dong**[2]

[1]*School of International Education and Humanities, Xi'an Kedagaoxin University, Xi'an 710000, Shaanxi, China*
[2]*Xi'an Space Radio Technology Institute, Xi'an 710199, Shaanxi, China*

Correspondence should be addressed to Yazhou Dong; dongyz@cast504.com

Received 10 May 2022; Accepted 18 June 2022; Published 6 July 2022

Academic Editor: Le Sun

Text similarity, as an important basis for scoring subjective items in the examination, directly determines the examination results of candidates and the work efficiency of teachers. Therefore, this paper first introduces the theoretical basis of text similarity computing and compares different calculation methods. Then, the text-similarity algorithm is designed, where by conceptualizing text terms, the computing methods based on corpus and knowledge base are combined. Then, according to the similarity computing model of text terms, an automatic grading system for exercises is designed, including the design of technical architecture, the design of functional modules, and the realization process of comprehensive grading. Among them, the grading module is the core module of the system and the key to automatic grading. The systematic test results show that the problem scoring system designed in this paper has little difference from manual marking and can achieve good scoring results.

1. Introduction

With the advent of the artificial intelligence era, Natural Language Processing is becoming more and more popular in the field of education. In modern education, to test students' skills and knowledge, examination is an important means to judge students' learning quality, but manual examination paper marking brings a huge workload [1, 2]. For objective questions or multiple-choice questions, the difficulty of judging test papers is still a little low. Objective test questions or closed answer questions are questions that offer multiple choices and generally have fixed answers. At the current stage, the technology of reviewing objective questions is very mature, and teachers only need to simply match them with the reference answers to get the answers.

However, the supervisor's test questions or open-ended answer questions require teachers to evaluate the answers. Because the answer contains many terms or words and is not unique, as long as it conforms to the central idea of the reference, students can get a certain score, and the score depends on the semantic similarity between the candidate's answer and the actual reference, which means that the greater the semantic similarity between the two, the higher the final score of candidates will be [3–5]. In addition, scoring subjective questions will have a certain scoring space, which will be influenced by the subjective factors of the examiners. To solve the above problems, some researchers use a series of related techniques in natural language processing, such as word segmentation, word vector model, text similarity, and so on, to score the answers to descriptive subjective questions [6, 7]. Therefore, if we can design a scoring system for subjective questions, the error of judging test papers caused by artificial subjective factors can be reduced, thus reducing the workload of teachers in the grading process, and the work efficiency of marking subjective questions can be improved.

It is also challenging to design a grading system to automatically grade the texts of students' test papers, which requires not only knowledge of spelling and grammar, but also knowledge of semantics, discourse, and pragmatics. Traditional models use sparse features, such as word bags, part-of-speech tags, grammatical complexity measure, word

error rate, and article length, which may have the disadvantages of time-consuming feature engineering and sparse data [8, 9], while the natural language processing technology can process these descriptive texts through Chinese word segmentation, word vectorization, part-of-speech tagging, semantic analysis, text semantic feature extraction, semantic similarity calculation, and other technologies, to realize the automatic scoring of subjective questions, which is of great significance to the development of the education industry and even the society.

This paper designs the text-similarity algorithm by conceptualizing the text terms and designs the automatic scoring system for exercises according to the text-similarity calculation model, including the technical architecture design, the functional module design, and the realization process of comprehensive scoring.

2. Theoretical Basis of Text Similarity

2.1. Basic Ideas. The concept of text similarity has many different definitions. Among them, there is a unified and informal definition of text similarity in information theory, which has nothing to do with the application field. Its basic idea is shown in Figure 1. The similarity between A and B is related to two characteristics. On one hand, the similarity between them increases with the increase of generality. When the two texts are identical, their similarity reaches the maximum value. On the other hand, it is the difference between them, that is, the similarity decreases with the increase of differences, and the greater the difference, the lower the similarity.

