Multiple Choice Question Generation Based on the Improved TextRank

Lai Wei[1,a], *Guosheng Hao[1,2,b], Xia Wang[1,c], Shuoshuo Meng[1,d], Xiaohan Yang[1,e], Yi Zhu[1,f]

{[a]flowclothesx@gmail.com, *[b]hgskd@jsnu.edu.cn, [c]lgzwx@163.com, [d]lovebxzd@vip.qq.com [e]2046835327@qq.com, [f]zhuy@jsnu.edu.cn}

School of Computer Science and Technology, Jiangsu Normal University, Xuzhou, China[1]
Jiangsu Wisdom-driven Research Institute Co., Ltd, Xuzhou, China[2]

Abstract. Currently, strategies for generating multiple choice questions (MCQ) seldom take the analysis of semantic and syntactic dependency features into consideration. A Chinese MCQ generation method is proposed based on the improved TextRank algorithm with semantic similarity and dependency relatedness to extract keywords, primarily entities, as knowledge points for MCQs. Verb weight is introduced to improve the accuracy of initial weight in keyword extraction to obtain knowledge points for MCQs more precisely in texts. Synonyms based on Word2Vec is used to generate distractors for MCQs, which are filtered to ensure that each distractor refers to a different entity. Experiments show that compared to human generated questions, the accuracy of identification is 59.5%, and the F_1 value is 0.58. The aspect of keyword extraction in the cloze questions generation task evaluation metrics shows some improvement. The calculated question difficulty exhibits a strong negative correlation with answer accuracy.

Keywords: improved TextRank; keywords extraction; MCQ Generation; question difficulty

1 Introduction

Question is an essential tool for evaluating students' level of knowledge mastery and comprehension, playing a crucial role in the learning process. MCQ is a very effective assessment way of questioning, and if creators of MCQs are well trained, MCQs can be quality assured [1].

MCQ, as an objective type of question, can prevent teachers from influencing grading due to subjective reasons, which is one of the advantages. Rao et al. [2] summarised several advantages of MCQs, in which the most significant one is the rapid assessment speed.

Goto et al. [3] pointed out that most existing questions are provided by experts, which may cost a lot of manpower and resources. Therefore, automatically generating questions based on textual resources is of great significance for enhancing the diversity of educational materials. In this paper, a Chinese MCQ generation method based on textual educational resources is proposed, using the improved TextRank algorithm, which can contribute to educational materials. This algorithm takes part-of-speech, word frequency, and semantic relationships into consideration, and introduces dependency syntactic analysis in combination with an improved probability transition matrix, providing a method for generating MCQs based on extracting keywords.

ITEI 2023, November 24-26, Zhengzhou, People's Republic of China

DOI 10.4108/eai.24-11-2023.2343612

The contribution of this paper includes: (1) The utilisation of the improved TextRank algorithm to enhance the effectiveness of extracting knowledge points for MCQs. (2) The implementation of generating, selecting, and filtering distractors using Synonyms. (3) The introduction of an evaluation metric for question generation based on the "Turing test" concept. (4) The proposal of an algorithm for calculating objective difficulty based on the options.

This paper is divided into three parts. Firstly, to enhance the quality of MCQ generation, the improved TextRank algorithm is utilised to increase the accuracy of keyword extraction [4], based on which the knowledge points can be obtained more accurately. Secondly, for the generating of distractors, synonyms generated by Synonyms are used to serve as distractors. Synonyms calculates the semantic distance between the generated synonyms and the keywords, allowing for the control of question difficulty based on semantic distance. Utilising the above methods makes the generated MCQs more flexible and adaptable to various contexts. Finally, in evaluating the quality of the MCQ generation, the Cloze Questions Generation (CQG) evaluation metric [5] is employed to evaluate the quality of generated MCQs. Additionally, a double-blind experiment that borrows the concept of the "Turing test" evaluates generated MCQs by the method in this paper and artificially generated. These experiments ensure that the MCQs generated by the method presented in this paper are more realistic and practical.

2 Related Work

2.1 Automatic MCQ Generation

Research on automatic MCQ generation can be dated back to 1997 when Coniam worked on English cloze test generation based on corpus word frequencies [6]. Subsequently, research in MCQ generation has covered various aspects. Mitkov et al. used corpora to identify significant concepts in the text and employed electronic documents as sources for generating MCQs [7]. Goto et al. [3,8] selected text from English textbooks as a source for generating cloze test questions and used conditional fields for keyword selection. Chen et al. [9], from a grammatical question perspective, collected real sentences from the web and applied generation strategies to transform the sentences into TOEFL-style questions. Sumita et al. [10] utilised Item Response Theory (IRT) to assess the English proficiency of users by generating English MCQs.

