Paradigm Academic Press
Innovation in Science and Technology
ISSN 2788-7030
SEP. 2023 VOL.2, NO.5

Analysis and Evaluation of Text Comparison Based on Intelligent Optimization Algorithm

Zixian Fang[1], Jiayi Wang[1] & Fenglan Luo[2]

[1] School of Economics, Management and Law, Jiangxi Science and Technology Normal University

[2] Associate Professor, School of Economics, Management and Law, Jiangxi Science and Technology Normal University

Correspondence: Fenglan Luo, Associate Professor, School of Economics, Management and Law, Jiangxi Science and Technology Normal University.

doi:10.56397/IST.2023.09.02

Abstract

Text transcription is crucial in Chinese information processing. Text transcription has always existed since ancient times, but no matter whether it is manual transcription in ancient times or modern transcription using communication and storage devices, random errors cannot be avoided when a message has been forwarded and transcribed many times. In this paper, we study how to measure the size of differences between different versions of texts, how to estimate the number of transmissions experienced between two texts, and how to design an effective and fast algorithm for the calculation of the first two types of problems in the study of text transcription, with respect to the characteristics of text transcription. This paper proposes the concept of text similarity, constructs the TF-IDF similarity evaluation model of text, the text transmission evaluation model based on Gaussian process (i.e., GFCT Model), and the model based on the immune frog jumping algorithm to analyze the comparative processing of text, so as to achieve accurate and effective information processing, with a view to providing a new method for text data processing, and improving the accuracy and effectiveness of text data processing.

Keywords: similarity, vector space, GFCT Model, immune frog jump, adaptive PSO

1. Summary

In the process of copying ancient texts, there are often various errors, so that a book may have been passed down in many versions. In bibliography, errors are often summarized as “black”, “off”, “derivative”, “inverted”, etc., and multiple errors may occur at the same time. There may be multiple errors at the same time. Errors can accumulate in the course of transmission.

Not only did ancient transcribers make mistakes, but even modern communication or storage devices are not immune to random errors after a message has been forwarded or transcribed multiple times. Here, we transform this problem into a more idealized form: assume that the original text is of sufficient length and that, in the process of transmitting it, the transcriber does not cross-check it with other versions. In this way, a large number of different versions may result from the superposition of different errors over a sufficiently long circulation or forwarding process. Comparative text analysis has always been a difficult and complex problem, involving the similarity between texts, different versions of texts, and the number of times a text has been copied. Based on the above background, the research problems of this paper are how to design a reasonable scheme to measure the size of the difference between two different versions of texts; how to establish a mathematical model to estimate the number of copying and transmitting times experienced between two texts if one version is copied from the other after many times of transmitting and transmitting; and how to design an effective and fast algorithm to compute them respectively by combining the first two problems.

2. Literature Review

Now many scholars at home and abroad are studying the calculation method of text comparison analysis. Overseas, Doldsdein et al. carried out similarity calculation by the method of Maximal Marginal Relevance. Chris H. Q. Ding (1999) adopted the method of Latent Semantic Indexing for similarity analysis. In China, Xue Huifang (2011) and Li Jiayuan (2014) studied Chinese sentence similarity calculation, the former from the Chinese language, the latter from the Chinese sentence, for different starting points, and put forward improved algorithms; Liu Qingquan (2020) studied the application of improved TFIDF-based algorithm in text analysis. Sujian Li (2002) proposes a quantitative computation model of utterance relevance based on Knowledge Network and synonym word forest; Yu Chen and Li-Wei Xu (2015) propose a Gaussian mixture model-based text classification algorithm for forestry information; Bing Qin et al. (2003) use the TFIDF method and a semantic-based approach to compute similarity between interrogative sentences for frequently asked question sets; Huan Cui et al. (2004) synthesize the order of keywords, distance between keywords, and the similarity between interrogative sentences and answers in the web-based question-and-answer system.

By reading the above articles, we compare the focuses of different studies, as Table 1 shown in the following table.

Table 1. Comparison with other research priorities

	Similarity	TFIDF	GFCT Model	algorithm optimization
Doldsdein et al.	√			
Chris H. Q. Ding. et al.	√			
Xue Huifang, Li Jiayuan	√			√
Liu Qingquan (2020)		√		
Sujian Li	√			√
Chen Yu, Xu Liwei, etc.		√	√	
Bing Qin	√	√		
This study	√	√	√	√

3. Model Assumption

3.1 Model Assumption

(1) Assume that the two texts being compared are being compared in the same setting, excluding the influence of the remaining factors;

(2) Assuming that all the contents of the two documents can be turned into characters, and can be encoded;

(3) Assuming that all the text is known and that the code spacing of its characters can be calculated.

3.2 Symbolic Analysis

Table 2. Explanation of the meaning of the symbols

Notation	Significance
σ	Scaling factors controlling the degree of variability, with adaptation after imputation — culturalization
	Degree values are inversely proportional
$Sim\ (T,T')$	Model for measuring differences between two different versions of the text
M	Subpopulation size, i.e., number of SFLA communities
B	Parameters controlling the inverse proportional decay function
N_C	Total clone size of subpopulations
ξ	The mean of the approximate solution
Ω	The variance of the approximate solution

D	retrieved document
D	Number of citations
$q(y_\tau=1\|D,\theta,x_\tau)$	probability of error
$Int()$	rounding function
v_i	passages
Q	user search
M	Total number of texts
| |	Euclidean distance
$GerNum$	total number of evolutionary generations
$Iter$	Current number of iterations
$c_{\max}$	Maximum value of the learning factor
w_i	Number of sentences formed by the ith text
$k_\tau=\left[k(x_1,x_\tau),\cdots,k(x_m,x_\tau)\right]^T$	A priori covariance vector between x and the training input X
$Q(a)$	While iterating each word as much as possible, minimize the total of all the Text Calculation Time

4. Model Building

4.1 A Study of the Differences Between the Different Versions of the Text

In order to study the differences between different versions of texts, it is necessary to measure the size of the differences between two different versions of the text, firstly, we proposed the concept of text similarity, applying the similarity to compare the size of the differences between the texts. Secondly, we construct a vector space based TF-IDF similarity evaluation model to quantify the similarity. The final output measures the size of the difference between two different versions of the algorithm model.

4.1.1 Similarity

As we can all know, similarity is a rather complex concept that is widely discussed in semantics, philosophy and information theory. There is no standard and universal definition of similarity because it involves language composition, utterance meaning and some other factors. As words constitute the basic unit of the Chinese language system, calculating their similarity is often also the basis for calculating sentence similarity, while sentence similarity calculation becomes the basis for text similarity calculation. Word similarity is more subjective, and the relationship between words is so complex that it cannot be measured by clear objective standards, which makes it difficult to measure the difference with pure data. In the text, its content is composed of many sentence elements, then we analyze the sentences and define the sentence similarity as:

Sentence similarity refers to the degree of semantic match between two sentences to be compared, the value is a real number between [0, 1], the larger the value indicates that the two sentences are more similar. When the value is 1, it indicates that the two sentences are semantically identical; the smaller the value is, the lower the similarity of the two sentences are, and when the value is 0, it indicates that the two sentences are semantically different.

Then text similarity refers to the degree of word and semantic match between two sentences in this paper, which is a real number between 0 and 1. Theoretically, the larger the value is, the more similar the contents of the two papers are. In the process of text similarity measurement, word similarity is the most basic measure. Word similarity can be discussed by converting it to word distance, which reflects the same relational features through a simple correspondence at the same time. We assume that the two words are W_1 and W_2 respectively, then their similarity is written as $\mathrm{Sim}(W_1,W_2)$, assuming that their word distance is $\mathrm{Dis}(W_1,W_2)$, then we can derive the similarity model of the two texts as shown in Equation 4-1 below. (Zhou Fang, 2005).

$$Sim(W_1,W_2)=\frac{\alpha}{Dis(W_1,W_2)+\alpha} \tag{4-1}$$

In Eq. α we take it as an adjustable parameter. We mean α as the distance value between words when the word similarity is 0.5.

4.1.2 TF-IDF Similarity Evaluation Model Based on Vector Space

Vector space model is an information retrieval model that has been widely used in similarity computation in recent years with outstanding results. Then we define that a text is composed of mutually independent word groups $(T_1,T_2,\dots,T_n)$, and assign a weight ϖ_i to each lexical item T, where ϖ_i denotes the importance of the lexical item in the text wood, and at the same time $(T_1,T_2,\dots,T_n)$ are taken as axes in an n-dimensional coordinate system, $(T_1,\varpi_1,T_2,\varpi_2,\dots,T_i,\varpi_i)$ are the corresponding coordinate values. In this way $(T_1,T_2,\dots,T_n)$ decomposed by the orthogonal word vector group and the corresponding weights form a vector space that can reflect the textual information, where each point is the embodiment of the corresponding text mapped into the space. We use the vectors $(T_1,\varpi_1,T_2,\varpi_2,\dots,T_i,\varpi_i)$ to represent all the text and user query information, where the user query is Q, the retrieved document D. Then at this point we will be able to use the angle between the vectors to measure the degree of similarity of information matching, the larger the angle the lower the similarity. That is to say, the related problem of text matching is transformed into the vector matching problem in vector space. The TF-IDF similarity evaluation model based on vector space integrates the frequency of occurrence of a word in all texts (TF value) and the discriminative ability of this word to different texts (IDF value) (Chunhui You, 2008).

Assuming that $(\varpi_1,\varpi_2,\dots,\varpi_i)$ are all words occurring in the text, then the n-dimensional vector is denoted $(T_1,T_2,\dots,T_n)$for each sentence in the text. Let n be the number of times a particular word W appears in the target text, m be the number of all other texts that appear or contain ϖ. M be the total number of texts, $T_i(l\leq i\leq n)$, then $T=n\times log(M/m)$. In this formula, the more times the word occurs, the larger the n value will be, but this does not mean that the word must have a high T value. For example, the word "的" appears very frequently in all Chinese language usage, so this word will have a large TF value (n value), but because it is not very useful for understanding and $(T_1^{'},T_2^{'},\dots,T_n^{'})$ distinguishing between different texts, instead, it has a small IDF value ($log(M/m)$) very small. Using this method to calculate the similarity, the frequency of word usage and word discrimination are taken into account. By the same token, the above method can be utilized for the calculation of n-dimensional to in our target text. After obtaining T and T' respectively, we calculate the cosine of the angle between the two vectors, and then obtain the similarity between the two texts. (Zhou Fang, 2005) The similarity between the two texts can be obtained.