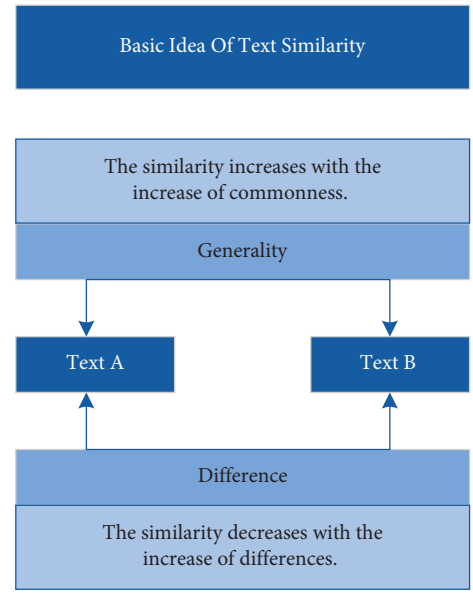

Figure 1: The basic idea of graph text similarity.

2.2. Computing Method. Text representation is the conversion of unstructured or semistructured text into characters or numbers that can be recognizable by computers [10].

2.2.1. Vector-Based Computing. The vector-based method is to represent a text as a vector in a high-dimensional space and then use the cosine distance relationship between vectors to represent the similarity between texts. Generally speaking, the cosine distance between two spatial vectors can reflect the similarity between two texts to some extent [11]. The cosine formula of the vector is

$$\begin{aligned} \operatorname{sim}(A,B) = \cos\theta &= \frac{\operatorname{vec}(A)\cdot\operatorname{vec}(B)}{|\operatorname{vec}(A)|\cdot|\operatorname{vec}(B)|} \\ &= \frac{\sum_{k=1}^{n} w_{1k}w_{2k}}{\sqrt{\left(\sum_{k=1}^{n} w_{1k}^2\right)\left(\sum_{k=1}^{n} w_{2k}^2\right)}}, \end{aligned} \tag{1}$$

where $\operatorname{vec}(A)$ and $\operatorname{vec}(B)$ are vector representations of text A and B, respectively, $\operatorname{vec}(A) = (w_{11}, w_{12}, w_{13}, \ldots w_{1n})$ and $\operatorname{vec}(B) = (w_{21}, w_{22}, w_{23}, \ldots w_{2n})$.

2.2.2. Computing Based on Sentence Length. In the process of calculating sentence similarity, the length of a sentence is also an important feature. Generally, if two sentences are similar in length, they are more likely to be similar. If there is a big difference in length between two sentences, the similarity between these two sentences will be small [12]. The formula for computing the similarity between sentence lengths can be expressed as

$$\operatorname{LSim}(T_1, T_2) = 1 - abs\left|\frac{\operatorname{Len}(T_1) - \operatorname{Len}(T_2)}{\operatorname{Len}(T_1) + \operatorname{Len}(T_2)}\right|, \tag{2}$$

where $\operatorname{LSim}(T_1, T_2)$ represents the similarity of sentence length between $T1$ and $T2$, while $\operatorname{Len}(T_1)$ and $\operatorname{Len}(T_2)$ represent the number of words in $T1$ and $T2$, respectively.

2.2.3. Computing Based on Deep Learning. For a text similarity algorithm based on supervised learning, the training model is a data set with labels that are needed to help the model train and learn, so that the text-similarity computing can be further completed. From the network structure, it can be divided into cross model and structural twin network structure, as shown in Figure 2.

The twin network structure is composed of a similarity measurement layer, coding layer, and input layer. The input layer is used to segment the original text and then express the words with their corresponding word vectors and input them to the next layer. The coding layer is used to encode the word vectors from the input layer to obtain their sentence vector representations, while the similarity layer mainly solves the similarity between sentence vectors according to the similarity algorithm [13]. After the cross model is processed by the interaction between coding layers, the outputs of its coding layer are input into the similarity layer to calculate the text similarity. The interaction introduced by the cross model in the structure of the twin network can obtain more effective and rich useful information, which reduces the deviation of calculating text semantic similarity caused by no interaction between coding layers in the twin network.