Annamaneni et al. [5] identified three primary components of MCQ generation: key sentence selection, keyword selection, and distractors generation. Subsequently, most MCQ generation research revolves around these three components.

For Chinese MCQ, Chu et al. [11]proposed using Wikipedia as a knowledge source for generating fact-based MCQs. Liu et al. [12]introduced a mixed similarity strategy to generate distractors based on lexical, phonetic, and semantic dimensions.

Currently, evaluating the quality of MCQ generation remains challenging. One reason is the lack of a unified and comprehensive set of MCQ generation standards, resulting in subjectivity, and making it difficult to achieve a consensus. Another reason is the difficulty in quantifying the quality of generated MCQs since the quality can only be obtained through feedback from test-takers. Rao et al. [2], through an analysis of recent research on MCQ generation, summarized six steps involved in the task and evaluated these steps based on these metrics. They also provided commonly used evaluation metrics for most MCQ generation, including key sentence and keyword selection metrics, and distractor metrics. Metrics for key sentence and keyword selection include question format, sentence length, sentence difficulty, contextual relevance, grammatical correctness, and so on. Distractors selection typically involves assessing the similarity between the distractors and the correct answer.

Rao et al. [2] also pointed out that the quality evaluation of most existing MCQ generation primarily relies on a combination of human feedback and specific questions, which indirectly acquires feedback from different perspectives and heavily depends on proprietary data. Therefore, to comprehensively evaluate the quality of MCQs more comprehensively, an evaluation metric is proposed, referring to the testing approach of machine learning models and the concept of the "Turing test". Building upon the work in [5], this metric quantifies the quality of generated questions by analysing the results of participants' identification of question sources.

2.2 The Improved TextRank Algorithm

In this paper, a keywords extraction algorithm based on the improved TextRank algorithm as proposed by Meng et al. [4] is used. This algorithm improves the initial weights of keywords based on multiple features. It also considers semantic similarity and dependency relatedness by using a probability transition matrix to describe the weights of keywords. Finally, these parameters are iteratively processed to calculate the final weights of words and provide a sorted list of candidate keywords. This algorithm improves the accuracy of keyword extraction, which directly impacts the quality of the MCQ generation. The following briefly outlines this algorithm.

The TextRank algorithm is a graph-based ranking algorithm used for keyword and document extraction. It is derived from Google's PageRank algorithm for ranking web pages. TextRank utilises co-occurrence information among words within a document, treating words as nodes in a graph and co-occurrence relationships as edges. It considers co-occurrence relationships but does not account for the semantic and syntactic relationships between keywords. To better extract keywords, the improved TextRank algorithm refines the TextRank algorithm from three aspects: part-of-speech tagging, semantic relatedness, and syntactic relatedness.

In the classic TextRank algorithm, the equation for calculating the weight of word W_i is as follows:

$$\mathrm{WS}(W_i) = (1 - d) + \left(d * \sum_{j \in \mathrm{In}(W_i)} \frac{\omega_{ji}}{\sum_{V_k \in \mathrm{Out}(W_j)} \omega_{jk}} \mathrm{WS}(V_j) \right) \tag{1}$$

where $\mathrm{WS}(W_i)$ represents the weight of the i-th node, which is the weight of the i-th keyword; $\mathrm{In}(W_i)$ represents the set of all nodes pointing to node W_i; $\mathrm{Out}(W_j)$ represents the set of nodes pointed to by node W_i; w_{ji} represents the transition probability from node W_i to node W_j; d is the damping factor, which prevents the weights from converging to zero after several iterations and is typically set to 0.85.

Meng et al. [4] introduced three probability matrices based on the classical TextRank algorithm, including part-of-speech weight, semantic similarity, and syntactic dependency relationships.

In TextRank, when calculating initial weights, each node's weight is generally set to $\frac{1}{N}$, where N represents the number of nodes, meaning that the initial weights for each node are the same. However, in [13], words of different part-of-speech are distributed differently. Nouns are the most prevalent, accounting for approximately 56.2% of the text, followed by verbs at about 22.7%. Adjectives are slightly less common, making up around 12.2%, while adverbs account for only about 4.4%. Other types of words make up approximately 4.5%.

Therefore, considering the varying distribution of words across different part-of-speech in the text and taking the outcomes of named entity recognition in natural language processing tasks into account, it is essential to consider the part-of-speech, especially the significance of proper nouns among nouns, when determining the initial weights. If a word is a noun (including nouns acting as adjectives, nouns acting as verbs, proper nouns, place names, organisations, and other proper nouns), its initial weight is set to 2. If the word is a verb, its initial weight is set to 1.5. If the word is an adjective, its initial weight is set to 1. All other part-of-speech are assigned an initial weight of 0.5.

