Computational Intelligence Oriented Intelligent Clustering Method for Keyword Groups in English Language

Hui Zhang[1*] and Fang Zuo[2]

[1]School of Foreign Languages, Sichuan Technology and Business University, Meishan, Sichuan, 620000, China, zuofang@cdutcm.edu.cn
[2]Foreign Language School, Chengdu University of Traditional Chinese Medicine, Wenjiang District, Chengdu, 611137, China, fangzuo512@outlook.com

Corresponding Author: Hui Zhang, zuofang@cdutcm.edu.cn

Abstract: In order to improve the effect of English speech feature extraction, an intelligent clustering method of English keyword group based on computational intelligence is proposed. In addition, this paper also compares the false positive rate, space utilization rate and query time of the two-stage Bloom filtering algorithm and the standard Bloom filtering method. At the same time, this paper analyzes the probability of a certain element of the standard Bloom filter completing the judgment by calculating the hash function t times in the fast judgment mode. Therefore, this paper deduces the average number of hash functions required to calculate the standard Bloom filter and TPBF in the fast judgment mode. After building the model structure, the performance of the model is verified by simulation experiments. The research shows that the English keyword group intelligent clustering method based on computational intelligence has good effect.

Keywords: computational intelligent; English; keyword group; intelligent clustering.
DOI: https://doi.org/10.14733/cadaps.2023.S15.145-162

1 INTRODUCTION

Lexical semantic representation models are mainly divided into distribution-based representation and prediction-based representation. The main idea of distribution-based representation is to consider words with similar contextual content, which themselves are similar [7]. The distributed representation method only describes the co-occurrence frequency of the context distribution as the semantics of the vocabulary, and the vocabulary is only regarded as a simple symbol in the frequency statistics process, regardless of the contextual content information of the vocabulary. Moreover, when based on massive text data, the semantic representation vectors of words are mostly sparse vectors, which will occupy a lot of memory and time in the subsequent calculation process. Therefore, the results of deep semantic discovery of vocabulary based on distributed representation methods are not ideal. The main idea of the prediction-based representation method is to predict the probability of the occurrence of the next word according to the given context, which is mainly based on the neuron distribution assumption of the artificial neural network [2]. This method can not only consider the context, but also the final output result is a low-dimensional real number vector, which solves the problem of high-dimensional sparse data, so this vocabulary representation method is also called word embedding. With the rapid development of deep learning technology, more and more researchers have begun to pay attention to the improvement of vocabulary representation learning methods. They mainly find good features for lexical semantic representation by training on large-scale corpus [5].

The semantic representation vectors obtained from the lexical semantic representation model are also widely used in various practical problems, such as knowledge base construction, machine translation and other practical scenarios. In terms of knowledge base construction, information researchers mainly use vocabulary vectors to mine entity relationships and build link relationships between entities. In order to reduce manual participation, they enrich the existing knowledge base on the basis of vocabulary semantic representation vectors. Research on knowledge base construction based on improved vocabulary vector representation [9]. In terms of machine translation, the use of lexical semantic representation to achieve machine translation on the deep learning model is mainly based on the idea that the original language and target language have the same characteristics. Matching the original words and target words with the same characteristics can not only improve the accuracy of machine translation results, but also reduce the dependence on human participation [3].

As the unit that can best represent the content and subject of a document, keyword analysis is different from other attribute units of a document. The core of keyword analysis lies not only in quantitative analysis at the statistical level, but also in in-depth analysis of its own semantics, which is an inevitable requirement for fine-grained, semantic and intelligent analysis of scientific and technological information. Moreover, keywords not only have important analytical value, but also many other elements (such as authors, institutions, journals, citations) are analyzed based on keyword analysis. Therefore, in keyword analysis of scientific and technological literature, how to express its semantic information is a crucial work [12]. However, previous studies using keywords mainly focused on co word analysis and word frequency statistical analysis using keywords, which only used keywords as simple symbols for statistics and did not involve semantic information of keywords, which may lead to deviation in the results of keyword analysis [1]. In fact, the selection of literature keywords is based on the theme of literature research, so keywords actually have a certain semantic function. However, few researchers pay attention to the semantic function of keywords. Literature [16] classifies the semantic types of keywords in scientific papers, and constructs a multi-dimensional keyword co-occurrence matrix to mine more keyword semantic information; Literature [13] takes literature keywords as the object of subject theme analysis and evolution research, and analyzes the evolution process of subject themes based on keyword semantic information on the basis of adding manual annotation.

As the product of co word analysis, the origin and basic principle of co word matrix should be started from the origin and principle of co word analysis. Co word analysis is not only one of the three quantitative methods of information science, but also one of the common methods of content analysis. It mainly takes keywords in document resources as the analysis object. The theory and application research of co word analysis is also constantly favored by the academic community, and has achieved fruitful research results [8]. From the current research results, the co word analysis method mainly uses the co-occurrence relationship of literature or text keywords to study the research hotspots and topics in the discipline field.