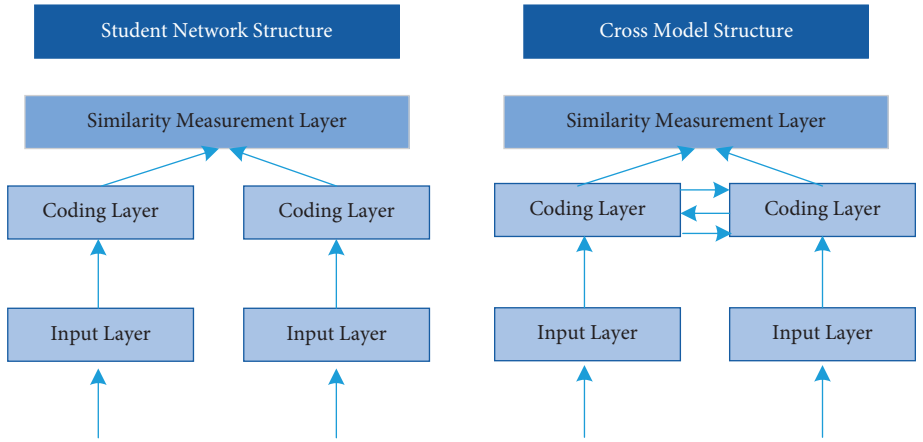

Figure 2: Network model diagram based on supervised learning.

3. Design of Text Similarity Algorithm

3.1. Calculation of Sememe Similarity. Sememe is the smallest unit of meaning to describe a concept, which is extracted from all Chinese characters and can be used to describe other words. The sememe similarity algorithm uses the relationship between the upper and lower parts of the sememe. Its calculation formula is as follows:

$$\mathrm{Sim}(S_1, S_2) = \frac{\alpha}{distance(S_1, S_2) + \alpha}, \tag{3}$$

where S_1, S_2 represent two sememes; $distance(S_1, S_2)$ represents the distance between S_1 and S_2 in the semantic tree. α is the regulating factor, which is generally 1.6 (the distance between two sememe similarity is 0.5). Based on formula (3), the hierarchical depth of semantic origin is introduced. Its calculation formula is as follows:

$$\mathrm{Sim}(S_1, S_2) = \frac{\alpha * m(depth_{S_1}, depth_{S_2})}{\alpha * m(depth_{S_1}, depth_{S_2}) + \mathrm{distance}(S_1, S_2)}, \tag{4}$$

where S_1, S_2 and dis $distance(S_1, S_2)$ have the same meanings as equation (3). α is the regulating factor, and its general value is 0.5. $\min(\mathrm{depth}_{S_1}, \mathrm{depth}_{S_2})$ indicates the minimum value of S_1 and S_2 in the semantic tree.

3.2. Calculation of Concept Similarity. Through the semantic description of content word concepts, concept similarity is calculated through the following four types of sememe similarity:

(1) The first independent sememe description: calculate by using the formula, and write its similarity as $\mathrm{Sim}_1(S_1, S_2)$.

(2) Other independent semantic descriptors: Other independent semantic or stylistic words other than the first independent semantic. Since these independent sememe or specific words are extremely numerous, the similarity of these sememes after any pairing can be calculated by the formula above, and the group with the largest similarity can be extracted and divided into the same set. Then, the pair similarity of the remaining sememes can be iterated continuously. The loop ends when all of these primitives are sorted into different sets. Finally, the mean values of its sememes are calculated and taken as the similarity of independent sememes. Its similarity is denoted as $\mathrm{Sim}_2(S_1, S_2)$.

(3) Relational semantic descriptors: all expressions described by relational semantics in the semantic description. The similarity of the relation sememe is composed of the maximum value in the combination of the same relation sememe. Its similarity is denoted as $\mathrm{Sim}_3(S_1, S_2)$.

(4) Symbolic semantic descriptors: All expressions described by symbolic semantic descriptors in the semantic description. The similarity of the sememe is formed by the maximum value in the same sememe combination. Its similarity is $\mathrm{Sim}_4(S_1, S_2)$.

To sum up, the calculation formula of concept similarity is as follows:

$$\mathrm{Sim}(C_1, C_2) = \sum_{i=1}^{4} \beta_i \prod_{j=1}^{i} \mathrm{Sim}_j(S_1, S_2), \tag{5}$$

where C_1 and C_2 represent two concepts. β_i $(1 \le i \le 4)$ is an adjustable parameter and $\beta_1 + \beta_2 + \beta_3 + \beta_4 = 1, \beta_1 \ge \beta_2 \ge \beta_3 \ge \beta_4$; $\beta_1 = 0.5, \beta_2 = 0.2, \beta_3 = 0.17, \beta_4 = 0.13$.

The weight of each word is weighted $a/(a + p(w))$, where a is set to 0.01 and $P(w)$ is an estimated frequency.