By implementing this approach in constructing initial weights, the significance of different part-of-speech can be better utilised, increasing the likelihood that more important nouns in the text, particularly proper nouns, organizations, and personal names, will have a higher probability of being selected as the final results.

In constructing the probability transition matrix, the TextRank algorithm utilises co-occurrence relationships between words to build the matrix. In this paper, in addition to the classical TextRank algorithm, semantic and dependency relationships are also considered when constructing the probability transition matrix. This enables the matrix to better consider the relationships between words, improving the accuracy of keyword extraction.

In terms of semantics, Meng et al. [4] used a toolkit named Synonym, which is based on a Word2Vec trained word vector model to calculate semantic similarity between words. The semantic similarity is calculated between words in sentences and constructs a semantic similarity matrix:

$$S = \begin{bmatrix} w_{12} & \cdots & w_{1n} \\ \vdots & \ddots & \vdots \\ w_{n1} & \cdots & w_{nn} \end{bmatrix} \tag{2}$$

where each entry w_{ij} in the matrix represents the semantic similarity between the i-th and j-th words in the sentence.

Dependency parsing (DEP) is a task of natural language processing that analyses the grammatical relationships between words in a sentence and represents them as a tree-like structure. In that paper, by analysing the dependency syntax tree in the sentence, words in the sentence are treated as nodes and semantic dependency relationships are considered as edges.

For example, in a sentence, there is a dependency path between any pair of words. If two words are directly connected, there is a direct dependency path between them, and their dependency path length is 1. However, if two words are not directly connected, the length of the shortest path between them is their dependency path length.

Through the dependency path, it is possible to obtain the dependency relationship between words and calculate the dependency relatedness. If the dependency relatedness between two words is higher, their weights will be higher. Here is the dependency relatedness matrix:

$$D = \begin{bmatrix} w_{12} & \cdots & w_{1n} \\ \vdots & \ddots & \vdots \\ w_{n1} & \cdots & w_{nn} \end{bmatrix} \tag{3}$$

in this matrix, each entry w_{ij} represents the dependency relatedness between the i-th word and the j-th word in the sentence.

By integrating the semantic similarity matrix and the dependency relatedness matrix, a new probability transition matrix can be obtained, as shown in equation (4):

$$M = S + D \tag{4}$$

Substituted the probability transition matrix into the iteration equation as equation (5):

$$B_i = (1 - d) \times e + (d * B_{i-1} \times M) \tag{5}$$

where B_i represents the weights after the i-th iteration, B_{i-1} represents the weights after the $(i - 1)$-th iteration, and e represents a $1 \times n$ vector where all dimensions have a value of 1.

Based on equation (5), several iterations are performed. The iteration process continues until the difference between B_i and B_{i-1} is smaller than a specified threshold. At this point, B_i represents the weights of the obtained keywords. The weights are then sorted in descending order to obtain the keywords from the text.

3 MCQ Generation Based on the Improved TextRank Algorithm

The proposed method for MCQ generation consists of three main components, as shown in **Figure 1**.

(1) Based on the improved TextRank algorithm, combined with the weights of verbs in the sentences as weights for key sentences, this component selects key sentences and extracts the primary keywords from the sentences.

(2) For generating distractors, the component is responsible for aligning entities with the keywords. If a single-choice question is required, the component discards distractors with the same references. If a multiple-choice question is required, these distractors are retained.

(3) The final component combines keywords and distractors to construct the MCQs.