The basic connotation of Co word Analysis is that when two keywords that can express the research theme or direction of a certain discipline appear in a document, it is determined that the two keywords are related [10]. Specifically, the more two or more keywords appear in the same document at the same time, the more relevant the topics represented by these two keywords are; If there are more identical keywords in two or more articles, it indicates that the two articles have certain similarities in research content. It is precisely because the co word analysis method can reflect the relationship between keywords in literature, and thus can also be used to analyze the structural changes of disciplines and topics represented by these words. Therefore, the co word analysis method can be used to aggregate the literature keywords representing the same subject, and then reveal the subject content of a certain subject area, such as which topics are distributed or hot topics. Literature [11] has made an in-depth study on the history of co word analysis. The co word analysis method mainly uses literature keywords to analyze the connection strength between the representative terms of related literature in a certain discipline field. Therefore, it can be considered that one of the main ways of co word analysis is to describe the research topic of a certain discipline field in detail by determining the concept map or knowledge network structure between these representative terms. As the most basic semantic representation method based on distributed hypothesis and the product of co word analysis, the basic idea of co word matrix is that words with similar contexts also have similar semantics. In previous studies, the co word matrix mainly includes "word word" matrix and "word document" matrix. Although the basic unit of analysis in both are keywords appearing in documents, the difference is that keywords in the former are used as feature columns, while documents in the latter are used as feature columns [6]. The "word word" matrix is mainly used to count the number of occurrences of two keywords in a group of words in an article. The "word document" matrix is used to count the number of occurrences of each keyword in a group of keywords in a document. With the development of co - word analysis methods, the co - word matrix as its product has also experienced different variations. Because the absolute co-occurrence frequency cannot accurately reflect the relationship between keywords, many existing analysis based on the co-word matrix mostly weights the keyword word frequency and converts the co-occurrence frequency into a similarity coefficient, so as to convert the co word matrix into a similarity matrix [18].

Literature [17] considered the analysis of co word matrix from the perspective of co word network. Co word network is a network constructed by keyword co-occurrence, which means that there is an edge between keywords that appear together in the same document. The co word network of a research field can explain the trend of knowledge structure, research topic evolution, topic diffusion, etc. of the field through complex network analysis. The small world effect in complex networks shows that there is a shortest path between any two nodes in a large-scale network, which makes the whole network a small world accessible to each other; The scale-free close-up shows that the validity value of nodes in the real network structure conforms to the characteristics of power distribution [4]. Specifically, nodes and edges are the basic elements of a complex network, where nodes refer to a specific individual or thing, while edges represent a relationship between two nodes, and edge weights represent the strength of the relationship between the two nodes. In practical application, in the field of literature cooperation, complex networks mainly include co authorship network, citation network and co word network. Among them, in the co authorship relationship network, the node is the document author, while the edge is the degree of co authorship between authors; In the citation relation network, citations are nodes, and the degree of co citation is the edge; In the co word network, literature keywords are nodes, and the degree of keyword co-occurrence is the edge. The higher the frequency of keyword co-occurrence, the greater the weight of the edge [14]. Therefore, with the continuous development of complex network theory, information researchers began to consider using the traditional common word matrix to build a common word network. Literature [15] uses the traditional co word network for clustering, and uses visual means to analyze and evaluate the research hotspots of international science of science. This paper proposes an intelligent clustering method of keyword groups in English language based on computational intelligent multimedia to improve keyword recognition in English speech.

2 INTELLIGENT RECOGNITION CLUSTERING ALGORITHM FOR ENGLISH KEYWORDS

2.1 Query False Positive Rate Comparison

First, this paper compares the false positive rate of standard Bloom filter and TPBF under different load factors. Standard Bloom filter will produce false positives and false positives. From the basic principle of TPBF, it can be seen that TPBF may produce false negatives and false positives. Therefore, in order to compare the performance of the two with the same standard, the false positive rate is uniformly expressed.
The false positive rate of standard Bloom filter is:

$$p_{BF-ALL}=\frac{n_U\text{-}n_A}{n_U}\times p_{BF} \tag{1}$$

It can be seen from the formula that when $k=(ln2)(m/n)$, the standard Bloom filter has the smallest false positive rate under the given bit vector length m or load factor Load Factor:

$$P_{BF\_ALI.}=\frac{n_U-n_A}{n_U}\times\left(\frac{1}{2}\right)^{(ln2)(m/n)} \tag{2}$$

The minimum false positive rate of TPBF under a given load factor Load Factor is calculated by the formula. The minimum false positive rate of standard Bloom filter under a given load factor Load Factor is calculated by formula (2).
In the following experiments, all set elements are represented in the form of strings. The hash algorithms in standard Bloom filter and TPBF use the APHash algorithm c. According to the formula, $m\geq\lceil 2.082n_f+1\rceil$ can be known, so Load Factor starts from 3.
The data of the random-list.txt file is used as the member set, the number of set elements is $n_A=20000$, and the number of elements of the non-member set (outliers set) is $n_{U-A}=63040$, and the number of elements of the complete set U is $n_U=83040$. The experimental process is divided into two steps:
Firstly, the required bit vector m is calculated according to the set Load Factor, and then the bit vector m is initialized and the elements of the set are inserted, and the corresponding standard Bloom filter and TPBF are generated.
The second step is to implement the query algorithm to evaluate the false positive rate.
Graphical representation of the results can more intuitively show some features of TPBF.

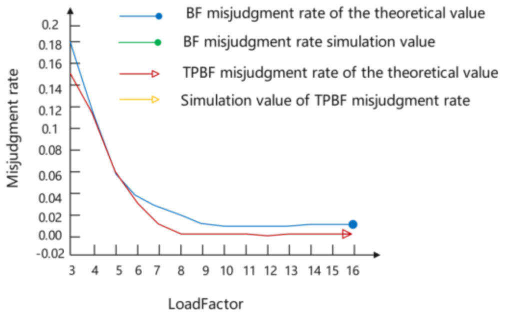

Figure 1: False positive rate of standard Bloom filter and TPBF under different loading factors.