Assuming that the two texts are $(T_1,T_2,\dots,T_n)$ and $(T_1^{'},T_2^{'},\dots,T_n^{'})$, then it can be concluded that they are similar as shown in the following equation.

$$Sin(T,T) = \begin{cases} \sum_{i=1}^{n} T_i * T_i^{\prime} \\ \dfrac{2\sum_{i=1}^{n} T_i * T_i^{\prime}}{\sum_{i=1}^{n} T_i^{2} + \sum_{i=1}^{n} T_i^{\prime 2}} \\ \dfrac{\sum_{i=1}^{n} T_i * T_i^{\prime}}{\sum_{i=1}^{n} T_i^{2} + \sum_{i=1}^{n} T_i^{\prime 2} - \sum_{i=1}^{n} T_i * T_i^{\prime}} \\ \dfrac{\sum_{i=1}^{n} T_i * T_i^{\prime}}{\sqrt{\sum_{i=1}^{n} T_i^{2}} * \sqrt{\sum_{i=1}^{n} T_i^{\prime 2}}} \end{cases} \tag{4-2}$$

The four formulas in Eq. (Wang Li-Bureau, 2008).

In summary, we can derive a model for measuring the difference between two different versions of the text *Sim*(*T*,*T*').

4.2 Study on the Number of Copies Between Texts

For the problem of the number of transmissions between texts, since each transmission super may not have the same error probability, in order to better estimate the number of transmissions that exist between two texts, we firstly take the acquaintance model constructed in Problem 1 as an object of study, and construct the text transmission evaluation model based on Gaussian process (i.e., GFCT Model). Secondly, assuming the error probability first condition, the similarity between two texts is calculated iteratively according to GFCT Model. Finally, the model outputs the number of transmissions between two texts by probabilistic transformation. The solution of the problem is shown in Figure 1 shows.

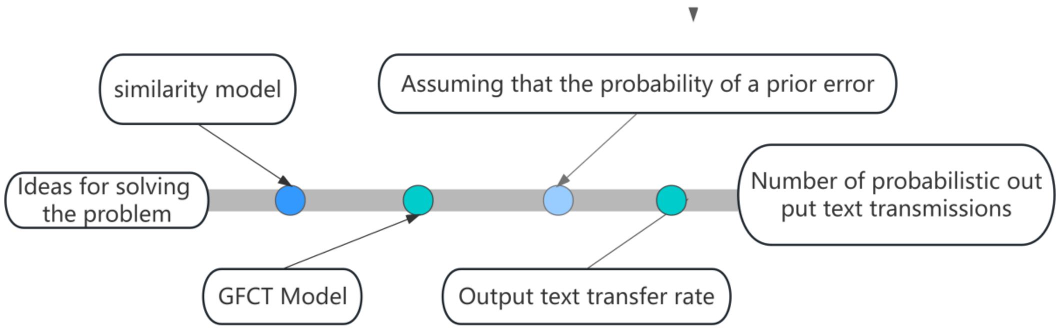

similarity model
Assuming that the probability of a prior error
Ideas for solving the problem
Number of probabilistic out put text transmissions
GFCT Model
Output text transfer rate

Figure 1. Problem Solution Idea

4.2.1 GFCT Model Algorithm

Gaussian process is a deep machine learning method proposed based on Bayesian theory, in which the distribution of any integer n ⩾ 1 finite random variables are Gaussian distributions of corresponding dimensions. By learning the existing sample set, summarizing the intrinsic complex correlation law, and building a Gaussian process model, the Gaussian process can predict the output state if there is a new sample input (He Jianjun, Zhang Junxing, Jia Siqi, et al., 2014).

Assuming that the vector x is the input value of the learning sample influencing factors and the corresponding binary classification flag bit y, $y \in \{0,1\}$, the set of learning samples is denoted as $D = \{(x_i, y_i)|\ i = 1, \ldots, m\}$, where m is the total number of learning samples. When x is determined,

$\vartheta(v|\ x)$ obeys the Bernoulli distribution; when y = 1, its associated probability can be derived as $\vartheta(y=1\,\big|\,x)=\Psi(f(x))$, where $\Psi()$ denotes the cumulative probability density function of the standard Gaussian distribution, and the function $\Psi(z)=\dfrac{1}{(1+e^{-z})}$ is chosen, where f (x) is the latent function (Qin Bing, Liu Ting, Wang Yang, Zheng Shifu & Li Sheng, 2003; Cui Huan, Cai Dongfeng & Miao Xuelai, 2004). In order to simplify the symbols, we simplify the function symbols as shown in the following table (Zhou Fang, 2005):

Potential Functions (Qin Bing, Liu Ting, Wang Yang, Zheng Shifu & Li Sheng, 2003; Cui Huan, Cai Dongfeng & Miao Xuelai, 2004) To keep the notation simple, we simplify the function notation as in Table 3 shown in (Chunhui You, 2008):

Table 3. Simplified table of symbols

Original representation	Improved presentation
$f(xi)$	fi
$[f_1,\cdots,f_m]^T$	f
$[y_1,\cdots,y_m]^T$	y
$[x_1,\cdots,x_m]^T$	X

After determining the latent function, the likelihood function of the learning sample is set as:

$$\vartheta(y|\ f)=\prod\nolimits_{i=1}^{m}\vartheta(yi|\ fi)=\prod\nolimits_{i=1}^{m}\Psi(yi|\ fi) \tag{4-3}$$

Then the prior distribution of its potential function is:

$$\vartheta(f|\ X,\theta)=N(0,K) \tag{4-4}$$

where $\theta=\{\sigma_f,l\}$ is the hyperparameter, which in turn is the covariance matrix with K of order m*m determined by optimization using the great likelihood of the latent function f , and $K_{ij}=k(x_i,x_j,\theta)$, where k() is the covariance function that is positively fixed with θ (ZHANG Binghui, ZHANG Yan, WANG Wei, et al., 2020).

For the covariance to be useful in facilitating the prediction of the GPC, it must be satisfied that, in the matrix, for any set of points, it is possible to randomly generate a non-negative positive definite writing covariance matrix. Suppose our squared exponential covariance function is:

$$k(x_p,x_q)=\sigma_f^2\sqrt{\left(-\frac{1}{2l^2}(x_p-x_q)^2\right)} \tag{4-5}$$

The sample collection will be collected, and when the sample collection has a certain size, the posterior distribution of the latent function f can be obtained from Bayes' principle as:

$$\vartheta(f\,\big|\,D,\theta)=\frac{\vartheta(y\,\big|\,f)\vartheta(f\,\big|\,X,\theta)}{\vartheta(D\,\big|\,\theta)}=\frac{N(0,K)}{\vartheta(D\,\big|\,\theta)}\prod_{i=1}^{m}\Psi(y_i,f_i) \tag{4-6}$$

From (4-3) to (4-6), it can be obtained that GPC has extracted the properties of the sample and the sample learning is finished. When the new prediction sample carries out the input, the x_τ conditional probability of the corresponding potential function f_τ of the prediction sample is expressed as (He Jianjun, Zhang Junxing, Jia Siqi, et al., 2014):

$$\vartheta(f_\tau | D,\theta,x_\tau)=\int\vartheta(f | f,X,\theta,x_\tau)\vartheta(f | D,\theta)df \tag{4-7}$$

Its corresponding predicted probability for y_τ is:

$$\vartheta(y_\tau | D,\theta,x_\tau)=\int\vartheta(y_\tau | f_\tau)\vartheta(f_\tau | D,\theta,x_\tau)df \tag{4-8}$$

Observing the experience of GPC and the principle of probability maximization test, we set $\sqrt{\vartheta(y_\tau | D,\theta,x_\tau)Sim(T,T^{'})}=0.5$ as the overall classification threshold, which belongs to "0" when the probability of error is greater than 0.5, and "1" when the probability of error is less than or equal to 0.5.

From the above can be the prediction probability of the equation (4-7) (4-8), the equation is more abstract, so we need to simplify its solution. Assuming that ξ and $\mathbf{\Omega}$ are the mean and variance of the approximate solution, respectively, the posterior distribution of the potential function f of the learning sample is transformed into a myopic Gaussian distribution as (He Jianjun, Zhang Junxing, Jia Siqi, et al., 2014; ZHANG Binghui, ZHANG Yan, WANG Wei, et al., 2020):

$$\vartheta(f | D,\theta)\approx q(f | D,\theta)=N(\xi,\Omega) \tag{4-9}$$

From this, we can derive the posterior distribution of the potential function f_τ for the predicted sample is represented as follows:

$$q(f_\tau | D,\theta,x_\tau)=N(\mu_\tau,\sigma_\tau^2)\mathrm{Sim}(T,T^{'}) \tag{4-10}$$

The mean and variance in equation (4-10) are shown below, respectively:

$$\begin{cases}\mu_\tau = k_\tau^T K^{-1}\boldsymbol{m}\\ \sigma_\tau^2 = k(x_\tau,\mathrm{x}_\tau)-k_\tau^T(\mathrm{K}^{-1}-K^{-1}AK^{-1})k\end{cases} \tag{4-11}$$

where $k_\tau=[k(x_1,x_\tau),\cdots,k(x_m,x_\tau)]^T$ denotes the prior covariance vector between x_τ and the training input

X. The predicted probability that belongs to the flag "1" is as follows (Chen Y & Xu Li-wei, 2015; Qin Bing, Liu Ting, Wang Yang, Zheng Shifu & Li Sheng, 2003). The predicted probability of x_τ belonging to flag "1" is assumed as follows (He Jianjun, Zhang Junxing, Jia Siqi, et al., 2014; ZHANG Binghui, ZHANG Yan, WANG Wei, et al., 2020):

$$q(\gamma_\tau=1 | D,\theta,x_\tau)=\frac{\theta\sum\left(Sim(T,T')*\Psi\left(\frac{\mu_\tau}{\sqrt{(1+\sigma_\tau^2)}}\right)\right)}{\frac{\mu_\tau}{\sqrt{(1+\sigma_\tau^2)}}} \tag{4-12}$$

The final result is the $q(y_\tau=1 | D,\theta,x_\tau)$ error probability, where D is the number of transmissions.