3.3. Calculation of Word Similarity. If the concept of the word W_1 is $C_{11}, C_{12}, \cdots, C_{1n}$, and the concept of the word W_2 is $C_{21}, C_{22}, \cdots, C_{2m}$, then the value with the greatest similarity among all the concept combinations between them represents their similarity. Its similarity calculation formula is as follows:

$$\mathrm{Sim}(W_1, W_2) = \max_{i=1,2,\ldots,n, j=1,2,\ldots,m} \left(\mathrm{Sim}(C_{1i}, C_{2j})\right) \tag{6}$$

3.4. Algorithm Flow of Text Similarity

(1) Read text d_1 and text d_2.

(2) Preprocess the two texts with word segmentation and stopping words. The words d_1 contains are $d_1 = \{t_{11}, t_{12}, \cdots, t_{1n}\}$, and the words d_2 contains are: $d_2 = \{t_{21}, t_{22}, \cdots, t_{2m}\}$.

(3) The words contained in text d_1 and text d_2 are combined in pairs to form a word similarity matrix:

$$M = \begin{pmatrix} \mathrm{Sim}(t_{11}, t_{21}) & \mathrm{Sim}(t_{11}, t_{22}) & \cdots & \mathrm{Sim}(t_{11}, t_{2m}) \\ \mathrm{Sim}(t_{12}, t_{21}) & \mathrm{Sim}(t_{12}, t_{22}) & \cdots & \mathrm{Sim}(t_{12}, t_{2m}) \\ \vdots & \ddots & \vdots & \\ \mathrm{Sim}(t_{1n}, t_{21}) & \mathrm{Sim}(t_{1n}, t_{22}) & \ldots & \mathrm{Sim}(t_{1n}, t_{2m}) \end{pmatrix}, \tag{7}$$

where $\mathrm{Sim}(t_{1n}, t_{2m})$ represents the similarity between the n-th word in the text d_1 and the m-th word in the text d_2.

(4) The similarity value of each $\mathrm{Sim}(t_{1n}, t_{2m})$ in the similarity matrix is calculated using the semantic similarity algorithm based on words. That is, formulas (4), (5), and (6) are used for calculation.

(5) Find the maximum similarity value of words in the similarity matrix, denoted as $\mathrm{Max}(k)(k = 1, 2, \ldots)$ and record the row i and column j where the value resides. Max(k) was compared with threshold δ, if $\mathrm{Max(k)} \geq \delta$, the weight values of the two words in Max and the words in their respective texts were recorded, and then the i-th row and the j-th column to which Max belonged in the similarity matrix were deleted.

(6) Repeat the process of Step (5) until the matrix is empty or does not meet the conditions.

(7) According to Steps (5) and (6), the set of maximum matching combinations of word similarity can be obtained. Assuming that the length of the set is L, the set can be expressed as $\mathrm{Max}L = \{\mathrm{Max}(1), \mathrm{Max}(2), \cdots, \mathrm{Max}(l)\}$, and the similarity calculation formula of the two texts is

$$Net\_\mathrm{Sim}\left(d_i, d_j\right) = \frac{\sum_{k=1}^{l} \mathrm{Max}(k)}{l}. \tag{8}$$

4. Design of Automatic Exercises Grading System

4.1. Technical Architecture. The software technology architecture of the system is mainly divided into three layers: the information presentation layer, business logic layer, and database layer. This system is developed based on Django architecture, and the database system uses MySQL database with good storage stability and maintainability, the overall technical framework of the system is shown in Figure 3.

Information presentation layer: it is mainly an interface for interacting with users, and its function is to receive users' request information and display data. And students and teachers send requests to the back-end server by clicking the page function button. The back-end system receives the requests and processes the business logic and then returns the corresponding information to the front-end interface.

Business logic layer: This layer is the core of the whole system and the communication bridge between the data presentation layer and the information presentation layer. It is mainly used to receive the request of the front-end interface, process the corresponding business logic, and transmit the data down to the data layer. The business logic layer code of this system is written in Python, and the development framework is based on Django's three-tier architecture.

Database layer: it is mainly used to add, delete, change, and check data in database tables, and it is used to store and manage system-related data, to realize adding, deleting, changing, and checking data in the business logic layer. This system uses MySQL and Redis to store data and build a database server, which facilitates the query, modification, and storage of application layer data.