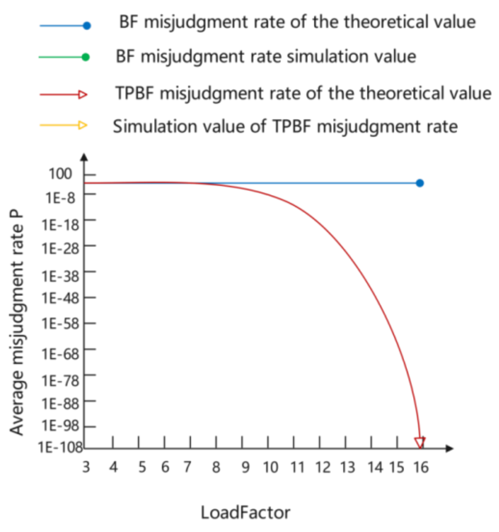

Figure 2: False positive rate (log-view) of standard Bloom filter and TPBF under different loading factors.

From Figure 1 and Figure 2, we can see that under the same load factor, the misjudgment rate of TPBF is always less than or equal to the standard Bloom filter, and the misjudgment rate of TPBF decreases rapidly with the increase of Load Factor.
From formula (2), it can be seen that the misjudgment rate of SBF and TPBF is not only related to Load Factor, but also related to the number n_U of elements in the complete set.

The member set scale factor r is the ratio of the number of elements n_A of the member set to the number of elements n_U of the complete set, that is:

$$r = \frac{n_A}{n_U} \tag{3}$$

Since the value range of n_A is $[1, n_U - 1]$, the value range of r is $\left[\frac{1}{n_t}, \frac{n_U - 1}{n_l}\right]$. When formula (3) is substituted into formula (2), the relationship between the false positive rate of SBF and r under a given loading factor is obtained.

$$P_{BF\_AlL.} = 2^{-L.roall\text{-}acthr(\ln 2)}(1-r) \tag{4}$$

When formula (3) is substituted into formula (4), the relationship between the false positive rate of TPBF and r under a given loading factor is obtained

$$P_{TPBF\_ALL} = e^{\frac{e^{-1+LeadFactor(\ln 2)^2} r}{-1+r}} r \tag{5}$$

It can be seen from the above two formulas that the false positive rate is only related to the load factor and the scale factor r. In order to compare the false positive rate of SBF and TPBF, the ratio of false positive rate of SBF to TPBF can be calculated by formula (4) and formula (5).

$$\frac{P_{BF\_ALL}}{P_{TPBF\_ALL}} = -\frac{2^{-Loadfactor(ln2)} e^{-\frac{e^{-1+-Loadfactor(ln2)^2} r}{-1+r}} (-1+r)}{r} \tag{6}$$

The relationship between the ratio of the false positive rate of SBF and TPBF and the scale factor r is shown in Figure 3.

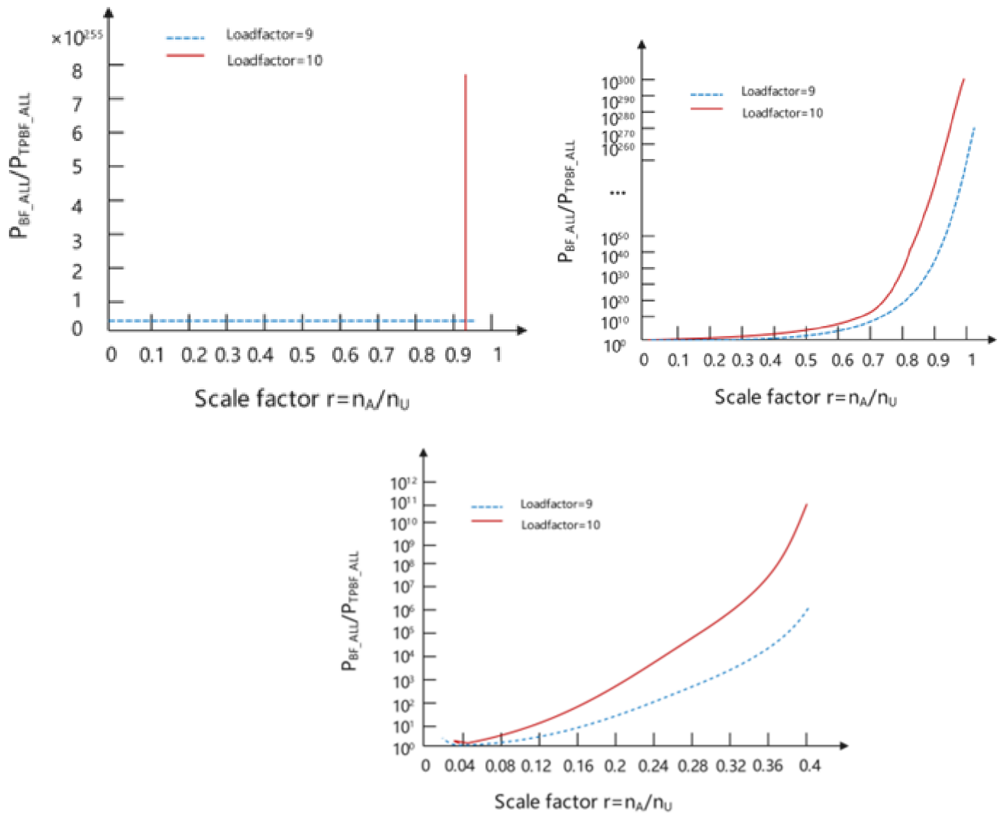

Figure 3: The ratio of SBF and TPBF false positive rate :(a) Scale factor r={0,0.1,..0.99} (NormalView), (b) Scale factor r={0,0.1,..0.99} (LogView), (c) Scale factor r={0,0.1,..0.99} (LogView).

The following conclusions can be drawn from the logarithmic view of Figure 3:

(1) The ratio of the false positive rate of SBF to TPBF increases rapidly with the increase of the scale factor r. It shows that when the same loading factor is used, the false positive rate of TPBF algorithm is significantly lower than that of the standard Bloom filter algorithm for queries with a large proportion of member set A in the whole set U.

