

PAPER • OPEN ACCESS

An Intelligent Test Paper Generation Method to Solve Semantic Similarity Problem

To cite this article: Hongman Wang and Weihai Yang 2019 *IOP Conf. Ser.: Earth Environ. Sci.* **252** 052126

View the [article online](#) for updates and enhancements.

An Intelligent Test Paper Generation Method to Solve Semantic Similarity Problem

Hongman Wang^{1, a}, Weihai Yang^{2, *}

¹Institute of Network Technology; Engineering Research Center of Information Network Beijing University of Posts and Telecommunications; Ministry of Education Beijing, China

²Institute of Network Technology Beijing University of Posts and Telecommunications Beijing, China

*Corresponding author e-mail: why1024@bupt.edu.cn, ^awanghm@bupt.edu.cn

Abstract. In order to solve the problem of semantic duplication in the Intelligent Test Paper Generation Method, Genetic Particle Swarm Optimization Algorithm was used to search multi groups of test papers that conform to the constraints of the test. Then the idea of density entropy was used to screen the test papers, so that test papers will cover more questions uniformly. After that, analyze the semantic similarity of the test papers that use the *TextRank* algorithm to extract the key words of test questions, and use knowledge-keywords weighted VSM model to calculate the semantic similarity of test questions. Eliminate the high repetition rate of the test papers and avoid the repetition of the inspection points. The experimental results show that the algorithm can solve the semantic similarity problem in intelligent test paper generation of mass item bank.

1. Introduction

Intelligent test paper generation is one of the core functions of the online examination system. Over the years, domestic and foreign research institutes and scholars have studied some methods of intelligent test paper generation. In the early research, many simple algorithms such as random selection method, backtracking heuristic method and priority (hierarchical processing) method have been applied [1], but they have great limitation in application scenarios and computing resource consumption. In recent years, most of the research is based on bionics algorithm to solve the multi-dimensional constraints of intelligent test paper generation and improve the speed of test paper generation [2][3].

With the increase in the scale of item bank, there are some shortcomings in the existing intelligent test paper method. In the process of intelligent test paper generation, the same semantics and different types of questions often appear in the same test paper, which is the problem of the duplication of test questions.

In the absence of semantic analysis in the existing generating test paper method, it is difficult to avoid the semantic similarity problem in test papers. Semantic similarity measurement has always been an essential part of natural language processing and information retrieval. Similarity calculation methods of short text mainly base on the semantic dictionary, large data sets and rich context



information by a search engine. Aiming at the requirement of time-limit and massive-short-text, a weighted model of subject feature keywords based on knowledge points is proposed to calculate similarity quickly and accurately, which belongs to the similarity calculation method of short texts based on a large number of data sets.

In order to solve the problem of semantic similarity, this paper screens the selected test papers through density entropy [4] to achieve a high degree of distribution, and then applies the calculation of semantic similarity to the intelligent test paper generation method, so that this paper can satisfy the semantic rationality.

2. The Key Issue

2.1. Research Goal

The intelligent test paper generation method proposed in this paper can be divided into four stages. The first stage: item bank preprocess, filter out the bad test questions that are not suitable for participating in the test paper generation. The second stage: choose test questions to generate test papers, select multiple sets of test papers to meet the examination requirements from the large scale of item bank. The third stage: screen test paper according to the density entropy, the multiple sets of test papers are screened according to the density entropy of the test papers to ensure a reasonable distribution. The last stage: screen test paper according to semantic similarity, calculate the similarity of each test paper and obtain quantitative indicators.

This paper focuses on research and design an effective test paper generation method which chooses test questions from the large scale of item bank to meet the test requirements and have no semantic similarity of the knowledge points. Compared with the existing methods of generating test paper, the advantages of this paper are as follows:

Semantic similarity between test questions is calculated, so the test paper generated by this method is more reasonable.

A test paper density entropy model is designed to ensure the distribution of test papers.

In large scale of item bank, this paper only semantic analysis test papers after screening, and it takes less time to calculate. Although the method proposed in this paper has several more stages of calculation than the existing method, the total time is in line with the actual requirements.