4.2.2 The Process of Modeling

This model is a probabilistic model, so we have to analyze and calculate, we must know the specific content of the two versions of the text in advance, and then apply the similarity model proposed in question 1 to analyze and calculate the text, output the distance of the text editing at this time, and then carry out the calculation of the GFCT Model.

In order to better calculate the number of transmissions between two texts and to facilitate the model's operation, we plan the calculation process of GFCT Model, which is calculated as follows (Chen C, Seff A, Kornhauser A, et al., 2015).

Step 1: Data Substitution

Substituting the two texts as textual data defines the primary probability as θ.

Step 2: Data Training

All training samples are learned and the optimal hyperparameters of the covariance function are obtained by maximizing the log-likelihood function of the learned samples;

Step 3: Sample data posterior probability

According to Bayes' rule, the training samples are subjected to "inductive inference learning", and the posterior approximate Gaussian distribution of the predicted sample latent function f_τ is obtained according to Eq. (4-10).

Step 4: Assignment of probabilities

Observing the experience of GPC and the principle of probability maximization test, we set $\sqrt{\vartheta(y_\tau | D, \theta, x_\tau) Sim(T, T^{'})} = 0.5$ as the overall classification threshold, which belongs to "0" when the probability of error is greater than 0.5, and "1" when the probability of error is less than or equal to 0.5.

Step 5: Output the dataset

Repeating the run five times yields the precise text transmission probability distributions for each of the two species, as well as the mean value of the number of transmissions.

4.2.3 Concluding Analysis

For the problem of the number of transmissions between texts, since each transmission super may not have the same probability of error, in order to better estimate the number of transmissions that exist between two texts, we must firstly know the specific content of the two versions of the text in advance, and then apply the similarity model proposed in Problem 1 to analyze and compute the text, and output the distance of the text editing at this time. Secondly, the similarity model constructed in Problem 1 is used as a research object to construct a text transmission evaluation model based on Gaussian process (i.e., GFCT Model). Next the similarity between two texts is calculated iteratively according to GFCT Model assuming the error probability first condition. Finally, the model outputs the number of transmissions between two texts through probabilistic transformation. The model outputs the number of copies passed between two texts through probabilistic transformation (Chunhui You, 2008).

4.3 Model Optimization

The model constructed for model optimization needs to provide a fast calculation scheme for the problem of differences between different versions of texts and the number of transmissions between texts, respectively. Firstly, in order to better calculate the similarity of the two texts, we apply the immune frog jump algorithm model to iteratively calculate the similarity of the entire text, the immune frog jump algorithm has a strong iterative optimization ability, which can speed up the calculation speed of the vector space-based TF-IDF similarity evaluation model. Secondly, due to the number of transmissions between texts the optimization ability of GFCT Model is weak, in order to improve efficiency, we apply the adaptive PSO algorithm its optimization to enhance the optimization ability and computing rate. Finally, we apply the algorithms to calculate the two models and output the optimal running time and evaluation results respectively.

4.3.1 Immune Frog Jump Optimized VSM Model

Although the hybrid frog jump algorithm was proposed relatively late, due to its unique grouping and reorganization mechanism, it has good parallelism, positive feedback, and self-organization, and many scholars have carried out targeted improvement research on this algorithm, such as the chaotic frog jump algorithm, the improved hybrid frog jump algorithm based on the threshold selection strategy, and the two-layer interaction hybrid differential evolution algorithm, etc. The purpose of these algorithmic improvements is mainly to improve the algorithm's ability of local optimization, overcome the algorithm's vulnerability to premature maturation, and improve the convergence speed of the algorithm. The purpose of these algorithms is to improve the local optimization ability of the algorithm, overcome the defects of the algorithm which is easy to fall into premature maturity, and improve the convergence speed of the algorithm, etc. The improvement method mainly focuses on the parameter adjustment method, and the fusion of multiple intelligent algorithms, etc. Compared with the traditional SFLA, these improvement methods to a certain extent can obtain better optimization ability, but they can not deal with the deficiencies of easy to fall into local optimal solution and slow convergence speed in later stage, which to a certain extent reduce the convergence efficiency of the algorithm. To some extent, the convergence efficiency of the algorithm is reduced, and the number of iterations as well as the computation time will be increased in order to improve the solution accuracy of the algorithm. The immuno-evolutionary algorithm converges to the global optimal solution with probability l, simple parameter setting, less dependent on human experience, and has strong local search capability. Combining the respective advantages of hybrid frog jumping algorithm and immuno-evolutionary algorithm, this chapter proposes the immuno-frog jumping algorithm, which makes full use of the global search ability of SFLA and the local search ability of the immuno-evolutionary algorithm, takes into account the balance between the global search and the local search ability of the evolution strategy, and adaptively adjusts the step factor to improve the algorithm's optimization efficiency in the evolution process, and ultimately obtains the near-optimal immuno-evolutionary algorithm of the optimization problem. The feasible solution with the largest fitness value in each generation is regarded as the optimal individual (antibody), and makes full use of the information of the optimal individual, and uses the optimal individual instead of the group to carry out the evolutionary search for the optimal; the cloning selection algorithm is an immune evolutionary algorithm that draws on the characteristics of the biological system, and takes the high-frequency variation as the main search mode, and there are mainly three kinds of operation operators, namely, cloning, variation and selection; moreover, the CSA constructs the memory antibody to maintain the diversity of the group to mimic the immune mechanism, so that it can be used to search for the best possible solution for the problem. In addition, CSA also constructs memory antibodies to maintain the diversity of the population to mimic the immune mechanism, so it has the advantages of good population diversity, not easy to converge prematurely and strong local search ability. At the late stage of SFLA evolutionary process, the population usually contains more similar individuals with better adaptation values and similar structures (convergence of the worst individual), which causes the worst individual to gradually approach these better solutions, which results in the rapid decline of population diversity, which is also the key to intelligent algorithms that are prone to fall into local optimization. To address this shortcoming, this paper considers the set of optimal solutions of each community after each mixing of SFLA as a sub-population, and performs the immune clone selection process to locally find the optimal. The main algorithm parameters are as follows (ZHANG Binghui, ZHANG Yan, WANG Wei, et al., 2020):

(1) Cloning scale

Considering all the words of the two texts as populations, then the size of the sub-population individual clones is proportional to the fitness value, which constitutes the clone temporary population $\left(C_g\right)$, which is calculated according to Equation (4-13):

$$N_i = Int(N_c * \frac{f(X_i)}{\sum_{i=1}^{M} f(X_i)}) \tag{4-13}$$

Where: N_i is the clone size of individual X_i, N_c is the total clone size of sub-population; generation $f(X_i)$ is the normalized fitness value of X_i; $Int()$ is the rounding function; M is the sub-population size, i.e., the number of SFLA communities. From Equation (4-13), it can be seen that the clone size of sub-population M is proportional to the affinity (fitness value) of individuals, and the larger the fitness value is, the larger the number of cloned individuals is, making the algorithm to a large extent to make the good genes of the individuals with high fitness values better preserved and replicated;

(2) Antibody variation

High-frequency mutations were applied to the clonal temporary population C_g to form the clonal mutation set $(C_g *)$, as calculated in Equation (4-14):

$$\begin{cases} x_j' = x_j(1+\sigma N(0,1)) \\ \sigma = \dfrac{1}{\beta}\exp(-f(X_i)) \end{cases} \tag{4-14}$$

Where: x_j、x_j' are the values before and after the variation of the jth dimensional component of the individual X_i respectively, N(0,1) is a random variable obeying normal distribution, σ is the scale factor controlling the degree of variation, which is inversely proportional to the adaptation value after normalization, and β is the parameter controlling the inverse proportional decay function. From equation (4-14), it can be seen that the variation operator is inversely proportional to the individual fitness value, and the higher the fitness value, the smaller the degree of variation, which to a certain extent makes the optimal individual better preserved, while the poorer individual gets greater improvement, which is the core of the immune clonal selection algorithm. This is also the core of the immunoclonal selection algorithm (Chen C, Seff A, Kornhauser A, et al., 2015).

(3) Antibody Selection

The antibody-to-antigen affinity (fitness value) in the antibody variant set $(C_g *)$ is evaluated against the similarity of the two texts, and the top M individuals with the highest affinity are selected to form a new population C_b, which is selected according to Equation (4-15) for $\forall i = 1, 2, M,$ specifically:

$$X_i(i = 1, 2, ...F) \tag{4-15}$$

Where: N_i is the clone size of individual X_i, $C_g^*(X_i)$ is the set of individuals in the set of mutated individuals obtained by cloning X_i.