(2) In the case of the same scale factor, the larger the load factor Load Factor, the more obvious the improvement of the TPBF algorithm in the false positive rate of the SBF algorithm.

2.2 Storage Space Comparison

The vector length m of Bloom filter, which represents the size of the storage space of the set in the algorithm, can be calculated according to the required false positive rate.

Here, the member set and the complete set are the same, and the vector lengths $m_{TpBF}m$ and m_{kF} required by the TPBF and standard Bloom filter algorithms are compared when the TPBF and the standard Bloom filter meet the given false positive rate.
For a given false positive rate, the required standard Bloom filter vector length can be obtained from formula (2) as:

$$m_{kl} = \frac{n_A \, ln\left(\frac{n_{li} - n_A}{n_{li} P_{Br} - Al.I.}\right)}{(ln\,2)^2} \tag{7}$$

From formula (5), the vector length of TPBF can be obtained as:

$$m_{TPBF} = \begin{cases} \dfrac{n_A\left(1 + 1n\left(\dfrac{(n_A\text{-}n_U)1n\left(\dfrac{n_U P_{TPBF\_ALL}}{n_A}\right)}{n_A}\right)\right)}{(1n2)^2}, & m \geq \lceil 2.082 n_A + 1 \rceil \\ \lceil 2.082 n_A + 1 \rceil, & m < \lceil 2.082 n_A + 1 \rceil \end{cases} \tag{8}$$

Usually, we use Load Factor to measure the space occupancy rate of the algorithm. In order to achieve the specified false positive rates P_{TPBF_ALL} and P_{BF-ALL}, the value of TPBF and the load factor (Load Factor) required by the standard Bloom filter can be obtained by formula (8).

$$Factor_{\gamma_{PBF}} = \frac{1 + ln\left(\dfrac{(n_A - n_{li})\,ln\left(\dfrac{n_l P_{Tp_{Bl}\cdot A+.I.}}{n_A}\right)}{n_A}\right)}{(ln\,2)^2} \tag{9}$$

and $LoadFactor_{TPBF} \geq 2.082 + 1/n_A$.
According to formula (7), we can get:

$$Factor_{BF} = \frac{ln\left(\frac{n_U - n_A}{n_U P_{BF} ALL}\right)}{(ln\,2)^2} \tag{10}$$

In order to obtain the Load Factor required by the standard Bloom filter and TPBF under different false positive rates. The following experiments use the random-list.txt file as a member set for comparative experiments.
In this paper, the data in random-list.txt table is used as the member set, the number of set elements is $n_A = 20000$, and the number of elements to generate an outliers set is $n_{U-A} = 63040$, and the number of elements of the universal set U is $n_{ll} = 83040$.
The data are shown in Figure 4 . It can be seen that in order to reduce the false positive rate to a lower degree, the $LoadFactor_{TPBF}$ required by TPBF grows slowly, while the loading factor $LoadFactor_{BF}$ required by the standard Bloom filter increases significantly faster.

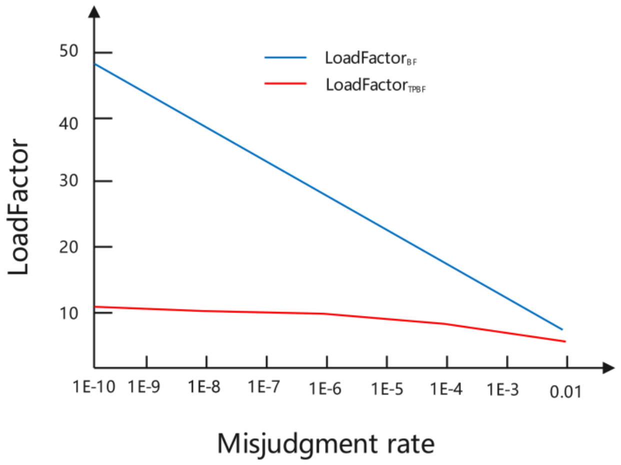

Figure 4: Load Factor corresponding to the specified false positive rate.

It can be seen from formula (9) that the Load Factor required by TPBF is not only related to the member set n_A, but also related to the complete set n_U . Further, the effect of the ratio of the membership set n_A to the full set n_U on the required Load Factor is considered.
The above formula is expressed with a scaling factor r in order to observe the effect of r on the loading factors $LoadFactor_{TPBF}$ and $LoadFactor_{BF}$.
From formula (3) and formula (9), we can get:

$$LoadFactor_{TPBF} = \begin{cases} 1 + ln\left(\dfrac{\left(-n_U + n_U, r\right) ln\left(\dfrac{n_U P_{TP_B F\_AL.I.}}{n_U r} \right)}{n_U r} \right) & LoadFactor_{TPBF} > ln2 + \dfrac{1}{n_U r} \\ ln\, 2 + \dfrac{1}{n_U r}, & LoadFactor_{TPBF} > ln2 + \dfrac{1}{n_U r} \end{cases} \quad (11)$$

From formula (10) and formula (3), we can get:

$$JFactor_{BF:} = \frac{ln\left(\frac{n_{ll} - n_{v}r}{n_{l\rangle}P_{BF} - Al.I}\right)}{(ln\,2)^{2}} \qquad (12)$$

The formula (12) is shown in Figure 6, and the changing trend of the loading factor required by the standard Bloom filter and TPBF can be seen intuitively when the scale factor changes.