2.2. An Intelligent Test Paper Generation Method to Solve Semantic Similarity Problem

2.2.1. Modeling and Algorithm. Theoretical basis. According to the principle of IRT (item response theory) [5], which is widely used in the field of educational testing, the constraints of intelligent test paper generation method should include difficulty index, discriminative power index and guessing index. Among them, the discriminative power index indicates the degree of discrimination of one question in the test to different levels of subjects. Guessing index indicates the probability of a question being guessed. There are two kinds of influencing factors: the objective factor of the recent appearance of the test paper, and the subjective factor of the content of the test question itself. For objective factors, we need to control the exposure time (the last time test question appeared in test paper); for subjective factors, we need to continuously optimize the question bank by improving the quality of test questions, which has nothing to do with this paper.

2.2.2. Modelling. In the stage of choose test questions to generate test papers, the following constraint condition may be considered:

Knowledge points: a1. Total score: a2. Difficulty degree: a3. Question type constraint: a4. Answer question time: a5. Exposure time: a6. Discrimination degree: a7.

Among them, a5 and a6 are attributes indicating the quality of the individual test question. Therefore, the filtering in the stage of item bank preprocess can satisfy the constraints of a6 and a7, and this paper is not limited.

To sum up, the index system in this paper presents as $S = \{a1, a2, a3, a4, a5\}$. Thus, all constraints of a test paper are composed of n attributes (here $n = 5$) of each m questions, which can be expressed by the objective matrix D .

$$D = \begin{pmatrix} a_{1,1} & a_{1,2} & \dots & a_{1,n} \\ a_{2,1} & a_{2,2} & \dots & a_{2,n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{m,1} & a_{m,2} & \dots & a_{m,n} \end{pmatrix}$$

2.2.3. Algorithm. In the stage of choose test questions to generate test papers, some scholars have proposed feasible models and algorithms. Literature [2] presents an intelligent test paper generation method based on genetic particle swarm optimization, which can solve the problem of generating papers from a large scale of item bank. According to the model proposed above, this paper reduces the dimension of the existing methods and has a faster iteration speed. The method is described as follows:

2.3. Part 1. Particle swarm optimization algorithm to initialize test paper population.

Step 1: set the initial parameters to generate the initial population. The parameters include particle velocity, location, population quantity, and the threshold of velocity and position.

Step 2: calculate fitness values and update the location and speed of particles.

Step 3: if a better solution is found, the individual optimal solution is updated; if the whole population finds a better solution, the population optimal solution is updated.

Step 4: Determine whether the result satisfies the end condition (the maximum number of iterations or the fitness value). If the end condition is not satisfied, return to step 2. If the end condition is satisfied, exit the iteration and output the optimal solution.

2.4. Part 2. Segmented adaptive genetic algorithm for solving new species groups

Step 5: segmented crossover, single point crossover will be performed by segment according to probability.

Step 6: segmented mutation, set mutation point for each segment, and replace another question with the same knowledge point and the same question type.

Step 7: select, according to the fitness value calculated, select the test paper with high fitness as the next generation population.

Step 8: iterate, if the constraints are met, then output; otherwise return to step 5.

2.5. Density entropy measurement of test papers

Suppose that the test papers P produced after the stage of choose test questions to generate test papers is composed of test questions q , q_{ij} represents each question in P , $1 \leq i \leq n$, $1 \leq j \leq m$, n represents the number of P , m represents the number of q , $q_{ik} \neq q_{il}$ ($1 \leq k, l \leq m, k \neq l$).

Suppose that the collection of all questions in test papers P is Q , $Q = \{q | q \in P_i, 1 \leq i \leq n\}$, and suppose that $Q_r = \{q | q \in P_i, 1 \leq i \leq n, i \neq r\}$. Q_r Represents the collection of all questions except test papers P_r . Then the density entropy (H) of test papers P_r is:

$$H_r = -\log \frac{|Q - Q_r|}{m}$$

$|Q-Q_r|$ Represents the number of questions that appear in the test papers P_r only. H_r Represents the importance of a test paper r in all papers.

At this stage, the test papers with the maximum density entropy are deleted to maintain the reasonable distribution of the paper population, and finally retain α sets of papers.

2.6. Semantic similarity model of test papers

The test paper is a whole composed of several questions. The stem and options of a question are called question content, and they are the input of the semantic similarity model. Because of the insufficient background knowledge and limited information in the short text, this paper uses keyword vectors to represent the information of the question content, so the similarity of the test questions can be described by the similarity between weighted keyword vectors. The problem of solving the semantic similarity of test papers is transformed into two problems: extracting the keywords and their weights and calculating the similarity between keywords vectors.