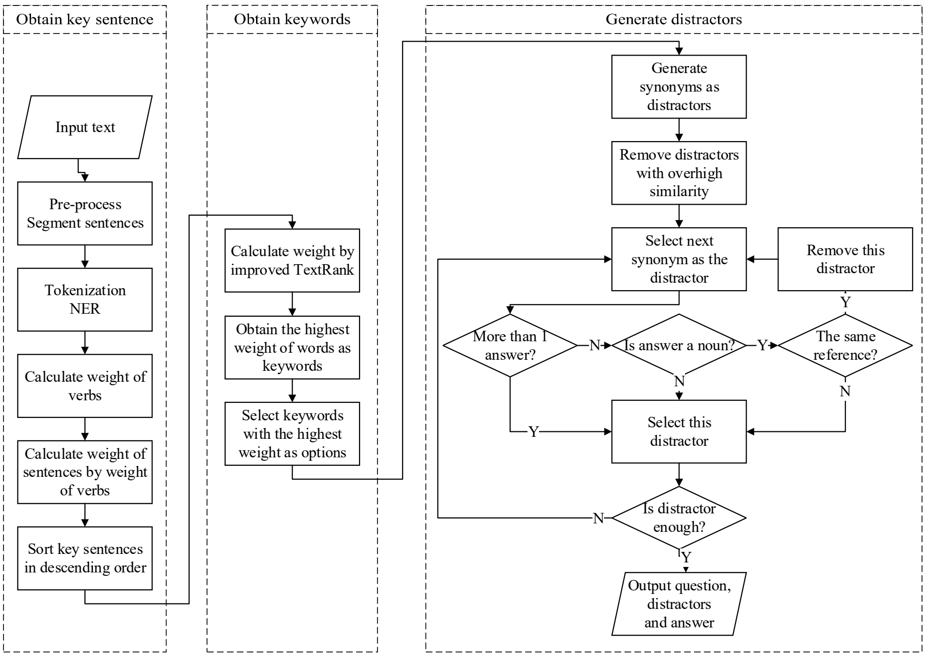
Obtain key sentence
Obtain keywords
Generate distractors
Input text
Pre-process
Segment sentences
Tokenization
NER
Calculate weight of verbs
Calculate weight of sentences by weight of verbs
Sort key sentences in descending order
Calculate weight by improved TextRank
Obtain the highest weight of words as keywords
Select keywords with the highest weight as options
Generate synonyms as distractors
Remove distractors with overhigh similarity
Select next synonym as the distractor
Remove this distractor
More than 1 answer?
Is answer a noun?
The same reference?
Select this distractor
Is distractor enough?
Output question, distractors and answer
Y
N

Fig. 1. Frame of MCQ generation based on improved TextRank.

(1) Based on the improved TextRank algorithm, combined with the weights of verbs in the sentences as weights for key sentences, this component selects key sentences and extracts the primary keywords from the sentences.

(2) For generating distractors, the component is responsible for aligning entities with the keywords. If a single-choice question is required, the component discards distractors with the same references. If a multiple-choice question is required, these distractors are retained.

(3) The final component combines keywords and distractors to construct the MCQs.

3.1 Key Sentences Selection Based on TextRank Algorithm

In this paper, for the selection of key sentences, which form the stems of MCQs, TextRank algorithm is used to calculate the sentence weights. The weights of verbs obtained from the improved TextRank algorithm are seen as the initial sentence weights. The reason why selects weights of verbs is based on the fact that the content under examination primarily revolves around nouns, and verbs are used to express actions, events, or states of existence, thereby determining the weight of sentences in the text.

Similar to the algorithm mentioned in section 2.2 for calculating word weights using the TextRank algorithm, the formula for calculating sentence weights with TextRank is as follows:

$$\mathrm{sim}(S_i, S_j) = \frac{\left|\{W_k \mid W_k \in S_i \,\&\, W_k \in S_j\}\right|}{\log(|S_i|) + \log(|S_j|)} \tag{6}$$

Here, S_i and S_j represent two sentences for which the similarity is being calculated, W_k represents words that appear in both sentences, and the denominator $\log(|S_i|) + \log(|S_j|)$ represents the sum of the logarithms of the number of words in both sentences.

By considering the weights of verbs obtained from the calculation in section 2.2, the most significant weights of verbs in each sentence are introduced into the adjacency matrix, and the probability transition matrix is computed as follows:

$$M_V = [v_1 \quad \cdots \quad v_n] \tag{7}$$

$$M_A = \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nn} \end{bmatrix} \tag{8}$$

Here, M_A represents the adjacency matrix, where each entry a_{ij} represents the similarity between sentence S_i and S_j, denoted as $\mathrm{Sim}(S_i, S_j)$. M_V represents the initial weights of sentences obtained based on the most significant weights of verbs in the sentences. Each entry v_i represents the initial weight of the i-th sentence. Subsequently, iteration is performed using equation (5) to calculate the final weights for each sentence. This process yields the $\mathrm{top}(N)$ sentence weights in the text, facilitating the selection of key sentences.

3.2 Keyword Extraction for MCQ Based on the Improved TextRank Algorithm

In this paper, building on the foundation of using the improved TextRank algorithm to extract keywords that served as knowledge points from text, the highest-weighted keyword in each sentence is selected to serve as the candidate option for generating MCQs.

Since MCQs primarily test learning levels and the ability to integrate information, the content assessed is often factual and focused on memorization. Therefore, the selection of options for MCQs should mainly revolve around nouns, such as proper nouns, event names, parameters, model names, and time-related terms.

In the improved TextRank algorithm used in this paper, the initial weight of keywords is changed. Special emphasis has been given to nouns, nominal phrases, and entities identified in named entity recognition, such as names of people, places, and organizations. Their weights have been increased to enhance the probability of these nouns, which are of particular relevance, becoming options for MCQs. The initial weights of nouns, as mentioned in section 2.2, have been redefined. For example, personal names, place names, and organizational names, as well as other proper nouns, have been set to 10. The redefinition ensures that nouns are the primary candidates for keywords while retaining the possibility of extremely important verbs serving as keywords to some extent.