4.2. Functional Architecture. The grading model is mainly used to assist teachers in the evaluation of examination papers, and its prototype mainly includes data set collection, text preprocessing, feature extraction, similarity calculation, and subjective question scoring modules. The overall design structure of the system is shown in Figure 4.

(1) Text preprocessing module. It is mainly to mark the collected data and process the data to remove stop words and punctuation.

(2) Feature extraction module. It mainly extracts the text features of the examinee's answers and standard reference answers, mainly extracts the text features and semantic feature vectors of the candidates and reference answers, and stores their features and the scores of the corresponding texts in the database.

(3) Grading module. In this module, through the Chinese word segmentation model based on the fused dictionary information, a higher word segmentation result is obtained. After semantic similarity calculation, the text similarity between the examinee and the standard reference answer is obtained. Finally, it is weighted with the score to obtain the final score of the subjective question.

4.3. Workflow of Grading System

(1) Preprocess and train the data training set and wiki Chinese corpus set to obtain the test paper data training set and the word vector model of the wiki corpus.

(2) Vectorize the student answers to be graded and the corresponding reference answers in the test paper.

(3) Input the vector obtained in Step 2 into the network model fused with dictionary information for training, and obtain the segmentation results of students' answers and references.

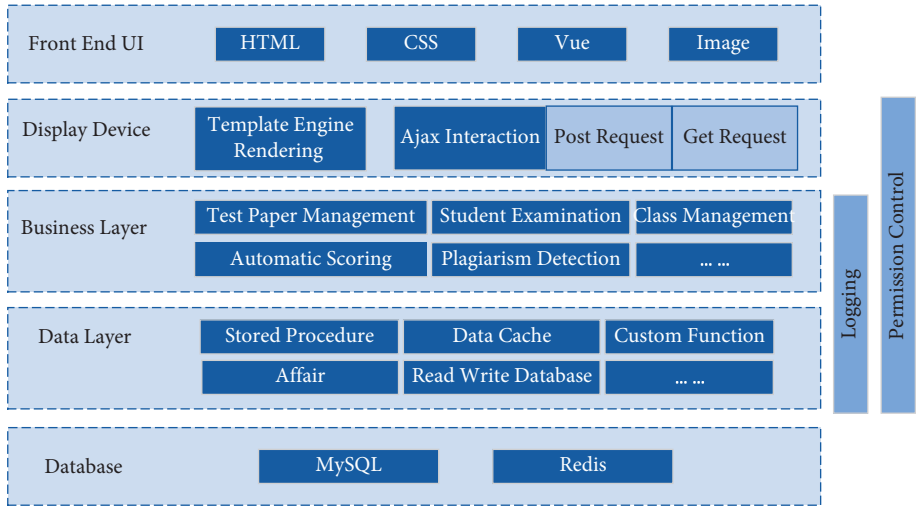

Figure 3: Technical architecture of the system.

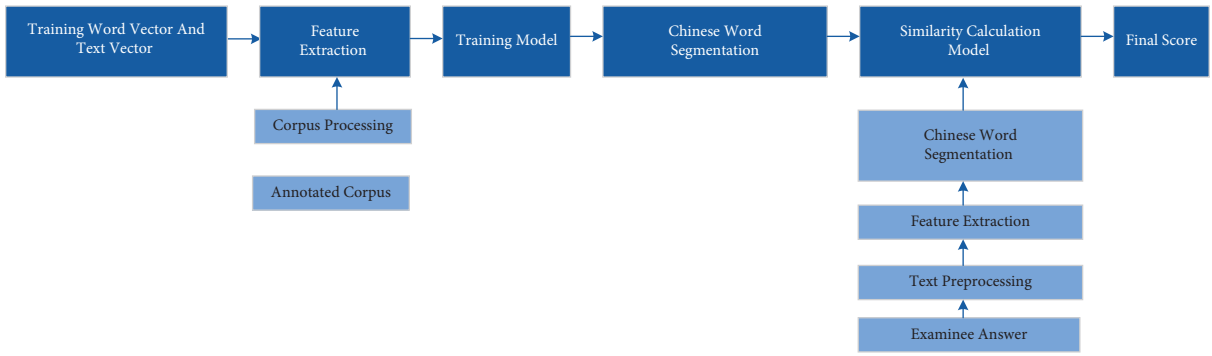

Figure 4: Functional architecture of the system.