2.7. Keyword extraction

Many scholars have studied the keyword extraction model a lot. The main methods can be roughly divided into four categories: statistical model, graph model, latent semantics model and integer programming model [6] [7]. Because of the characteristics of the test paper, TextRank algorithm is used to extract keywords.

TextRank is an unsupervised keyword extraction algorithm. By dividing the text into several phrases and building a graph model, using the voting mechanism to sort the phrases of the text, this paper can use the information of the text itself to extract keywords. The steps for keyword extraction are described as follows [8]:

Step 1: Divide the question content C into a series of complete sentences. $C = [S_1, S_2, \dots, S_m]$

Step 2: For each sentence, word segmentation and part-of-speech tagging are carried out, and common words and stop words are filtered out. Nouns, verbs, and adjectives are phrases that can distinguish the meaning of the test questions, and only these meaningful phrases are retained. $S_i = [w_{i1}, w_{i2}, \dots, w_{in}]$, w_{ij} represents candidate keywords retained.

Step 3: Constructing the candidate keywords graph $G = (V, E)$, V represents node collection composed of candidate keywords generated by step 2. E Represents edge collection. The edge of any two nodes exists only and only if the two nodes are in the presence of co-occurrence, which the sequence length is K .

Step 4: For a node V_i in figure G , $In(V_i)$ represents a collection of nodes that point to the node i , $Out(V_i)$ represents a collection of nodes that the node i point to. The score of point V_i can be calculated as follows:

$$S(V_i) = (1-d) + d \times \sum_{V_j \in In(V_i)} \frac{\omega_{ij}}{\sum_{V_k \in Out(V_j)} \omega_{jk}} S(V_j)$$

d Is damping coefficient; ω_{ij} is the weight from V_i to V_j , represents the number of co-occurrence of the two keywords. The weight of any point begins to the same default weight, iterates the formula 3, and propagates the weight of each node until convergence.

Step 5: Normalizing the scores of nodes, sorting in reverse order to get the most important M phrases which make as candidate keywords.

2.8. Keyword weighting model based on knowledge points

Test questions are the inspection of knowledge points. Therefore, the questions with the same knowledge point have the same semantic theme. On the one hand, the weight of keywords depends on the frequency of keywords, on the other hand, it is influenced by the semantic theme. This keyword weighting model based on knowledge points can more truly reflect the semantic environment of test papers.

Suppose that questions with same knowledge point are Q_1, Q_2, \dots, Q_n , K_n represents collection of all keywords in any question Q_i , $K_i = [w_{i1}, w_{i2}, \dots, w_{it}]$ W_{xy} represents the number of times the keyword w_{xy} appeared. The theme weight of the key word is $f_{xy} = \frac{W_{xy}}{n}$, $f_{xy} \in (0, 1]$. The weight coefficient is calculated as follows:

$$\lambda = \frac{2}{e^{k(1-2f_{xy})} + 1}$$

k is intervention coefficient which influences degree of control weight; and $\lambda \in (0, 2)$. Suppose that f_{raw} is the weight of keyword by keyword extraction, finally the value of weighted keywords is:

$$f = \lambda \cdot f_{raw}$$

2.9. Vector Space Model (VSM) based on weighted keywords

The weighted keywords sequence of test question is $T = [w_1 : f_1, w_2 : f_2, \dots, w_n : f_n]$. For any two question text A and B , the collection of all phrases is $S = \{w | w \in T_A \cup T_B\}$ and weight values is

$$V = \{f | f \in T_A \cup T_B\}$$

And $V_A = \{f_i | \forall f_i \in V, \text{if}(w_i \in T_A) f_i = f_i \text{ else } f_i = 0\}$, $V_B = \{f_i | \forall f_i \in V, \text{if}(w_i \in T_B) f_i = f_i \text{ else } f_i = 0\}$.

According V_A and V_B above, the formula for the similarity of test questions is as follows

$$\text{Similar}(A, B) = \frac{\sum_{i=1}^m VA_i \times VB_i}{\sqrt{\sum_{i=1}^m (VA_i)^2} \times \sqrt{\sum_{i=1}^m (VB_i)^2}}$$

2.10. Quantification of test paper similarity

The similarity between individuals determines the overall similarity of the test paper and affects the quality of the test paper. In practice, most of the test questions have uniform knowledge points and low similarity, but the test paper is unacceptable once high similarity exists.

Suppose $x = \text{Similar}(A, B)$, similarity sequences for all test papers is $X = x_1, x_2, \dots, x_n$.