3.3 Distractors Generation and Difficulty Calculation

Synonyms is a Chinese synonym toolkit that uses Word2Vec to train word vectors, which can be applied to text alignment, recommendation algorithms, similarity calculations etc. In this paper, the task of generating distractors is achieved using synonyms retrieval feature of Synonyms. Several synonyms for the keywords obtained in section 3.1 are generated through Synonyms to serve as distractors in the MCQs. Since word vectors are used to calculate the semantic distance between words, the correlation between the distractors and the correct answer can be controlled by selecting words with different semantic distances, allowing for the artificial adjustment of MCQ difficulty.

In general, the difficulty of MCQ is influenced by various factors, including: (1) The difficulty of the knowledge points covered in the MCQ, with MCQs addressing easier knowledge points generally being less difficult; (2) The distinctiveness of the options also affects MCQ difficulty—if all incorrect options are easily distinguishable, the MCQ will be less difficult, while if the options are difficult to differentiate, the MCQ will be more challenging; (3) If the MCQ requires additional background knowledge, its difficulty will increase; (4) The length of the stem text adds to the reading cost, and excessively long stem can make the task more difficult.

Denote the options as W_i and W_j, and their corresponding word vectors as V_i and V_j. In this context, cosine similarity is used to measure the distance between word vectors.

$$\mathrm{sim}(W_i, W_j) = \frac{V_i \cdot V_j}{\|V_i\| \cdot \|V_j\|} \tag{9}$$

For the n options in an MCQ, construct an $n \times n$ similarity matrix of options S, where each entry S_{ij} represents the similarity $\mathrm{sim}(W_i, W_j)$ between option W_i and option W_j. The difficulty $\mathrm{D}(W_i)$ of each option W_i can then be represented as follows:

$$\mathrm{D}(W_i) = \frac{\frac{1}{n}\sum_{i=1}^{n} \mathrm{sim}(W_i, W_j)_{(i \neq j)}}{\max_{(i \neq j)}\mathrm{sim}(W_i, W_j) - \min_{(i \neq j)}\mathrm{sim}(W_i, W_j)} \tag{10}$$

where $\frac{1}{n}\sum_{i=1}^{n} \mathrm{sim}(W_i, W_j)_{(i \neq j)}$ represents the average of semantic similarity between option W_i and other options, and $\max_{(i \neq j)}\mathrm{sim}(W_i, W_j)$ represents the maximum in semantic similarity between option W_i and other options, and $\min_{(i \neq j)}\mathrm{sim}(W_i, W_j)$ represents the minimum in semantic similarity between option W_i and other options.

The meaning of equation (10) is that the difficulty of option W_i depends on its similarity to other options and the range of similarity variations. The lower the average similarity, the less likely the option is to be confused, and therefore, it is less difficult. The greater the range of similarity variations, the higher the option's distinctiveness, and thus, it is less difficult.

Finally, the average of the difficulties of n options is the objective difficulty of the MCQ:

$$D = \frac{1}{n}\sum_{i=1}^{n} D(W_i) \tag{11}$$

At the same time, if the keyword is a noun and the MCQ is a single-choice question, the distractors cannot point to the same real-world object as the correct answer. To address this issue, a knowledge graph, CN-DBPedia, is employed for entity alignment. CN-DBpedia is a general-domain structured encyclopaedia developed by the Knowledge Works Research Laboratory at Fudan University, which extracts information from Chinese encyclopaedic websites and provides high-quality structured data [14]. All the "alias" relationships from CN-Dbpedia are extracted and consolidated into an entity alignment knowledge graph.

This entity alignment knowledge graph contains approximately 287,000 relationships and covers various domains, making it suitable for covering most mainstream nouns. With the integrated entity alignment knowledge graph, the generated distractors with the original words are compared in the text. If they refer to the same entity, the distractor is skipped and removed, and another synonym will be selected as the distractor. This approach enhances the quality of the distractors to some extent and mitigates the issue of repeated options.

4 Experiments and Result Analysis

In this paper, the quality of MCQ generation is evaluated through three aspects: (1) A double-blind experiment using the concept of the "Turing Test"; (2) Evaluation metrics for Cloze Questions Generation (CQG) tasks [5]; (3) Statistical analysis of answer accuracy and its correlation with MCQ difficulty as described in section 3.3.

The corpus for this study is based on the "Data Structures" course textbook. Several different MCQs were generated, and approximately 1000 pieces of feedback were collected through a survey. The experiment involved 100 participants primarily at the master's level of education from a Chinese university.