(4) Determine the objective function

In order to quickly get the optimal value of two text comparison iterations, let n texts, M words, the number of i-th text composing a sentence is w_i, paragraph is v_i, M is the total number of texts, the user query is Q, and the retrieved document D, d_{ij} is the computed value of $\mathrm{Sim}(T, T')$ each time. In summary, the optimization-based mathematical model we derive for measuring the minimum time to compute the difference between two different versions of a text is (Zhang Jun, 2011):

$$\min \sum_{k=1}^{M} \sum_{i=0}^{N-1} \sum_{j=1}^{N-1} (x_{ij}^k d_{ij} + x_{j0}^k d_{jo}) \sigma \tag{4-16}$$

$$x_{ij}^0 = \begin{cases} 1, \text{Execute the kth match after the jth match is completed} \\ 0, or\ else \end{cases} \tag{4-17}$$

$$x_{ij}^1 = \begin{cases} 1, \text{The kth text comparison goes from node i to node j} \\ 0, or\ else \end{cases} \tag{4-18}$$

$$V_{ik}^3 = \begin{cases} 1, \text{The text of node i is computed from k similarities} \\ 0, or\ else \end{cases} \tag{4-19}$$

4.3.2 Updating of the Worst Individual in the Community

When the worst individual in the SFLA community is updated, it only utilizes the optimal frog in the community and the global optimal solution at one time (intuitively represented as a frog jumping along the line between the worst and optimal individuals), which may fall into the local optimum and cause the algorithm to “mature prematurely”. In order to compensate for the shortcomings of the original method, drawing on the particle swarm algorithm particle flight update strategy, the learning factor C is introduced, which changes adaptively with the number of iterations during the iterative process of the algorithm. At the beginning of the algorithm, the learning factor is small, and the frog individuals in the worse position are easy to be close to the better individuals; with the depth of the iterative process, the learning factor gradually becomes larger (it can not be too large, otherwise it affects the algorithm convergence), which is conducive to the algorithm to jump out of the local optimum. In this paper, the learning factor is changed in a linear way, and the improved formula for updating the worst frog individual is as follows (4-20) (Zhang Jun, 2011).

$$\begin{cases} D_i = rand() \times (X_b^k - X_w^{k,old}) * C \\ C = C_{\min} + iter/GerNum * (C_{\max} - C_{\min}) \end{cases} \tag{4-20}$$

Where: *iter* is the current number of iterations, *GerNum* is the total number of evolutionary generations, $C_{\max}$ and $C_{\min}$ are the maximum and minimum values of the learning factor, respectively.

4.3.3 Immune Frog Jump Optimized VSM Model

Based on the above description of the optimization of the similarity model, the entire frog population search space (feasible domain range) is first defined as $\boldsymbol{\Omega}$, and the number of individuals in the frog population is F = 20 (the number of segments). Any frog individual is available at $X_i(i=1,2,\ldots F)$, representing a candidate solution to the problem. Where $X_i = (x_{i1}, x_{i2}, \cdots, x_{is})$, s represents the number of dimensions of the problem variables to be optimized. The flowchart of the model construction is shown in Figure 2 (ZHANG Binghui, ZHANG Yan, WANG Wei, et al., 2020).

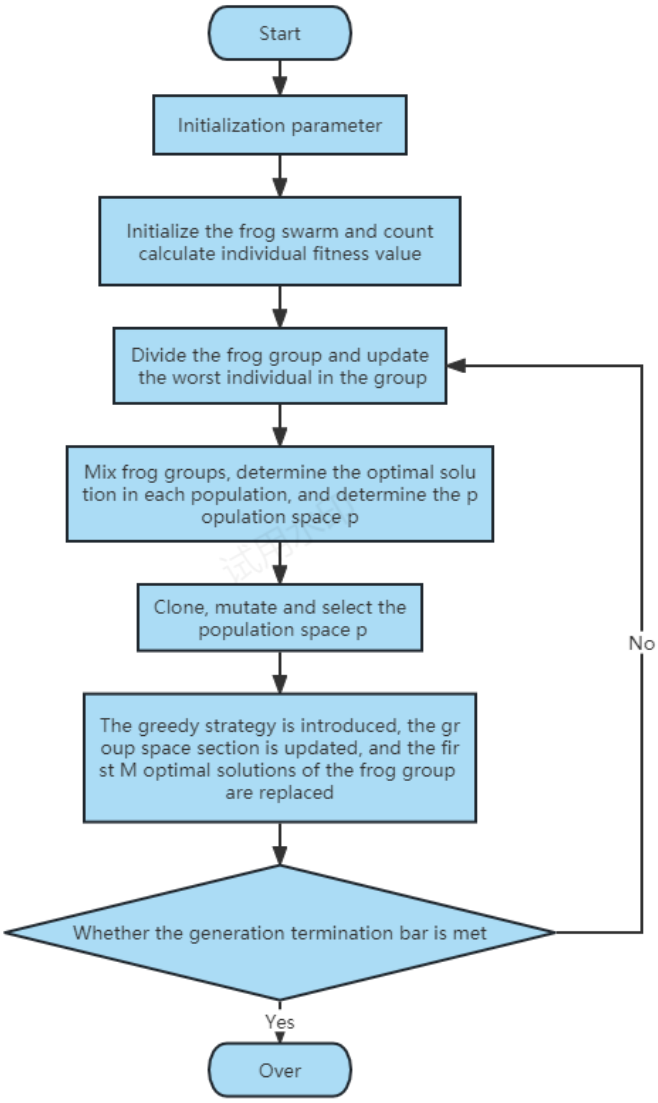

Start
Initialization parameter
Initialize the frog swarm and count calculate individual fitness value
Divide the frog group and update the worst individual in the group
Mix frog groups, determine the optimal solu tion in each population, and determine the p opulation space p
Clone, mutate and select the population space p
The greedy strategy is introduced, the gr oup space section is updated, and the fir st M optimal solutions of the frog group are replaced
Whether the generation termination bar is met
No
Yes
Over

Figure 2. Flowchart of the VSM model for immune frog jump optimization

Step 1: Initialization of algorithm parameters

The model parameters are set according to the size of the problem to be optimized: the number of frog individuals P, the number of communities M, the number of intra-community iterations Gin, the number of mixed iterations Gshuff the minimum learning factor $C_{\min}$, the maximum learning factor $C_{\max}$, the size of clones C for the clone selection algorithm, and the size of the memory population m.

Step 2: Calculate the degree of adaptation

Randomly initialize F individual frogs that meet the constraints and compute the fitness value for each individual and note i = 0;

Step 3: Forming a Community

The initial populations were ranked in descending order according to the size of the fitness value, and the individuals within the frog population were divided into populations according to the principle of population division to form M populations;

Step 4: Preliminary Iterative Calculation

For each community, determine the optimal individual X_{b}, the worst individual X_w, and the optimal individual X_g of each frog group, and use the improved step update formula to update X_{b}, and update X_{b}, X_w, and X_g at the same time, and repeat this step until the iterative computation of Gin times for M communities is completed (Li Ting, 2018).

Step 5: Arrangement Iteration

Recombine and reorder the entire frog population by fitness values. Remember the first M better solutions in the frog population (i.e., the optimal solutions of each population) as the population space P. These better individuals represent the better individuals searched by the current frog population, and merely grouping and aggregating frog populations does not allow for a more adequate search for these better individuals (to a certain extent, it only ensures the global convergence of the algorithm), whereas the optimal solutions are usually in the vicinity of the better individuals, and a clonal selection operation is carried out on the population, which is usually in the vicinity of the better individuals. Perform further search helps to improve the efficiency of the algorithm to find the optimal. The specific steps are as follows (Step6~Step8) (He Jianjun, Zhang Junxing, Jia Siqi, et al., 2014).

Step 6: Clone Collection

Normalize the individual affinity (fitness value) in the population space P and clone the group size to the set population size, as shown in Equation (6-2) to obtain the set of cloned groups (C_g);

Step 7: Variant Collection

Perform the high frequency mutation operation according to equation (6-3) on, the C_g set to obtain the mutated population set $(C_g *)$.

Step 8: Intervenes in the greedy algorithm

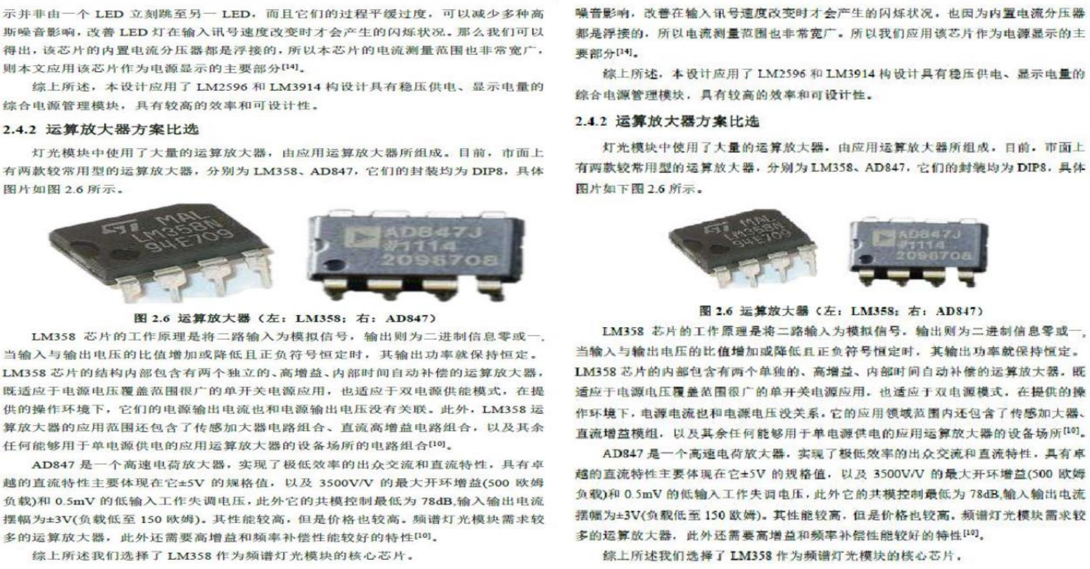
A neighborhood-based greedy strategy is applied to the pre-mutation population p and the post-mutation population $(C_g *)$ to select the better individuals and replace the first M better solutions of the frog population, followed by updating the memory population and simulating the cell extinction process.

Step 9: Judge whether the termination condition is reached

Determine whether the evolutionary process of the algorithm has reached the termination condition. The termination condition can generally be set to the maximum number of iterations or the difference between the optimal fitness values of two neighboring generations to meet a predetermined threshold. If the termination condition is reached, the optimization search ends and the optimal result is output; otherwise, return to Step 3 and iterate again. Finally, the global iteration outputs the similarity between the two texts under the optimized conditions, and outputs the optimal time at this time.

4.3.4 Example Analysis and Calculation Results

Assuming that the two texts have 800 characters each, their specific contents are shown below.