$E(X) = \frac{\sum_{i=1}^n x}{n}$, $S(X) = \sqrt{\frac{\sum_{i=1}^n (x_i - E)^2}{n}}$. The quantification of test paper similarity is as follows:

$$F = \omega_1 \times S(X) + \omega_2 \times MAX(X)$$

ω_1 , ω_2 are weight coefficients. The greater the F , the higher the repetition rate. At this stage, the test papers with the greatest F are deleted to satisfy semantic requirements, and finally retain β sets of papers.

3. Implementation And Verification

3.1. Implementation

Firstly, this paper filters the item bank and generates multiple sets of test papers that meet the constraints of the test. Secondly, it screens the test papers that can keep the distribution of the test questions. Then, it analyses the semantic similarity of the screened test papers, eliminates the test papers with high repetition rate, and obtains the satisfactory test papers.

When the number of questions in item bank is enough, the method can produce multiple sets of qualified papers; when the number of questions is insufficient, the algorithm will not trigger the calculation of screening and eliminating papers, does not affect the number of test papers, and has better versatility.

3.2. Computation process

The process description and parameters of this paper are as follows.

1	BEGIN
2	Input: Knowledge point, Test score, Difficulty degree, Question type, Test time
3	ItemBank -> filter (differentiation<0.5, exposure-Time <1month)
4	adopt the particle swarm optimization to initialize the population
5	adopt the genetic algorithm to get a new population
6	if (not meet constraints) goto 5
7	result= {paper_1, paper_2, paper_n}
8	if (result. Size <= α) goto 11
9	result -> density Filter ($H < \log_2 \frac{3}{2}$)
10	result= { paper_1', paper_2', paper_n ' }
11	if (result. Size <= β) goto 14
12	result -> similar Filter (K=5, M=8, k=3, $\omega_1=0.4$, $\omega_2=0.6$)
13	result= {paper_1'', paper_2'', paper_n '' }
14	output result
15	END

3.3. Optimization in computation

3.3.1 Permanently store keywords and its weights of test questions. Keyword extraction is a part of the algorithm, but it takes a lot of time to calculate. When the test questions are added, the keywords and their frequency information are calculated and stored. Therefore, the data can be read directly when calculating.

3.3.2 Optimization of similarity calculation between test questions. The themes of the same knowledge points are the same, so it is necessary to solve the problem of semantic similarity; there is no semantic repetition in different knowledge points. Therefore, we only need to compare the similarity of the same knowledge points to reduce the calculation time.

3.3.3 Caching test question similarity calculations. In the process of calculating the similarity of test papers, it is possible to calculate the similarity of the same test questions many times, so caching the calculation results can avoid repeated calculation.

3.4. Verification

In order to verify effects of the proposed intelligent test paper generation method to solve the problem of semantic similarity, this paper adopts Computer Network course as test data, the item bank has more than ten thousand questions. Literature [9] presents a good idea for verification, the author uses python programming language to implement the algorithm in this paper. The part of word segmentation and keyword extraction is implemented by jieba, which is a Chinese word segmentation Library. Computing optimization includes pre-computing and pre-storing keywords and weight information of the test questions, and caching the similarity calculation results into memory.

Attribute information input of experimental verification includes: list of knowledge points, total score (100 points), difficulty (set to 0.50), type of test questions (including single choice questions, multiple choice questions, true and false questions, blank questions, comprehensive questions), and answer time (90 minutes).

3.5. Time-consuming comparison of test paper generation

In order to analyze the time-consuming situation of each stage of the algorithm, the experiment chooses different number of test questions and changes the parameters to control the number of test papers. After times of calculations, the average time-consuming of the algorithm is obtained, which is shown in Table I below.

Table. 1 Test paper number and consuming time of algorithm generation

Number of questions involved	Number of papers generated in the second stage	Number of papers screened by density entropy	Number of papers obtained after semantic analysis	Computation time (Unit: sec.)
10k	10	4	1	9.65
2k	10	4	1	5.99
2k	5	3	1	3.29
1k	5	3	1	1.47

As can be seen from Table I, with the increase of the number of test questions, the time-consuming of the algorithm mainly consumed in the second stage, that is, the stage of choosing test questions to generate test papers. This is mainly due to the increase in the number of questions involved in the calculation, and the overall time-consuming is acceptable.