4.1 Double-Blind Experiment Using the "Turing Test" Concept

The Turing Test, proposed by British computer scientist Alan Turing in 1950, involves a test in which an interrogator (human) engages in a conversation with both a human and a computer, without knowledge of which is which. If, through a series of MCQs, the interrogator cannot distinguish between the computer and the human in more than 30% of the responses after multiple tests, the computer can be considered to have passed the Turing Test. The purpose of this test is to evaluate whether a machine can exhibit human-like intelligence.

Since most MCQs are currently generated artificially, this paper adopts the concept of the Turing Test, using both the MCQ generation method presented in this paper and human MCQ creators as the test subjects. Their task is to generate MCQs based on the same text. In the double-blind experiment, interrogators are presented with a set of MCQs derived from the same text, without being informed whether the MCQs were generated by computers or humans. The interrogators' task is to determine the source of each MCQ. In this experiment, the MCQs were based on the same text, and an equal number of MCQs were generated by human MCQ creators and the method presented in this paper.

Precision (P), recall (R), F_1-score, and accuracy (A) are used to quantify the interrogators' ability to distinguish between MCQs. The equation for calculating these three values are as follows:

$$P = \frac{\text{Correctly identified as computer} - \text{generated MCQs by participants}}{\text{Totally identified as computer} - \text{generated MCQs by participants}} \tag{12}$$

$$R = \frac{\text{Correctly identified as computer} - \text{generated MCQs by participants}}{\text{Totally computer} - \text{generated MCQs}} \tag{13}$$

$$F_1 = \frac{2PR}{P + R} \tag{14}$$

$$A = \frac{\text{Correctly identified MCQs by participants}}{\text{Total Number of MCQs}} \tag{15}$$

In usual circumstances, precision (P) and recall (R) are applied in machine learning to assess the accuracy of an algorithm. A good algorithm aims to balance both precision (P) and recall (R), striving for high values in both measures, thereby achieving a high F_1-score, indicating better performance. However, in this paper, as the identification is conducted artificially, the goal is to make the MCQs generated by computers as consistent as possible with those generated artificially, making them difficult to distinguish. Therefore, contrary to most research, this paper seeks to minimize precision (P), recall (R), and F_1-score in the double-blind experiment. The experimental results are presented in **Tables 1** and **2**.

From **Table 1**, it can be observed that in the double-blind experiment, human identification only achieved an accuracy of 59.52%, with 40.48% of the MCQs being identified incorrectly. This proportion exceeds the 30% error rate in Turing tests. In **Table 2**, when calculating precision (P), recall (R), and F_1-score for human identification, the values are merely 0.58, indicating a relatively low level of agreement. From this experiment, it can be concluded that the Chinese MCQs generated by the method presented in this paper can to some extent confuse human judgment and are challenging to distinguish from artificially generated MCQs. This suggests that the MCQs have practical value to a certain extent.

Table 1. Accuracy in the blinded experiment with Computer generated and Artificially generated.

Number	Correct Answer	Proportion of identification of C-generated	Proportion of identification of A-generated.	Accuracy
1	C-generated	47.619%	52.381%	47.619%
2	A-generated	42.857%	57.143%	57.143%
3	A-generated	33.333%	66.667%	66.667%
4	C-generated	52.381%	47.619%	52.381%
5	C-generated	66.667%	33.333%	66.667%
6	A-generated	28.571%	71.429%	71.429%
7	C-generated	52.381%	47.619%	52.381%
8	A-generated	33.333%	66.667%	66.667%
9	C-generated	80.952%	19.048%	80.952%
10	C-generated	23.810%	76.190%	23.810%
11	A-generated	23.810%	76.190%	76.190%
12	C-generated	38.095%	61.905%	38.095%
13	C-generated	38.095%	61.905%	38.095%
14	A-generated	42.857%	57.143%	57.143%
15	C-generated	47.619%	52.381%	47.619%
16	A-generated	28.571%	71.429%	71.429%
17	C-generated	71.429%	28.571%	71.429%
18	A-generated	19.048%	80.952%	80.952%
19	C-generated	47.619%	52.381%	47.619%
20	C-generated	76.190%	23.810%	76.190%

Table 2. P, R and F_1-score in the blinded experiment with the method presented in this paper.