(4) Determine the part-of-speech judgment of each word after obtaining the segmentation result. Using the text-similarity computing model proposed in Chapter 3 conceptualizes terms to obtain its term set, and then get the text similarity between the student answer to be graded and the reference answer.

(5) According to the total score of the test paper, the similarity is weighted with the total score of the test paper to get students' final scores.

5. System Testing

5.1. Functional Test

5.1.1. Testing Environment. The development language of this system is Python, the framework is based on Django, the database is MySQL, and the scoring module uses Gensim and Jieba. The specific testing environment is shown in Table 1.

5.1.2. Testing Methods. The testing methods used in this paper are mainly black box testing, compatibility testing, performance testing, and user interface testing. The specific test steps are as follows:

Table 1: Information on testing environment.

Testing environment	Configuration details
Server hardware environment	Intel (R) i5-8265U 16 G DDR4
Network bandwidth	More than 10M
Code running environment	PyCharm, Python3.6, Django2.2
Database	My SQL 5.7, Redis 3.2
Browser	Chrome, Microsoft Edge, IE11

(1) Black-box test: Test whether the functions of each module of the scoring system are available normally, find the errors of each module in time, and debug and modify the code. After the modification of the code is completed, the regression test is conducted to ensure that the modified code does not introduce new errors.

(2) Compatibility test: Considering the different ways users access the system, Google browser, Microsoft Edge browser, and IE browser is used to test the functions of the system.

(3) Performance test: Simulate a large number of users using the system at the same time, and test whether the response time of each function page of the system is within the acceptable range.

Table 2: Results of manual grading test.

Test module	Teacher grading
User role	Teacher
Test content	Grade the students' answers.
Precondition	Teachers have logged into the system.
Test step	Teachers select a test paper, review it, and input the score.
Expected result	Teachers can enter scores by clicking on the manual scoring input box.
Actual result	Consistent with expectations
Conclusion	Pass

Table 3: Test results of automatic grading.

Test module	Teacher grading
User role	Teacher
Test content	Teachers use automatic grading methods to grade students' answers.
Precondition	Students mock the exam and submit 80 data.
Test step	Teachers select a test paper and click "automatic grading"
Expected result	After the teacher clicks the automatic scoring, the scoring results of the algorithm will appear in the scoring box, and the error between the scoring result and the teacher's score is within ±2 points
Actual result	The data tested were within ±2 points.
Conclusion	There is some error in automatic scoring, but the error is within a reasonable range.

5.1.3. Test Results. According to the test method, design test cases and test the scoring module. The test results are shown in Table 2 and 3.

Therefore, the scoring function of the system can be used normally. Besides the functional test, the compatibility and performance of the system are also tested. The results show that the modules of the system, such as question bank management, test paper management, and automatic grading, can be used normally in different browsers.

5.2. Test of Grading Effect

5.2.1. Experimental Data. Generally, there are two ways to collect data sets. The first way is to use optical character recognition technology to extract the text from the test paper by scanning, and the second way is to manually input information. Because the correct rate of text input by OCR technology is not ideal, this paper uses manual input of candidates' answers, references, and similarities to complete the collection of data sets. The experimental data is a Chinese test paper data set of a middle school. According to 1000 samples of test paper, 2400 pieces of text data are collected, including students' answers, references, teachers' scores, and the total score of the questions. The text data is stored in CSV format, which is divided into four columns of data for storage, candidate number, student answer, reference, and the ratio (text similarity) between the teacher's score and the total score of the test questions. The 4:1 ratio of the data set is divided into the corresponding training set and test set.

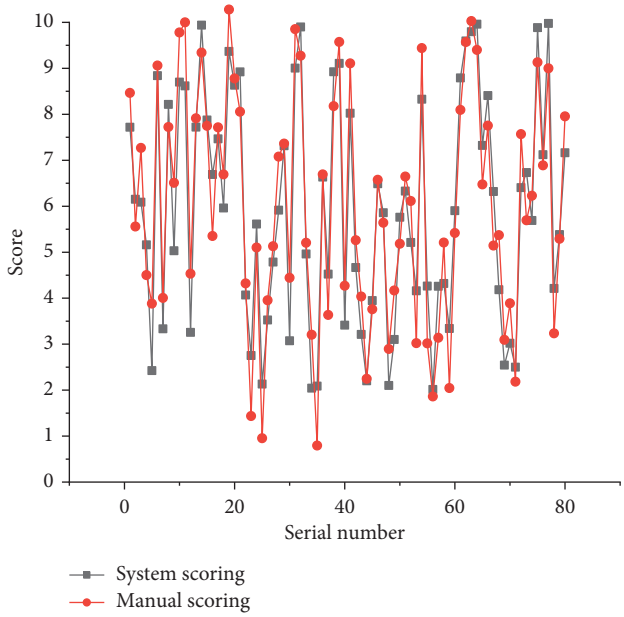

Figure 5: Comparison of grading results.