示并非由一个 LED 立刻跳至另一 LED，而且它们的过程平缓过度，可以减少多种高斯噪音影响，改善 LED 灯在输入讯号速度改变时才会产生的闪烁状况。那么我们可以得出，该芯片的内置电流分压器都是浮接的，所以本芯片的电流测量范围也非常宽广，则本文应用该芯片作为电源显示的主要部分[14]。

综上所述，本设计应用了 LM2596 和 LM3914 构设计具有稳压供电、显示电量的综合电源管理模块，具有较高的效率和可设计性。

2.4.2 运算放大器方案比选

灯光模块中使用了大量的运算放大器，由应用运算放大器所组成。目前，市面上有两款较常用型的运算放大器，分别为 LM358、AD847，它们的封装均为 DIP8，具体图片如图 2.6 所示。

图 2.6 运算放大器（左：LM358；右：AD847）

LM358 芯片的工作原理是将二路输入为模拟信号，输出则为二进制信息零或一，当输入与输出电压的比值增加或降低且正负符号恒定时，其输出功率就保持恒定。LM358 芯片的结构内部包含有两个独立的、高增益、内部时间自动补偿的运算放大器，既适应于电源电压覆盖范围很广的单开关电源应用，也适应于双电源供能模式，在提供的操作环境下，它们的电源输出电流也和电源输出电压没有关联。此外，LM358 运算放大器的应用范围还包含了传感加大器电路组合、直流高增益电路组合，以及其余任何能够用于单电源供电的应用运算放大器的设备场所的电路组合[10]。

AD847 是一个高速电荷放大器，实现了极低效率的出众交流和直流特性，具有卓越的直流特性主要体现在它±5V 的规格值，以及 3500V/V 的最大开环增益(500 欧姆负载)和 0.5mV 的低输入工作失调电压，此外它的共模控制最低为 78dB,输入输出电流摆幅为±3V(负载低至 150 欧姆)。其性能较高，但是价格也较高。频谱灯光模块需求较多的运算放大器，此外还需要高增益和频率补偿性能较好的特性[10]。

综上所述我们选择了 LM358 作为频谱灯光模块的核心芯片。

噪音影响，改善在输入讯号速度改变时才会产生的闪烁状况。也因为内置电流分压器都是浮接的，所以电流测量范围也非常宽广。所以我们应用该芯片作为电源显示的主要部分[14]。

综上所述，本设计应用了 LM2596 和 LM3914 构设计具有稳压供电、显示电量的综合电源管理模块，具有较高的效率和可设计性。

2.4.2 运算放大器方案比选

灯光模块中使用了大量的运算放大器，由应用运算放大器所组成，目前，市面上有两款较常用型的运算放大器，分别为 LM358、AD847，它们的封装均为 DIP8，具体图片如下图 2.6 所示。

图 2.6 运算放大器（左：LM358；右：AD847）

LM358 芯片的工作原理是将二路输入为模拟信号，输出则为二进制信息零或一，当输入与输出电压的比值增加或降低且正负符号恒定时，其输出功率就保持恒定。LM358 芯片的内部包含有两个单独的、高增益、内部时间自动补偿的运算放大器，既适应于电源电压覆盖范围很广的单开关电源应用，也适应于双电源模式，在提供的操作环境下，电源电流也和电源电压没关系，它的应用领域范围内还包含了传感加大器、直流增益模组，以及其余任何能够用于单电源供电的应用运算放大器的设备场所[10]。

AD847 是一个高速电荷放大器，实现了极低效率的出众交流和直流特性，具有卓越的直流特性主要体现在它±5V 的规格值，以及 3500V/V 的最大开环增益(500 欧姆负载)和 0.5mV 的低输入工作失调电压，此外它的共模控制最低为 78dB,输入输出电流摆幅为±3V(负载低至 150 欧姆)。其性能较高，但是价格也较高。频谱灯光模块需求较多的运算放大器，此外还需要高增益和频率补偿性能较好的特性[10]。

综上所述我们选择了 LM358 作为频谱灯光模块的核心芯片。

Figure 3. Two different versions of the text

Then the Figure 3 in the two texts as the object of study, will be substituted into the construction of the VSM model based on the immune frog jump optimization for iterative calculations, calculated between the two different versions of the difference between the size of the calculation results of the two texts to the Word model, will be substituted into the application of MATLAB constructed based on the immune frog jump optimization of the VSM model, through the iterative calculations, can be derived from the model of the optimal time output value is shown in the following figure.

Two kinds of text to Word model, will be substituted into the application of MATLAB constructed based on the immune frog jump optimization VSM model, through iterative calculation, can be derived from the model optimal time output value is shown in the figure below.

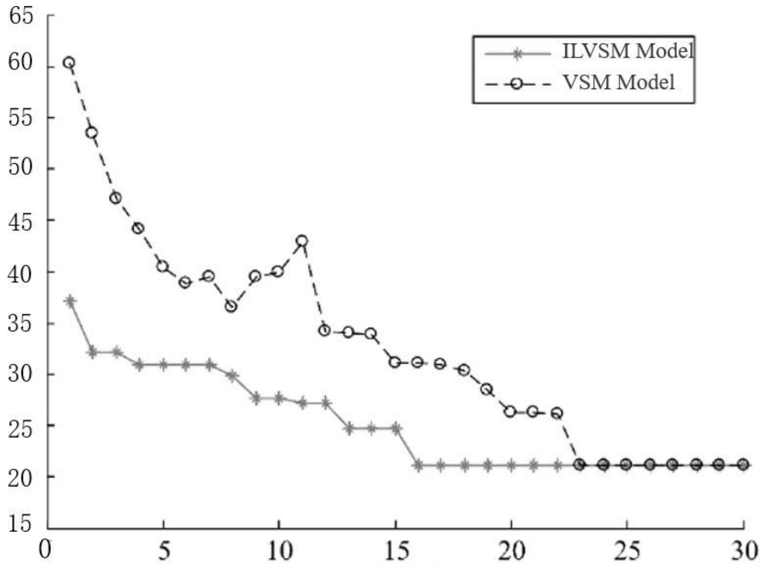


Figure 4. Iteration diagram of the model

According to Figure 4, it can be concluded that when the model is iterated to the 15th generation, the minimum time for the model to run is 21ms, and the similarity of the two texts at this point in the output is 93.79%, then we can conclude that the difference value between the two different versions at this point in time is 6.21%.

4.3.5 Concluding Analysis

In this paper, the algorithm based on immune frog jumping makes full use of the global search capability of SFLA and the local search capability of immune evolutionary algorithm, takes into account the balance between the global and local search capabilities in the evolutionary strategy, and adaptively adjusts the step factor to improve the algorithm's efficiency in searching for excellence in the process of evolution, and applies the algorithmic model of immune frog jumping for iterative calculation of similarity to the whole text, and is able to accelerate the calculation speed of the vector space-based TF-IDF similarity evaluation model. TF-IDF similarity evaluation model calculation speed. Through the example calculation (800 characters), it is concluded that when the model is iterated to the 15th generation, the minimum running time of the model is 21.6ms, and the output of the similarity of the two texts at this time is 93.79%, then we can conclude that the difference value between the two different versions at this time is 6.21%.

4.4 Adaptive PSO Optimization GFCT Model

4.4.1 Adaptive PSO Algorithm Flow

In a D-dimensional target search space, there are n particles forming a particle swarm, where each particle is a D-dimensional vector whose spatial position is denoted as $x_i = (x_{i1}, x_{i2},..., x_{iD})$, $i = 1, 2,..., n$. The spatial position of the particle is a solution in the objective optimization problem, and substituting it into the fitness function can calculate the fitness value, and the superiority of the particle can be measured according to the size of the fitness value; the flight speed of the ith particle is also a D-dimensional vector, which is denoted as $v_i = (v_{i1}, v_{i2},..., v_{iD})$; the position experienced by the ith particle with the best fitness value is called the individual historical best position, denoted as $p_i = (p_{i1}, p_{i2},..., p_{iD})$, and the best position experienced by the entire swarm of particles is called the global historical best position, denoted as $p_g = (p_{g1}, p_{g2},..., p_{gD})$, and the evolutionary equation of the particle swarm can be described as (ZHANG Binghui, ZHANG Yan, WANG Wei, et al., 2020):

$$\begin{gathered} v_{ij}(t+1) = v_{ij}(t) + c_1 r_1(t)(p_{ij}(t) - x_{ij}(t)) + c_2 r_2(t)(p_{gj}(t) - x_{ij}(t)) \\ x_{ij}(t+1) = x_{ij}(t) + v_{ij}(t) \end{gathered} \tag{4-21}$$

Where: subscript j denotes the jth dimension of the particles, subscript i denotes particle i, t denotes the tth generation, c_1, c_2 are acceleration constants, which usually take values between (0,2), and $r_1 \sim U(0,1), r_2 \sim U(0,1)$ are two mutually independent stochastic functions. From the above particle evolution equation, it can be seen that c_1 regulates the step size of particles flying in the direction of their own best position, and c_2 regulates the step size of particles towards the global best position.

By analyzing some characteristics of the group of elementary particles, we can know that the first part of equation (4-21) is the previous speed of the particles; the second part is the "cognitive" part, which represents the thinking of the particles themselves; and the third part is the "social" part, which represents the social information sharing among the particles. The third part is the "social" part, which represents the social information sharing among the particles. Although the relative importance of the social and cognitive parts of the model has not yet been theoretically determined, some studies have shown that for some problems, the social part of the model appears to be more important than the cognitive part (Chunhui You, 2008; Wang Li-Bureau, 2008).