Number	P	R	F
1	0.833	0.417	0.556
2	1.000	0.167	0.286
3	0.571	0.667	0.615
4	0.833	0.417	0.556
5	0.700	0.583	0.636
6	0.692	0.750	0.720
7	0.889	0.667	0.762
8	0.733	0.917	0.815
9	0.600	0.250	0.353
10	0.600	1.000	0.750
11	1.000	0.250	0.400
12	1.000	0.583	0.737
13	0.556	0.417	0.476
14	0.600	0.500	0.545
15	0.667	0.167	0.267
16	0.714	0.417	0.526
17	0.500	0.083	0.143
18	0.600	1.000	0.750
19	1.000	0.667	0.800
20	1.000	0.750	0.857

4.2 Evaluation Guidelines of Cloze Question Generation

Evaluation guidelines for Cloze Question Generation (CQG) tasks provided in reference [5] are employed in this paper. These guidelines comprehensively assess the choice, quality, and distractors in the MCQ generation task, considering four aspects, as shown in **Table 3**:

Table 3. Evaluation Guidelines of CQG.

Score	1	2	3	4
Sentence	Very informative	Informative	Remotely informative	Not at all informative
	Very relevant	Relevant	Remotely relevant	Not at all relevant
Keyword	Question worthy	Question worthy but span is wrong	Question worthy but not the best	Not at all question worthy
Distractor	Three are useful	Two are useful	One is useful	None is useful

The comparative results are presented in **Figure 2** as shown:

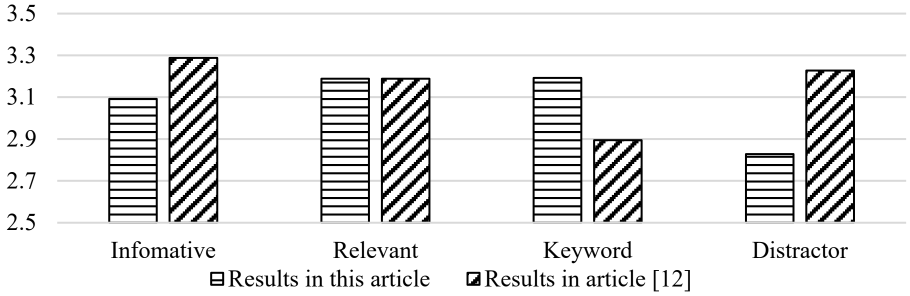

Fig. 2. Results comparison.

The results indicate that method presented in this paper can accurately identify key sentences, especially those that contain definitions, such as declarative sentences. In the aspect of keyword extraction, the improved TextRank algorithm, which considers the initial weights of keywords, enhances the recognition of proper nouns, personal names, and organizational names. Moreover, by incorporating the semantic distances between keywords and other words in the sentence, as well as the dependency syntax relationships within the sentence and integrating them into the probability transition matrix, the accuracy of obtaining keywords has been improved.

4.3 The Correlation between MCQ Difficulty and Answer Accuracy

The correlation between MCQ difficulty and answer accuracy is calculated in this study using Pearson's correlation coefficient. Pearson's correlation coefficient is a method for measuring the linear relationship between two variables. It represents the covariance between two variables divided by the product of their standard deviations. In this study, the two variables are MCQ difficulty and answer accuracy, which can be expressed as follows:

$$r = \frac{cov(D, A)}{\sigma D \times \sigma A} \tag{16}$$

Where D represents the MCQ difficulty, and A represents the answer accuracy for that question. The line chart is illustrated in **Figure 3**.

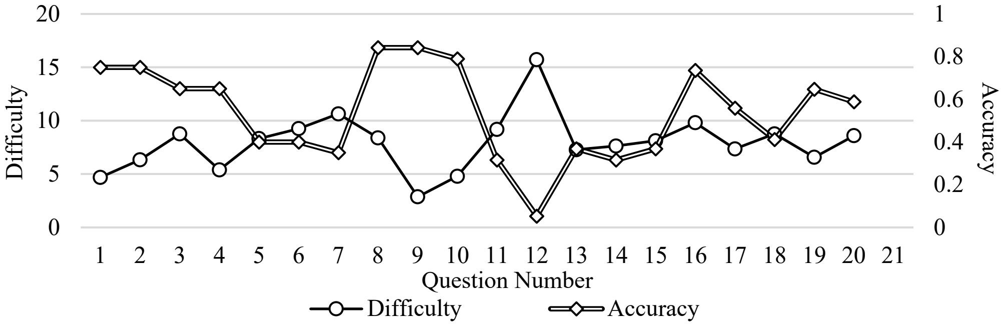

Fig. 3. Difficulty and accuracy of MCQs.

Substituting the experimental results into formula (16), it can be obtained that the Pearson correlation coefficient in this experiment is -0.663, with a significance level of 0.001. This indicates a strong negative correlation between the difficulty of the MCQs generated in this paper and the accuracy of the answers, and the significance level is less than 0.05, indicating a high degree of reliability.