5.2.2. Test Results. Take a test set of reading comprehension as an example (score: 10 points), and compare its scoring results with manual scoring. The comparison results of the top 80 scores are shown in Figure 5.

As can be seen from the above figure, the exercise grading system designed in this paper based on similarity analysis under text has achieved relatively ideal scoring results to a certain extent, and there are some differences in the scoring results of some samples, which may be composed of the following two parts. One is that there are few improper word segmentations, and there may also be incomplete extraction of semantic feature information, the other reason may be that manual evaluation of subjective questions may lead to errors in subjective questions scoring due to personal subjective opinions.

6. Conclusion

By conceptualizing text terms, this paper designs the text-similarity algorithm, and according to the text term similarity computing model, an automatic exercise grading system is designed, including the design of technical architecture, the design of functional modules, and the realization process of comprehensive grading. The system function test results show that the scoring function of the system can be used normally; in addition, the test set of the test paper is selected for the experiment, and the automatic exercise grading system designed in this paper can achieve ideal grading to a certain extent. However, the system still needs to be improved in teacher-student interaction, and follow-up work can be carried out around this.

Data Availability

The dataset can be accessed upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] Z. Deng, *Design and Implementation of Multi-Question Automatic Grading Examination System*, Hunan University, China, 2016, in Chinese.

[2] Y. Ke, "Application of text clustering in college English composition automatic scoring system," *Electronic Technology and Software Engineering*, vol. 2, no. 05, p. 205, 2017, in Chinese.

[3] S. Zhang, *Research on Automatic Scoring Method of Short Text Subjective Questions Based on Text Similarity*, Wuhan University of Technology, Hubei, 2017, in Chinese.

[4] J. Cao, *Research and Implementation of Automatic Scoring System for Subjective Questions Based on Natural Language Processing*, Beijing University of Technology, Beijing, China, 2015, in Chinese.

[5] I. Zeroual and A. Lakhouaja, "Data science in light of natural language processing: an overview," *Procedia Computer Science*, vol. 127, pp. 82–91, 2018, in Chinese.

[6] J. Liu, M. Yu, and vilen, "Text similarity computing method based on sentence vector," *Science, Technology and Engineering*, vol. 20, no. 17, pp. 6950–6955, 2020, in Chinese.

[7] M. Tang, L. Zhu, and X. Zou, "A document vector representation based on Word2Vec," *Computer Science*, vol. 43, no. 6, pp. 214–217, 2016, in Chinese.

[8] P. A. A. Dumal, W. K. D. Shanika, S. A. D. Pathinayake, and T. C. Sandanayake, "Adaptive and automated online assessment evaluation system," in *Proceedings of the 2017 11th International Conference on Software, Knowledge, Information Management and Applications (SKIMA)*, pp. 1–8, IEEE, Malabe, Sri Lanka, December 2017.

[9] W. Sun and B. Zhang, "The development of natural language processing technology under the background of artificial intelligence," *Electronic Technology and Software Engineering*, vol. 33, no. 13, pp. 104-105, 2020, in Chinese.

[10] T. Qiu, C. Yu, Y. Zhong, An Lu, and G. Li, "A scientific citation recommendation model integrating network and text representations," *Scientometrics*, vol. 126, no. 11, 2021.

[11] E. Chen and E. Jiang, "Review of the research on text similarity computing methods," *Data Analysis and Knowledge Discovery*, vol. 1, no. 6, pp. 1–11, 2017, in Chinese.

[12] P. Zhang, "Sentence similarity calculation model based on multi-feature fusion," *Computer Engineering and Application*, vol. 46, pp. 136-137, 2010, in Chinese.

[13] S. Zhang, "Automatic scoring technology of subjective questions based on twin neural network," *Modern Computer*, vol. 26, no. 5, pp. 23–25, 2020, in Chinese.