Next, this paper compares the proposed method with the traditional intelligent test paper generation method based on genetic particle swarm optimization. Each parameter is set with the same input, and the average results are shown in Table II below.

Table. 2 Comparison between this algorithm and the traditional test paper generation algorithm

algorithm	Number of questions involved	Computation time (Unit: sec.)	Quantification of examination papers F
An Intelligent Test Paper Generation Method to Solve Semantic Similarity Problem	1k	1.33	0.0221
	10k	8.96	0.0235
The traditional intelligent test paper generation method based on genetic particle swarm optimization	1k	0.38	0.0281
	10k	7.28	0.1936

As can be seen from Table II, the method proposed in this paper consumes more time than the traditional intelligent test paper generation method, but because this paper reduces the constraint dimension of test paper generation, the overall time-consuming is little worse. When the number of item bank increases, this paper can maintain the quality of test papers, and the time-consuming gap becomes smaller. It proves that this algorithm is suitable for solving the problem of semantic similarity in large scale of item bank.

3.6. Quality comparison of test paper generation

In order to show the influence of semantic similarity, the author randomly selected the test questions in the experiment. The examples and their calculation data are shown in Table III below.

Table. 3 Comparison of semantic similarity measurement

Type	Question content	Keywords	Similar
Test questions with semantic repetition	The roles of the network layer include (ABC)? A. Route Selection B. Flow Control C. Congestion Control D. Error Detection	control, flow, congestion, network layer, include, roles, selection, route	0.1594
	The functions of the network layer include route selection, flow control, congestion control and error detection. (×)	control, flow, include, selection, route, congestion, functions, detection	
Test questions without semantic repetition	In the OSI architecture, the actual communication is performed in (A) entities? A. Physical Layer B. Link Layer C. Network Layer D. Transport Layer	link layer, network layer, actual, perform, entity, architecture, physical layer, transport layer	0.0693
	Which layer in the TCP/IP model architecture corresponds to the physical layer and link layer of the ISO-OSI model (A)? A. Network Interface Layer B, Transport Layer C, Internet Layer D, Application Layer	link layer, physical layer, transport layer, internet layer, model, corresponds	

As can be seen from Table III, the proposed algorithm can accurately calculate the test questions with semantic similarity.

4. Conclutions

This paper mainly studies an intelligent test paper generation method to solve the problem of semantic similarity. Firstly, this paper puts forward the research goal of this paper. Then, establishes a mathematical model for the intelligent test paper generation and describes the specific algorithm. At last, this paper analyses and compares the feasibility of the intelligent test paper generation method proposed in time consumption and test paper quality. Experiments show that the proposed intelligent test paper generation method can effectively avoid the semantic similarity problems in the intelligent test paper generation of a large scale of item bank, and has good application value.

Acknowledgments

This work is supported by the Fundamental Research Funds of Beijing University of Posts and Telecommunications (No.2018XKJC07).

References

- [1] Ma F, Dong Y, and Shi J. "Intelligent test paper generation system based on slicing processing," in *Intelligent Control and Auto-mation*, IEEE, 2012, pp.506-511.
- [2] Hu Zhenfeng , and Xing Chunxiao. "An Intelligent Test Paper Generation Method Based on Genetic Particle Swarm Optimization," in *Web Information Systems & Applications Conference*, IEEE, 2017, pp.188-194.
- [3] Liu Yanyang. "Research and Implementation of Intelligent Test Paper Composition system based on Knowledge Key Point," Hebei University of Technology, 2009.
- [4] Song Wu, and Zheng Jinhua. "MOPSO algorithm based on density entropy," *Computer Engineering & Applications*, 43.24, pp.41-44, 2007.
- [5] Zhan Muqing, and Lu Ronghua. "Item Response Theory Model," *Science & Technology Information*, Vol.15, pp.28-28, 2009.
- [6] Su Hongyu. "Research and Application of Text Similarity," Xiamen University, 2015.
- [7] Li Kui. "Research on Short TeXt Semantic Similarity Computation," Harbin Engineering University, 2016.
- [8] Lu Yonghe, and Li Yanfeng. "Improvement of Text Feature Weighting Method Based on TF-IDF Algorithm," *Library and Information Service*, Vol.57, pp.90-95, 2013.
- [9] Wang Hongman, et al. "Finding Main Causes of Elevator Accidents via Multi-Dimensional Association Rule in Edge Computing Environment," *China Communications*, Vol.14, pp.39-47, Nov. 2017.