In the algorithm, the potential solution of each optimization problem can be regarded as a point on the n-dimensional search space, i.e., it is assumed that there are no "particles" with volume and mass, and there is a degree of fitness determined by the objective function and their corresponding positions and velocities, and then the particles are dynamically adapted according to the flight experience of the individual and the group. Since the particle swarm algorithm has the advantages of fewer parameters to be adjusted and decentralized search, it is beneficial to the implementation of the projection tracing method. In this paper, the projection tracing method based on particle swarm algorithm will calculate the unit projection vector of each index as the weight. The steps are as follows (Zhou Fang, 2005):

Step 1: Establish linear projection function

Let the jth indicator of the ith sample be $x_{ij}(i=1,2,...,n, j=1,2,...,m)$; where n is the number of samples and m is the number of indicators. If $(a_1,a_2,...,a_m)$ is the m-dimensional projection vector, the expression for the projected eigenvalue Zi of sample i in one-dimensional linear space is (Chunhui You, 2008; Wang Li-Bureau, 2008; Jin Bo & Shi Yanjun, 2005):

$$Z_i=\sum\nolimits_{j=1}^{m}a_jx_{ij} \tag{4-22}$$

Step 2: Establish the objective function

This question needs to address the shortest time to complete the overall computation, i.e., the shortest time to analyze the text.

So our objective function is mediated by the time to carry out. The objective function equation for the projective tracing method is:

$$Q(a)=S(a)\times d(a) \tag{4-23}$$

$Q(a)$ The meaning is: minimize the total text computation time for all while iterating as much as possible for each word, when $Q(a)$ is smaller, the time value is minimized.

$S(a)$ is the sample standard deviation of z_i, i.e., there:

$$S(a)=\sqrt{\frac{1}{n-1}\sum\nolimits_{i=1}^{n}(z_i-\bar{z})^2} \tag{4-24}$$

Where $\bar{z}$ is the mean value of z_i, the larger $S(a)$ is, the more dispersed the values are. $\mathrm{d}(a)$ is the local density of the projected value z_i, i.e., there:

$$d(a)=\sum\nolimits_{i=1}^{n}\sum\nolimits_{i=1}^{n}(R-r_y)\cdot I(R-r_y) \tag{4-25}$$

r_{ij} is the distance between the projected eigenvalues $r_{ij}=|\ z_i-z_j|$, whose value also indicates the degree of dispersion. r is the density window width parameter, whose value is related to the structure of the sample data, and the analysis of R can be seen that its value range $r_{\max}<R\le 2m$, which generally takes the value of $R=r_{\max}+\frac{m}{2}$, where $r_{\max}=\max(r_{ij})$. $I(R-r_{ij})$ For the unit step function, its value satisfies the equation (ZHANG Binghui, ZHANG Yan, WANG Wei, et al., 2020):

$$I(R-r_{ij})=\begin{cases}1, & R-r_{ij}>0\\ 0, & R-r_{ij}\le 0\end{cases} \tag{4-26}$$

Step 3: Optimize projection direction

The objective function $\mathrm{Q}(a)$ will change according to the change of the projection vector a. The proper projection vector a can maximize the possibility to express the feature structure of the high-dimensional data, so the corresponding projection vector a can be found by solving the maximum value of the projected objective function, i.e., there:

Minimum objective function: $\min\mathrm{Q}(a)$

The constraints are: $||\ a\ ||=\sum\nolimits_{i=1}^{m}a_i^2=1;$

Step 4: Construct the solution model

The flowchart of the constructed algorithm is shown in Figure 5 shows (Wang Li-Bureau, 2008):

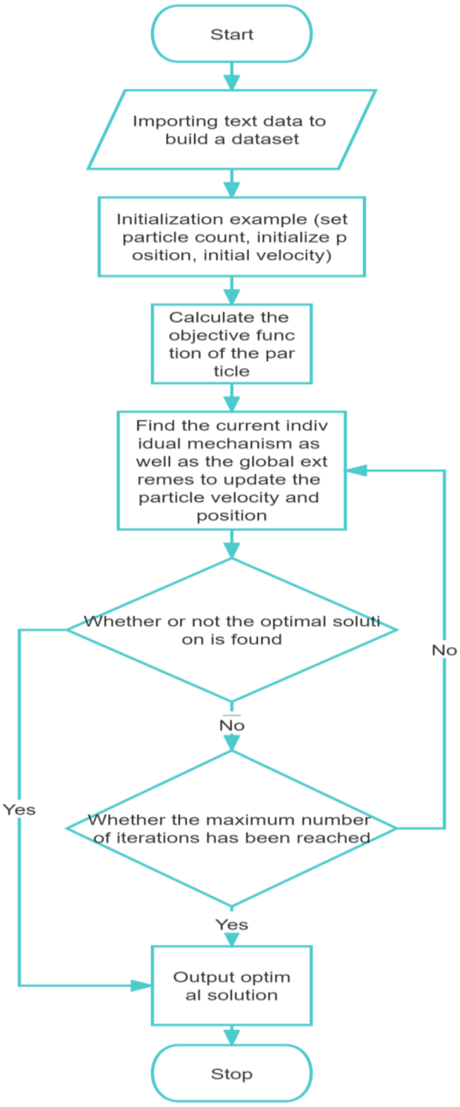

Start
Importing text data to build a dataset
Initialization example (set particle count, initialize p osition, initial velocity)
Calculate the objective func tion of the par ticle
Find the current indiv idual mechanism as well as the global ext remes to update the particle velocity and position
Whether or not the optimal soluti on is found
No
Yes
Whether the maximum number of iterations has been reached
No
Yes
Output optim al solution
Stop

Figure 5. Adaptive PSO algorithm to optimize VSM flowchart

1) Initialize particles (including number of particles, initial velocity, initial position);

2) Calculate the fitness of the particle’s objective function $fit(i)$

3) For each particle, replace it with its fitness value $fit(i)$ and the individual’s extreme value $best\_p(i)$, and if $fit(i) > best\_p(i)$ then $fit(i)$ $best\_p(i)$

4) For each particle, compare its fitness value $fit(i)$ with the global extreme value $best\_g$ and if $fit(i) > best\_g$ replace it with $fit(i)$ $best\_p(i)$

5) Update the velocity v_{ij} and the position x_{ij} of the particle according to the following equation:

$$\begin{gathered} v_{ij}(t+1) = v_{ij}(t) + c_1 r_1(t)(p_{ij}(t) - x_{ij}(t)) + c_2 r_2(t)(p_{gj}(t) - x_{ij}(t)) \\ x_{ij}(t+1) = x_{ij}(t) + v_{ij}(t) \end{gathered} \tag{4-27}$$

If the end condition is met (error is reached or maximum number of cycles is reached) then the loop is exited and the requested single is outputted.

Bitwise projection vector (weight) a and projection eigenvalue zi, otherwise repeat 2 through 5 (Li Ting, 2018).

4.4.2 Adaptive PSO Optimization GFCT Model

In PSO algorithm, the velocity of each dimension of particle i is adjusted by $pbest_i$ (the best position experienced by particle i so far) and $gbest_i$ (the optimal particle position in the whole population); while in LPSO algorithm, the velocity of each dimension of particle i is adjusted by $pbest_i$ and $\overline{p_{neighbor_i}}$ (the optimal particle position in the neighbors of particle i). That is to say, in both PSOs, for the learning of the "self-knowledge part" and the "social part", each dimension of the particle is learned from the corresponding dimension of the same particle (Li Ting, 2018; He Jianjun, Zhang Junxing, Jia Siqi, et al., 2014).

Step 1: Deep self-learning design

On the basis of the global optimization of particle swarm proposed in problem 1, a deep learning strategy is proposed, that is, when updating the particle velocity, the "social part" of the learning samples of each particle is not the particle that is optimal in the neighborhood, but all of its neighbors (including its own history of optimal position), which ensures that the particle is optimal in the current iteration. One-dimensional learning object in the current iteration of the moment are optimal. At a certain iteration moment, the current "social part" learning sample of particle i is shown. The learning objects for each dimension of the particle are determined as follows (Chunhui You, 2008):

$$p_{bin_{(i)}}^{d} = \arg\left\{ \max\left[\frac{fitness(p_j) - Fitness(p_i)}{|\, p_{jd} - x_{id} \,|} \right] \right\}, (i = 1, 2, ..., ps; \ j \in neighbor_i) \qquad \text{(4-28)}$$

$$Sim(T, T') = w \cdot \overline{v_i}(t) + c_1 r_1 (\overline{p_g}(t) - \overline{x_i}(t)) + c_2 r_2 (\overline{p_{bin(i)}}(t) - \overline{x_i}(t)) \qquad \text{(4-29)}$$

Where, $\overline{p_{bin(i)}}$ denotes each one-dimensional learning sample of particle i; $neighbor_i$ denotes the set consisting of neighbors of particle i; $p_{bin_{(i)}}^{d}$ $\arg()$ denotes the identification of the corresponding particle. Equation (4-30) shows that if the Euclidean distance between the dth dimension of particle i and the dth dimension of the jth particle of its neighbor is smaller (meaning closer), the dth dimension of the jth particle will be selected as the dth dimension learning sample of particle i, which makes full use of the optimal information of the particle's neighbors in each dimension (Li Ting, 2018).

Step 2: Self-search process

For the sake of convenience, a population containing 6 particles is used to study the potential search space hypothesis P for both learning strategies; denotes the third particle in the population (its position in the population is divided into three conditions: $P_3 < min(pbest_i)$.

$min(pbest_i) \le P_3 \le max(pbest_i)$和$P_3 \ge max(pbest_i), i = 1, 2...., 6)$, $\quad pbest_4 = gbest, pbest_1 = min(pbest_i)$, $\quad pbest_i = max(pbest_i)$.

L₁ and L₂ denote the search space of the standard PSO algorithm, and L3 and L denote the potential search space of the potential search space of DNMPSO The potential automated search space of the dth dimension of the ith particle in the population of DNMPSO in the standard PSO search process is represented as follows:

$$S_1^d(i) = L_1^d(i) + L_2^d(i) = |\ gbest^d - P_i^d| + |\ pbest_i^d - P_i^d| \qquad \text{(4-30)}$$

Where P_i^d denotes the dth dimension of the ith particle; and | | denotes the Euclidean distance.