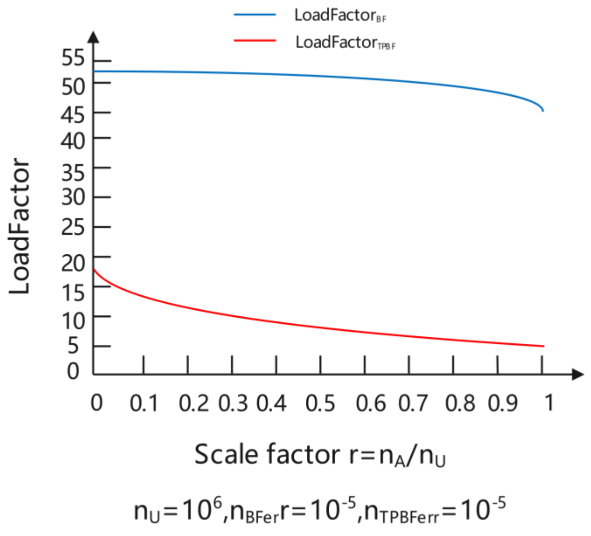

Figure 5: Loading factors required for standard BF and TPBF under different scale factors r.

It can be seen from Figure 5 that when the scale factor r is smaller, that is, the smaller the ratio of the number of elements n_A of the member set A to the number of elements n_U of the universal set U, the larger the required loading factor is.

When $n_A = 1$, that is, $r = 1/n_U$, the required loading factor is the largest.

The Load Factor required by standard Bloom filter and TPBF decreases with the increase of r. When $n_A = n_U - 1$, that is, $r = (n_U - 1)/n_U$, the load factor required by standard Bloom filter is the smallest. However, according to formula (11), the Load Factor of TPBF should be greater than or equal to $\lceil ln\,2 + 1/(n_V r) \rceil$, so when r is close to 1, LoadFactor $_{Tp\beta F} = \lceil 2.081 + 1/(n_U r) \rceil$ will appear.

2.3 Query Time Comparison

From the query process of the standard Bloom-filter, we can know that the query time T of an element is:

Query time $T = Hash$ function calculation time $t_{Hash} \times Hash$ function calculation times $C +$ operation time $T_{\cap}$ +parameter passing time $T_{argumen}$

The AND operation time $T_{\cap}$ and the parameter transfer time $T_{argumen}$ are very fast and account for a small proportion of the query time T. In order to facilitate the analysis and ignore it, the calculation formula of the query time is simplified as:

$$T_{Bl} = t_{Heasi} \times C_{BF} \tag{13}$$

Similarly, from the query work process of TPBF, we can get:

$$T_{TPRF:} = t_{Hesh} \times C_{TPBr} \tag{14}$$

The ratio of the time it takes to judge an element to the time it takes for a standard Bloom filter to judge an element is the query time ratio λ.

$$\lambda = T_{TPBF} / T_{BF} = C_{TPBF} / C_{BF} \tag{15}$$

The ratio less than 1 means that TPBF takes less time to judge an element than standard Bloom filter. A equal to 1 means that standard Bloom filter and TPBF will take the same time to judge an element, and if the ratio is greater than 1, it means that TPBF takes longer to judge an element than standard Bloom filter. Therefore, a smaller query time ratio A indicates that TPBF is more efficient in query judgment time than standard Bloom filter.

In the performance analysis of the query time in this paper, the calculation times C_{BF} and C_{TPBF} of the Hash function will be mainly considered.
The number of Hash functions of the standard Bloom filter is k, and the number of Hash function calculations required to query an element is a fixed value, which is

$$C_{BF} = k \tag{16}$$

From the false positive rate P_{BF_ALL}, we can get:

$$k = \frac{ln\left(\frac{n_U - n_A}{n_i P_{BF\_} .4LL}\right)}{ln(2)} \tag{17}$$

Therefore, the number of Hash function computations C_{Br} required by the standard Bloom filter to query an element is

$$T_{BF} = \frac{ln\left(\frac{n_{II} - n_A}{n_{Ii} P_{BF\_+LL}}\right)}{ln(2)} T_{h\omega\omega h} \tag{18}$$

The theoretical estimate of the average number of judgments during TPBF query is:

For TPBF, the number of Hash functions of BF1 is $k1$, and the number of Hash functions of BF2 is $k2$. Due to the use of two-stage judgment, the calculation time is not fixed. If the element to be checked is a member of the set (U-A-E), it only needs to be judged by BF1 and come to the conclusion that it is not a member of the set, so that k_1 times of Hash function operations can complete the judgment. If the element to be checked is a member of the set (A+E), it needs to be judged by BF1 and BF2 respectively. In this way, $k_1 + k_2$ hash operations are required to complete the judgment of whether the element to be checked is a member of set A. When the elements to be checked are evenly distributed in U, the probability of $\left(n_{lJ} - n_A - n_{l:}\right)/n_{lj}$ requires k_1 hash operations to complete the judgment, and the probability of $\left(n_A + n_{l:}\right)/n_l$ requires $k_1 + k_2$ hash operations to complete the judgment. Therefore, the average number of hash functions required to query an element is:

$$C_{Trpll} = \frac{\left(n_u - n_A - n_{l:}\right)}{n_n} k_1 + \frac{\left(n_A + n_E\right)}{n_{ll}} \left(k_1 + k_2\right) \tag{19}$$

For a given average false positive rate P_{TPBF} ALLL, the corresponding k_1、k_2 and n_F can be obtained. The number k_1 of Hash functions of BF1 can be obtained from the formula.

$$k_1 = \frac{ln\left(\frac{\left(n_A - n_{l:}\right) ln\left(\frac{n_V P_{TPBF} + AII}{n_A}\right)}{n_A}\right)}{ln 2} \tag{20}$$