4.4 Comprehensive Evaluation

Combining the results from experiments, the method presented in this paper has a significant advantage in keyword extraction, for better effectiveness in key sentence selection and keyword extraction from the text, especially important nouns such as names of people, places, and organizations, capturing the core of the text to generate MCQs. In the double-blind experiment, it effectively confuses the test participants, making it difficult to distinguish between MCQs generated by the method presented in this paper and those generated artificially. In the CQG evaluation guidelines, the method shows better results in extraction tasks. Additionally, an objective difficulty calculation algorithm for MCQs based on the distance between option word vectors is proposed, and it is verified through experiments that this difficulty is strongly negatively correlated with MCQ accuracy. This makes the method feasible and practical for specific applications.

5 Conclusions

Based on the improved TextRank algorithm for keyword extraction, a Chinese MCQ generation method is proposed in this paper. It has shown satisfactory results in keyword extraction. In the aspect of generating distractors, Synonyms based on word2vec is used to generate distractors in terms of semantic distance. Experiments show that this method is effective in extracting text keywords and key sentences, allowing the generated MCQs to closely align with the main topics of the texts. Moreover, the calculated MCQ difficulty matches real-world scenarios, which could assess the grasp of the initial text by the respondents better.

In future work, to provide an even better algorithm for calculating the objective difficulty of MCQs, further statistical analysis of the relationships between options is required. It is also essential to research the impact of option distribution on the psychology of respondents based on cognitive theories, including factors such as the respondents' mastery of knowledge and their subjective perception of MCQ difficulty. These factors also influence the subjective difficulty of MCQs.

Acknowledgement: The authors acknowledge the support from the National Natural Science Foundation of China under Grants 62277030 and 62077029, and Jiangsu Normal University Postgraduate Research and Practice Innovation Program Project 2022XKT1531.

References

[1] Downing, S.: Reliability: On the reproducibility of assessment data. Medical Education, 38(9), 1006-1012 (2004)

[2] Rao, C. D., & Saha, S. K. S.: Automatic multiple-choice question generation from text: A survey. IEEE Transactions on Learning Technologies, 13(1), 14-25 (2020)

[3] Goto, T., et al.: Automatic generation system of multiple-choice cloze questions and its evaluation. Knowledge Management & E-Learning: An International Journal, 2(3), 210-224 (2010)

[4] Meng, S. S., Hao, G. S., & Yang, Z. H.: Multi-feature fusion TextRank algorithm for sentence-oriented keyword extraction. In 2nd International Conference on Algorithms, High Performance Computing, and Artificial Intelligence (pp. 12-18). Guangzhou (2022)

[5] Annamaneni, N., Agarwal, M., & Shah, R.: Automatic cloze-questions generation. In Proceedings of the International Conference Recent Advances in Natural Language Processing (pp. 511-515). Bulgaria: INCOMA (2013)

[6] Coniam, D.: A preliminary inquiry into using corpus word frequency data in the automatic generation of English language cloze tests. Calico Journal, 15-33 (1997)

[7] Mitkov, R., Ha, L. A., & Karamanis, N.: A computer-aided environment for generating multiple-choice test items. Natural Language Engineering, 12(2), 177-194 (2006)

[8] Goto, T., et al.: An automatic generation of multiple-choice cloze questions based on statistical learning. In Proceedings of the 17th International Conference on Computers in Education (pp. 415-422). Asia-Pacific Society for Computers in Education. (2009)

[9] Chen, C. Y., Liou, H. C., & Chang, J. S.: FAST – An automatic generation system for grammar tests. In Proceedings of the COLING/ACL 2006 Interactive Presentation Sessions (pp. 1-4). Association for Computational Linguistics (2006)

[10] Sumita, E., Sugaya, F., & Yamamoto, S.: Measuring non-native speakers' proficiency of English by using a test with automatically-generated fill-in-the-blank questions. In Proceedings of the second workshop on Building Educational Applications Using NLP (pp. 61-68). Ann Arbor: Association for Computational Linguistics (2005)

[11] Chu, M. H., Chen, W. Y., & Lin, S. D.: A learning-based framework to utilize E-Hownet ontology and Wikipedia sources to generate multiple-choice factual questions. In 2012 Conference on Technologies and Applications of Artificial Intelligence (pp. 125-130). IEEE Computer Society (2012)

[12] Liu, M., Rus, V., & Liu, L.: Automatic Chinese multiple choice question generation using mixed similarity strategy. IEEE Transactions on Learning Technologies, 11(2), 193-202 (2018)

[13] Zhang, J. E.: Method for the extraction of Chinese text keywords based on multi-feature fusion. Information Studies: Theory & Application, 36(10), 105-108 (2013)

[14] Xu, B., et al.: CN-DBpedia: A never-ending Chinese knowledge extraction system. In International Conference on Industrial, Engineering, and Other Applications of Applied Intelligent Systems (pp. 428-438). Springer, Cham (2017)