Then particle swarm based deep learning and automatic search is based on the model as follows (He Jianjun, Zhang Junxing, Jia Siqi, et al., 2014):

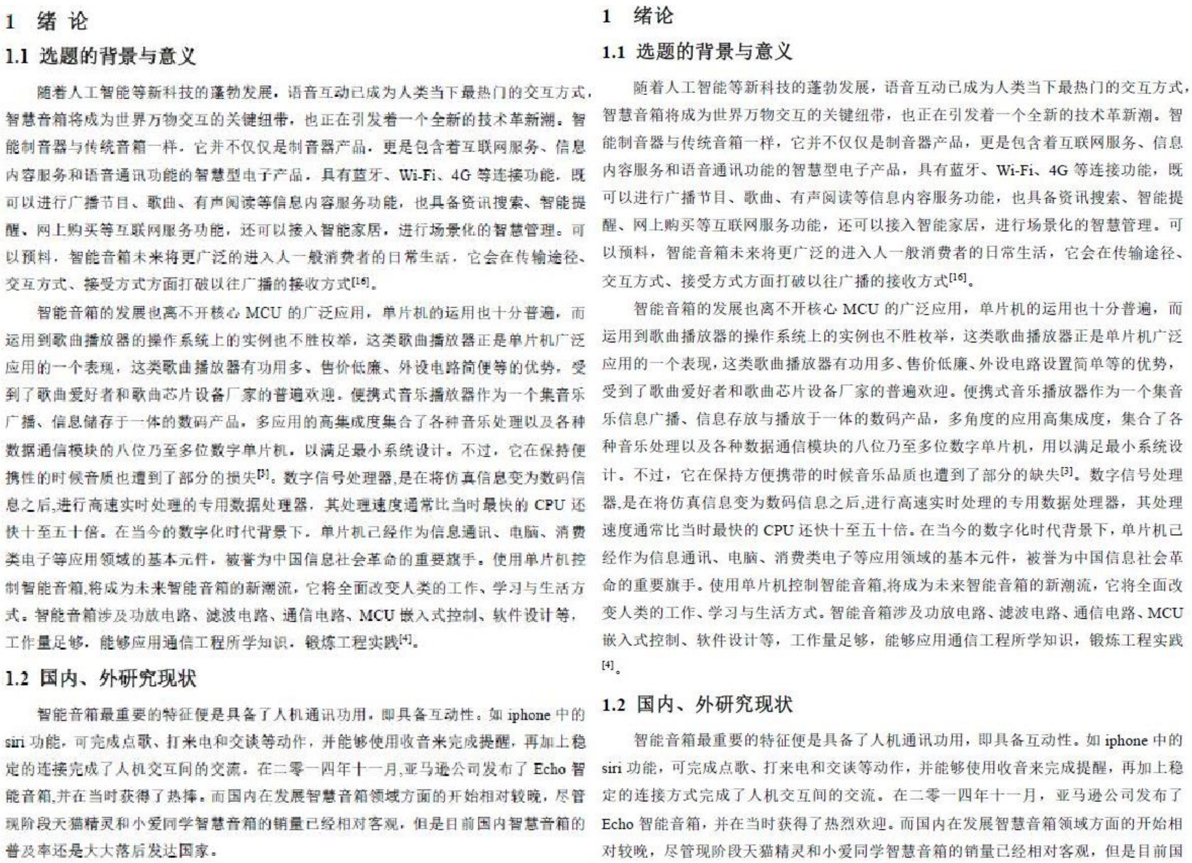
$$q(y_{\text{蝌}}=1|\ D,\theta,x\ )=\begin{cases} L_3'(i)+L_4'(i)=\max(pbext_j')-\min(pbext_j'),if \quad \min(pbext_j')\le P_i'\le\max(pbext_j') \\ \max(L_3'(i),L_4'(i)=\max(pbext_j')-P_i',if \quad P_i'<\min(pbext_j') \\ \max(L_3'(i),L_4'(i))=P_i'-\min(pbext_j'),if \quad P_i'>\max(pbext_j') \end{cases} \tag{4-31}$$

$$\tau_{T'}=\min\left(\frac{\sum_{i=1}^{rs}T'(i)}{ps}-\frac{\sum_{i=1}^{rs}T(i)}{ps}\right)$$

Where, i, j = 1,2,..., ps; ps denotes the population size and T’ denotes the shortest computational time.

5. Simulation Results

In order to validate the results of the adaptive PSO optimization GFCT Model, we analyze two texts of 1000 words from the first edition and the old edition, apply the constructed model to calculate the number of transmissions experienced between the two texts, and output the calculated time. Figure 6 shows the text model of the old and new texts.



1 绪 论

1.1 选题的背景与意义

随着人工智能等新科技的蓬勃发展，语音互动已成为人类当下最热门的交互方式，智慧音箱将成为世界万物交互的关键纽带，也正在引发着一个全新的技术革新潮。智能制音器与传统音箱一样，它并不仅仅是制音器产品，更是包含着互联网服务、信息内容服务和语音通讯功能的智慧型电子产品，具有蓝牙、Wi-Fi、4G 等连接功能，既可以进行广播节目、歌曲、有声阅读等信息内容服务功能，也具备资讯搜索、智能提醒、网上购买等互联网服务功能，还可以接入智能家居，进行场景化的智慧管理。可以预料，智能音箱未来将更广泛的进入人一般消费者的日常生活，它会在传输途径、交互方式、接受方式方面打破以往广播的接收方式[16]。

智能音箱的发展也离不开核心 MCU 的广泛应用，单片机的运用也十分普遍，而运用到歌曲播放器的操作系统上的实例也不胜枚举，这类歌曲播放器正是单片机广泛应用的一个表现，这类歌曲播放器有功用多、售价低廉、外设电路简便等的优势，受到了歌曲爱好者和歌曲芯片设备厂家的普遍欢迎。便携式音乐播放器作为一个集音乐广播、信息储存于一体的数码产品，多应用的高集成度集合了各种音乐处理以及各种数据通信模块的八位乃至多位数字单片机，以满足最小系统设计。不过，它在保持便携性的时候音质也遭到了部分的损失[3]。数字信号处理器,是在将仿真信息变为数码信息之后,进行高速实时处理的专用数据处理器，其处理速度通常比当时最快的 CPU 还快十至五十倍。在当今的数字化时代背景下，单片机已经作为信息通讯、电脑、消费类电子等应用领域的基本元件，被誉为中国信息社会革命的重要旗手。使用单片机控制智能音箱,将成为未来智能音箱的新潮流，它将全面改变人类的工作、学习与生活方式。智能音箱涉及功放电路、滤波电路、通信电路、MCU 嵌入式控制、软件设计等，工作量足够，能够应用通信工程所学知识，锻炼工程实践[4]。

1.2 国内、外研究现状

智能音箱最重要的特征便是具备了人机通讯功用，即具备互动性。如 iphone 中的 siri 功能，可完成点歌、打来电和交谈等动作，并能够使用收音来完成提醒，再加上稳定的连接完成了人机交互间的交流。在二零一四年十一月,亚马逊公司发布了 Echo 智能音箱,并在当时获得了热捧。而国内在发展智慧音箱领域方面的开始相对较晚，尽管现阶段天猫精灵和小爱同学智慧音箱的销量已经相对客观，但是目前国内智慧音箱的普及率还是大大落后发达国家。

1 绪论

1.1 选题的背景与意义

随着人工智能等新科技的蓬勃发展，语音互动已成为人类当下最热门的交互方式，智慧音箱将成为世界万物交互的关键纽带，也正在引发着一个全新的技术革新潮。智能制音器与传统音箱一样，它并不仅仅是制音器产品，更是包含着互联网服务、信息内容服务和语音通讯功能的智慧型电子产品，具有蓝牙、Wi-Fi、4G 等连接功能，既可以进行广播节目、歌曲、有声阅读等信息内容服务功能，也具备资讯搜索、智能提醒、网上购买等互联网服务功能，还可以接入智能家居，进行场景化的智慧管理。可以预料，智能音箱未来将更广泛的进入人一般消费者的日常生活，它会在传输途径、交互方式、接受方式方面打破以往广播的接收方式[16]。

智能音箱的发展也离不开核心 MCU 的广泛应用，单片机的运用也十分普遍，而运用到歌曲播放器的操作系统上的实例也不胜枚举，这类歌曲播放器正是单片机广泛应用的一个表现，这类歌曲播放器有功用多、售价低廉、外设电路设置简单等的优势，受到了歌曲爱好者和歌曲芯片设备厂家的普遍欢迎。便携式音乐播放器作为一个集音乐信息广播、信息存放与播放于一体的数码产品，多角度的应用高集成度，集合了各种音乐处理以及各种数据通信模块的八位乃至多位数字单片机，用以满足最小系统设计。不过，它在保持方便携带的时候音乐品质也遭到了部分的缺失[3]。数字信号处理器,是在将仿真信息变为数码信息之后,进行高速实时处理的专用数据处理器，其处理速度通常比当时最快的 CPU 还快十至五十倍。在当今的数字化时代背景下，单片机已经作为信息通讯、电脑、消费类电子等应用领域的基本元件，被誉为中国信息社会革命的重要旗手。使用单片机控制智能音箱,将成为未来智能音箱的新潮流，它将全面改变人类的工作、学习与生活方式。智能音箱涉及功放电路、滤波电路、通信电路、MCU 嵌入式控制、软件设计等，工作量足够，能够应用通信工程所学知识，锻炼工程实践[4]。

1.2 国内、外研究现状

智能音箱最重要的特征便是具备了人机通讯功用，即具备互动性。如 iphone 中的 siri 功能，可完成点歌、打来电和交谈等动作，并能够使用收音来完成提醒，再加上稳定的连接方式完成了人机交互间的交流。在二零一四年十一月，亚马逊公司发布了 Echo 智能音箱，并在当时获得了热烈欢迎。而国内在发展智慧音箱领域方面的开始相对较晚，尽管现阶段天猫精灵和小爱同学智慧音箱的销量已经相对客观，但是目前国

Figure 6. Calculated text data (left: old text, right: new text)

According to Figure 6, it can be concluded that the number of characters of the text data is 1000, and the two texts are taken as the object of study, and MATLAB is applied to construct the model and import the text for computation, then the results of the model are shown in the following figure.