The number n_U of the misjudgment set E of BF1 is obtained by the formula:

$$n_{l:} = -\frac{n_4}{ln\left(\frac{n_{li} P_{TPB;}}{n_A}\right)} \tag{21}$$

The number of hash functions k_2 of BF2 can be obtained from the formula:

$$k_2 = -\frac{ln\left(\frac{n_{ii} P_{TPBF_A Al..L}}{n_4}\right)}{ln 2} \tag{22}$$

When formula (20), formula (21) and formula (22) are substituted into formula (19), we get:

$$C_{TP_{BI}} = \frac{n_A - n_A\, ln\left(\frac{n_{ll} P_{TP_{BF}-ALII}}{n_A}\right) + n_{ll}\, ln\left(\frac{(n_A - n_{ll})\, ln\left(\frac{n_{li} P_{TPBF_i+IIII}}{n_A}\right)}{n_A}\right)}{n_U\, ln 2} \tag{23}$$

Therefore, the time required to complete a TPBF query for one element is:

$$T_{TPBF} = \frac{n_A - n_A\, ln\left(\frac{n_U P_{TPBF-ALL}}{n_A}\right) + n_U\, ln\left(\frac{(n_A - n_U)\, ln\left(\frac{n_U P_{TPBF\_ALL}}{n_A}\right)}{n_A}\right)}{n_U\, ln 2} T_{hash} \tag{24}$$

whole set U is small, and the standard Bloom filter may produce a false positive part (U-A) is large. This requires the standard Bloom filter to use more Hash functions to achieve the specified false positive rate. The part of TPBF that will cause misjudgment is the element in the member set A. The small set A can make the elements in the set (U-A-E) only calculate the k_1 hash functions in BF1 to obtain a negative judgment. These elements do not need to be judged in the second stage. Only when the element to be checked is in the set (A+E), the second-stage k_1 Hash function calculations will be performed. Since k_1 is much smaller than k_2, the set (U-A-E) at this time is also large, so that the overall query time of the uniformly distributed elements to be checked using TPBF is shorter than that of the standard Bloom filter.

For each value of r, we calculate the scale $n_A = r \times n_U$ of set A according to the scale factor r, and take n_A integers starting from 1 as the elements of set A. Then, we use standard Bloom filter and TPBF to judge all elements in the corpus U, respectively. Since this experiment is to compare the query time performance of the two, in order to avoid the impact of other operations on the time performance test, here is only to judge whether the element to be tested belongs to the element of set A and no longer consider whether there is a misjudgment. In order to obtain the query time of each element, we use the following method: before the query process starts, the time T_{sturr} r is recorded to the millisecond, and the time T_{end} is recorded again after the judgment of n_U elements is performed. The total time $T_{end} - T_{start}$ for judging n_{li} elements can be obtained. The above process is repeated 10 times, so the average time spent on each element is:

$$T_{real\_query} = \sum_{i=1}^{10} \left(T_{end_1} - T_{starl,}\right) / \left(10 n_{vJ}\right) \tag{25}$$

In this way, the actual average query time $\left(T_{real_BF}\right)$ of each element of the standard Bloomfilter and the actual average query time T_{real_TPBF} of each element of the TPBF can be obtained according to the formula calculation.
The computer hardware and software configuration in this experiment is IntelE8400, 2GRAM, Windows2003. Figure 6 shows the average judgment time obtained by simulating the standard Bloom filter and TPBF with false positive rates 10^{-10} and 10^{-4} respectively.

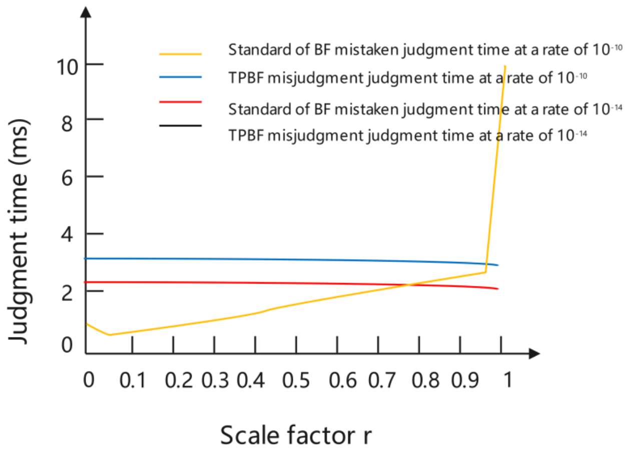

Figure 6: Judgment time under all hash function methods.

As can be seen from Figure 6, for the standard Bloom filter and TPBF that meet the same number of false positives, their hash function calculation times will be equal at a certain r value. The false positive rate 10^{-10} is specified in this experiment and $T_{real_T PBF}$ will not be greater than $T_{real_B F}$ until $r \geq 0.76$. When specifying a false positive rate 10^{-14}, $T_{real_T PBF}$ will not be greater than $T_{real\ BF}$ until $r \geq 0.82$.

In order to avoid that the query judgment time of TPBF at $\lambda > 1$ is greater than that of standard BF, before generating a TPBF instance, formula (23) can be used to calculate the average calculation times $C_{TPBF}\left(U, A, P_{TPBF\ ALL}\right)$ of the hash function of the TPBF set A, and at the same time, the calculation amount $C_{TPBF}\left(U, U-A, P_{TPBF\_ALL}\right)$ of the TPBF set (U-A) can be calculated. If $C_{TPBF}\left(U, A, P_{TP_B F\_ALL}\right)$ is less than $C_{TPBF}\left(U, U-A, P_{TPBF\_ALL}\right)$, the algorithm directly generates an instance of $TPBF_{(U,A)}$ without misjudgment in the normal way, and judges whether the element to be checked is in set A. Otherwise, the algorithm uses the set (U-A) as the member set of TPBF to generate a false positive instance of $TPBF_{(U,U-A)}$, and then judges whether the element to be checked is in the set (U-A). If it is not, the element to be checked belongs to set A. If it is in set (U-

A), it is definitely not in set A, so that the correct judgment result can be obtained with less calculation time.
When the misjudgment rate is large, there are more elements that cause misjudgments, resulting in a larger number of elements that need to be judged for k-th hash function. When the misjudgment rate is very low, there are very few misjudged elements, and most of the elements that do not belong to set A are determined to be elements that do not belong to set A in less than k hash function calculations.