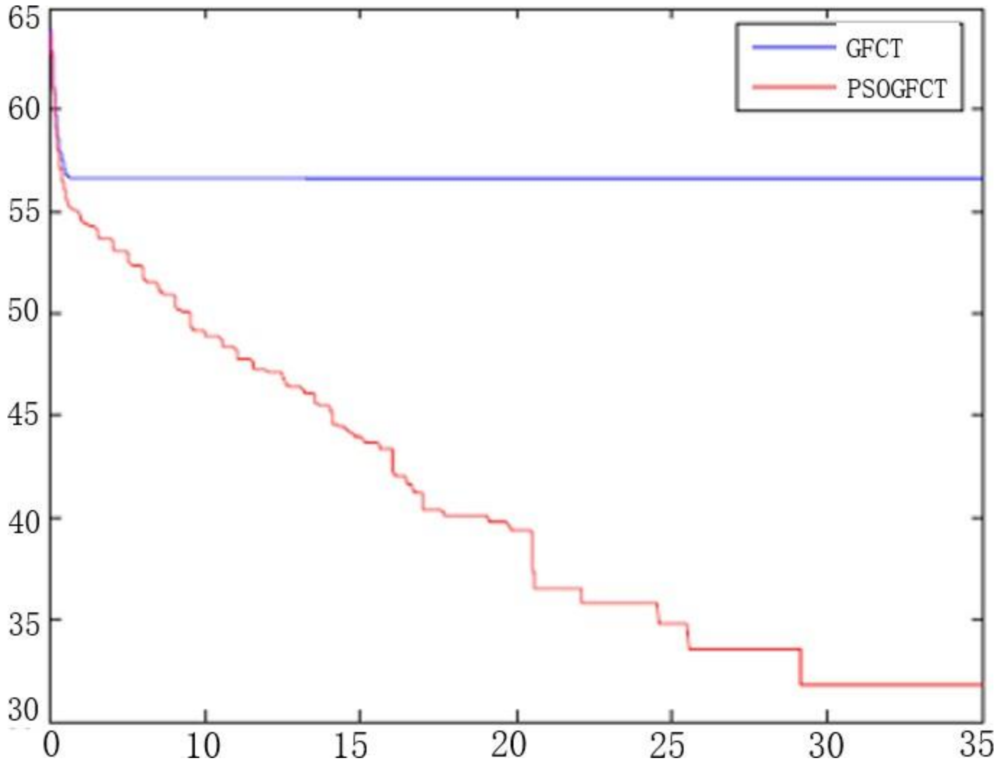


Figure 7. Results of model calculations

According to Figure 7, it can be concluded that there is a large difference between the calculation results before and after the algorithm optimization, the running time before optimization is longer, reaching 58ms, when the model iterates to 29 generations, the model outputs that the optimal running time at this time is 31ms, and at this time, the number of text transmissions is 5 times, and the similarity is 86.7%.

In summary, using the adaptive particle swarm algorithm with few control parameters, strong optimization seeking ability, and small local convergence, the text transmission evaluation model based on Gaussian process (i.e., GFCT Model) is optimized and calculated to endow it with the ability of deep learning, and by citing the example of calculation, it is concluded that there is a big difference in the calculation results before and after the optimization of the algorithm, and the runtime before the optimization is longer, which is up to 58ms, when the model iterates to 29 generations, the model outputs that the shortest running time at this time is 31ms, the number of text transmissions at this time is 5 times, and the similarity is 86.7%.

This paper first constructed based on the immune frog jump algorithm model optimization calculation of the model constructed in problem 1, make full use of the global search capability of SFLA and the local search capability of the immune evolutionary algorithm, taking into account the balance between the global search in the evolutionary strategy and the local search capability asked, and in the process of evolution adaptive adjustment of the step factor to improve the algorithm's efficiency of the search for the optimal to accelerate the vector space based TF- IDF similarity evaluation model based on vector space. Secondly, through the example calculation (800 characters), it is concluded that when the model is iterated to the 15th generation, the minimum running time of the model is 21.6ms, and the similarity of the two texts at this time is 93.79%, which can be concluded that the difference value between the two different versions at this time is 6.21%. Furthermore, the construction of adaptive PSO algorithm on the Gaussian process based text transmission evaluation model (i.e., GFCT Model) to optimize the calculation, endowed with the ability of deep learning, through the example of the algorithm, concluded that the algorithm before the optimization and the algorithm after the optimization of the results of the calculation of the existence of a large difference between the pre-optimization and post-optimization, optimization of the runtime before the longer time, up to 58ms, when the model iterative to the 29th generation, the model outputs at this time, the shortest running time is 31ms, the number of text transmissions at this time is 5 times, and the similarity is 86.7%.

6. Model Strengths and Weaknesses (Evaluation)

6.1 Advantages of the Model

1) The TF-IDF similarity evaluation model based on vector space uses the angle between the vectors to measure the similarity of information matching, the larger the angle the lower the similarity, which

transforms the related problem of the size of the difference between two texts into a vector matching problem in vector space, which can simplify the calculation steps and better realize the quantitative evaluation of the difference;

2) Based on the Gaussian process model has a small sample, model parameters adaptive determination, identification accuracy and other advantages in addition to the ability to give probabilistic significance to the prediction results of the credibility of the prediction accuracy is high, more reasonable to produce five properties of the classification of the prediction results;

3) The immune frog jump algorithm constructed in this paper makes full use of the global search capability of SFLA and the local search capability of the immune evolution algorithm, takes into account the balance between the global search and the local search capability in the evolution strategy, and adjusts the step factor to improve the algorithm's optimization efficiency in the process of evolution, and the application of the immune frog jump algorithm model for iterative calculation of similarity to the whole text can speed up the calculation speed of the TF-IDF similarity evaluation model based on the vector space. -IDF similarity evaluation model based on vector space;

4) The adaptive particle swarm algorithm with few control parameters, strong optimization-seeking ability, and small local convergence is optimized to compute the text transmission evaluation model based on Gaussian process (i.e., GFCT Model), which is endowed with deep learning ability, and greatly improves the speed of the operation and the efficiency of the operation.

6.2 Disadvantages of the Model

1) The TF-IDF similarity evaluation model based on vector space is a method based on statistics, which has a high degree of realism and effect only when the target text contains enough units and the related elements are repeated. At the same time, the TF-IDF method does not take into account the relevant factors (such as meaning, emotion, structure, etc.) that the words themselves have, but only the statistical characteristics of the word occurrence, which has some limitations.

2) The model has more formulas, more complex principles, and is more difficult to program;

3) All models did not go into cross-sectional validation of the data, so there is uncertainty in the models.

7. Reach a Verdict

In this paper, we investigate the problem of text comparison. Firstly, the TF-IDF similarity evaluation model based on vector space is utilized to propose a method for analyzing the size of the differences between the texts of different versions for comparison. Secondly, with the specific content of the two versions of the text known, the similarity evaluation model was applied to analyze and calculate the text, and the Gaussian process-based text transmission evaluation model (i.e., GFCT Model) was constructed for iterative calculation. Finally, we construct the immunity frog jump algorithm model to optimize the similarity evaluation model and the adaptive PSO algorithm to optimize the Gaussian process based text transmission evaluation model (GFCT Model).

Our conclusions are as follows: in the era of information flood, the transmission of information between texts is getting faster and wider, text information workers should strive to adopt innovative science and technology to better analyze and process text information, and help users capture text information more quickly and accurately.

References

Chen C, Seff A, Kornhauser A, et al., (2015). Deep Driving: Learning Affordance for Direct Perception in Autonomous Driving[C]// IEEE International Conference on Computer Vision. *IEEE computer society*, 2722-2730.

Chen Y, Xu Li-wei, (2015). Text classification algorithm for unbalanced forestry information based on optimized LM fuzzy neural network. *Journal of Central South Forestry University of Science and Technology, 35*(04), 27-32+59. DOI: 10.14067/j.cnki.673-923x.2015.005.

Chris H. Q. Ding, (1999). A similarity-based probabilistic model for latent semantic indexing Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 58-65.

Chunhui You, (2008). Text Similarity Calculation Based on Semantic Emotional Tendency. [Master's thesis]. Chengdu: University of Electronic Science and Technology.

Cui Huan, Cai Dongfeng, Miao Xuelai, (2004). Web-based Chinese Q&A System and Information Extraction Algorithm. *Journal of Chinese Information, 18*(3), 24-31.

He Jianjun, Zhang Junxing, Jia Siqi, et al., (2014). A new Gaussian process classification algorithm. *Control and Decision, 29*(9), 1587-1592.

Jin Bo, Shi Yanjun, (2005). Text Likelihood Algorithm Based on Semantic Decomposition. *Journal of Dalian Polytechnic University, 45*(2), 293.

Li Jiayuan, (2014). Chinese Sentence Similarity Calculation Technology and Its Application. Beijing University of Information Science and Technology.

Li Sujian, (2002). Research on utterance relevance based on semantic computing. *Computer Engineering and Application*, (07), 75-76+83.

Li Ting, (2018). Research on multi-AGV path planning and collision avoidance strategy for automated warehouse system. Harbin Institute of Technology.

Liu, Qing Quan, (2020). Application of improved TFIDF-based algorithm in text analysis. Nanchang University. DOI:10.27232/d.cnki.gnchu.2019.000283.

Qin Bing, Liu Ting, Wang Yang, Zheng Shifu, Li Sheng, (2003). Research on Chinese Question and Answer System Based on Frequently Asked Questions Set. *Journal of Harbin Institute of Technology, 35*(10), 1179-1182.

Wang Li-Bureau, (2008). Sentence similarity computation based on semantic analysis tree kernel. [Master's thesis]. Dalian: Dalian University of Technology.

Xue, Huifang, (2011). Research on the theory and application of sentence similarity calculation. Northwest University.

ZHANG Binghui, ZHANG Yan, WANG Wei, et al., (2020). Cave size prediction method based on Gaussian process binary classification model. *China Karst, 39*(02), 259-263.

Zhang Jun, (2011). *Algorithm design and analysis*, pp. 100-215. Tsinghua University Press.

Zhou Fang, (2005). Research on the calculation method of Chinese sentence similarity and its application. Zhengzhou: Henan University.

Copyrights