Therefore, the elements in the universal set U are judged as the total number C_{BF_Accept} of Hash function calculations performed in the set A as:

$$C_{BF-Accept}=\left(n_A+n_{E:}\right)\times k \tag{26}$$

The total number C_{BF_Rejec} of Hash function calculations performed by elements that are judged to be not in set A is:

$$C_{BF_{-refec}}=\left(n_{li}-n_A-n_{E:}\right)\times\sum_{i=1}^{k}\left(t\times 2^{-1}\right) \tag{27}$$

Then, the total number C_{BF} of Hash function calculations that need to be performed for all elements in the full set U through the standard Bloom filter judgment is:

$$C_{BF}=C_{BF\_accept}+C_{BF\_reject} \tag{28}$$

In this way, the standard Bloom filter algorithm can obtain the average number C_{BF_mean} of hash functions that needs to be calculated for a uniformly distributed element to be calculated as:

$$C_{BF-mecui}=C_{BF:}/n_{li} \tag{29}$$

Taking the number of elements $n_{t=}=1000000$ of the complete set U and the number of elements $n_A=300000$ of the set A as an example, if the specified false positive rate is g, according to formulas (28), (29) and (30), the standard Bloom filter algorithm can obtain the theoretical value $C_{HI_m eam_d erived}$ of the number of hash functions that needs to be calculated on average to judge a uniformly distributed element, which is 4.09198.

When the TPBF algorithm adopts the fast judgment method, if $E1$ and $E2$ are the misjudgment sets of BF1 and BF2, respectively, and k_1 and k_2 are the optimal hash numbers of BF1 and BF2, respectively, the elements in the universal set U can be divided into the following four cases:
1. The number of calculations of the element hash function in the set (U-A-E1) is less than or equal to k_1, and the total number of calculations $C_{TP_{BF}F_{-BF}I\_Re\,ject}$ is:

$$C_{TP_{BF}BF1\_R_{eject}}=\left(n_{UV}-n_A-n_{E1}\right)\times\sum_{t=1}^{k_1}\left(t\times 2^{-t}\right) \tag{30}$$

2. For the elements in the set $(A+E1)$, the number of calculations of the hash function is equal to k_1, that is, the total number of calculations $C_{\_TPBF\_BFI\_ACCEPT}$ is:

$$C_{TPBF_{-BF1_1}+ACEPT} = \left(n_A + n_{E1}\right) \times k_1 \tag{31}$$

3. Since the elements in the set $(A+E1)$ cannot be distinguished by BF1, a second-stage judgment is required. Then, the elements in the set (A-E2) are judged by BF2 as not members of the set $E1$, so

$$C_{rPBFF-BF2..R_e/CCl} = \left(n_{+1} - n_{F_{:2}}\right) \times \sum_{i=1}^{k_2}\left(1 \times 2^{-t}\right) \tag{32}$$

4. The elements in the set (E1+E2) are judged by BF2 as members of the set E1, so the total number of computations $C_{|PBF|:.B1/2+Accep1}$ of the elements in the set (E1+E2) in BF2 is:

$$C_{TPBF:-BF\cdot 2\_Hceep} = \left(n_{F:1} + n_{F:2}\right) \times k_2 \tag{33}$$

Therefore, the total number of times C_{Trpl} of the hash function calculation of the universal set U is:

$$C_{TPBF} = C_{TPBF\_BF1\_Re\,ject} + C_{TPBF\_BF1\_Accept} + C_{TPBF\_BF2\_Re\,ject} + C_{TPBF\_BF2\_Accept} \tag{34}$$

In this way, it can be obtained that the average number $C_{TPBR-me+m}$ of hash functions required to be calculated by the TPBF algorithm to judge a uniformly distributed element is:

$$C_{TPBF\_mean} = C_{TPBF} / n_U \tag{35}$$

3 THE INTELLIGENT CLUSTERING METHOD OF KEYWORD GROUPS IN ENGLISH LANGUAGE BASED ON COMPUTATIONAL INTELLIGENT MULTIMEDIA

Figure 7 shows the query framework based on negative keywords. It consists of three parts: iterative update module, storage management module and query engine module. The iterative update module is used to support the function of updating and deleting the index, and the user can perform spatial English speech query with negative keywords through the query engine module.

The above constructs an intelligent clustering method of keyword groups in English language based on computational intelligent multimedia, and then analyzes the effect of the model. In this paper, the intelligent clustering analysis of keywords is carried out through multiple groups of English phonetics, and the clustering effect is counted, and the results shown in Table 1 are finally obtained. It can be seen from the above research that the intelligent clustering method of keyword groups in English language based on computational intelligent multimedia proposed in this paper has a better effect.

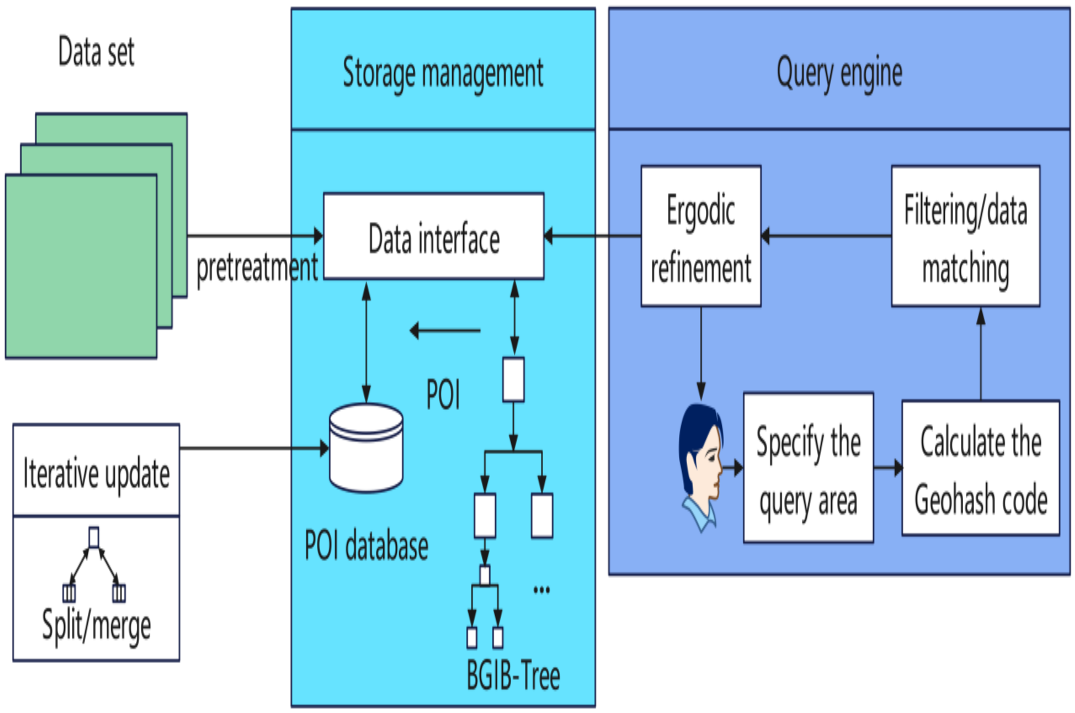

Figure 7: Query framework of English speech keywords.

4 CONCLUSION

This paper proposes a spatial keyword query framework combining negative keywords. At the same time, the query algorithm is mainly based on the improved spatial index, and the query speed of spatial location is faster and faster. However, most keyword string matching is based on complete matching, and only a small part of keyword matching is considered to meet user needs. The complete matching and partial matching of keywords belong to complete matching, and polysemy is not uncommon in queries. Considering that users may make input errors when inputting, this paper proposes an intelligent clustering method of English keyword group based on computational intelligence to improve the effect of English speech keyword recognition. The research shows that the intelligent clustering method of English keyword group based on computational intelligence proposed in this paper has good effect.

Hui Zhang, https://orcid.org/0000-0002-4612-9518
Fang Zuo, https://orcid.org/0000-0001-5968-3764

Num	Cluster analysis	Num	Cluster analysis	Num	Cluster analysis
1	89.557	25	93.219	49	92.030
2	93.423	26	95.841	50	90.930
3	92.874	27	93.226	51	95.803
4	92.318	28	89.543	52	90.063
5	94.406	29	92.756	53	91.060
6	92.870	30	89.881	54	91.769
7	94.528	31	94.618	55	89.208
8	93.225	32	89.061	56	95.131
9	95.660	33	93.804	57	89.706
10	90.895	34	91.792	58	89.584
11	91.103	35	89.072	59	91.879
12	89.417	36	92.274	60	94.679
13	90.314	37	93.041	61	89.615
14	91.095	38	92.446	62	93.316
15	92.216	39	91.024	63	92.265
16	93.490	40	94.338	64	94.466
17	91.253	41	91.016	65	93.766
18	90.734	42	89.883	66	89.179
19	90.900	43	94.643	67	92.284
20	93.796	44	94.356	68	89.882
21	94.624	45	93.791	69	91.632
22	90.417	46	92.047	70	94.777
23	94.609	47	90.199	71	94.887
24	95.842	48	95.792	72	92.348

Table 1: Evaluation of the effect of intelligent clustering method of keyword groups in English language based on computational intelligent multimedia.

REFERENCES

[1] AlGhamdi, M. A.: Arabic Learners Preferences for Instagram English Lessons, English Language Teaching, 11(8), 103-110. https://doi.org/10.5539/elt.v11n8p103

[2] Apriani, E.; Hidayah, J.: The ICT Used by the English Lecturers for Non-English Study Program Students at STAIN Curup, Vision: Journal of Language and Foreign Language Learning, 8(01), 2019, 26-37. https://doi.org/10.21580/vjv8i13280

[3] Bin, Y.; Mandal, D.: English teaching practice based on artificial intelligence technology, Journal of Intelligent & Fuzzy Systems, 37(3), 2019, 3381-3391. https://doi.org/10.3233/JIFS-179141

[4] Dinara, K.: Linguo-cultural study of anthroponyms in English and Uzbek, Ijodkor o ʻqituvchi, 2(19), 2022, 431-435.

[5] Godwin-Jones, R.: Second language writing online: An update, Language Learning & Technology, 22(1), 2018, 1-15. https://dx.doi.org/10125/44574

[6] Khasawneh, M. A. S.: An electronic Training Program on Developing the Written Expression Skills among a Sample of foreign language learners EFL who are at-risk for Learning disabilities during the emerging Covid-19, Academy of Social Science Journal, 7(10), 2021, 1974-1982. https://doi.org/10.15520/ASSJ.V7I10.2713

[7] Liu, S.; Wang, J.: Ice and snow talent training based on construction and analysis of artificial intelligence education informatization teaching model, Journal of Intelligent & Fuzzy Systems, 40(2), 2021, 3421-3431. https://doi.org/10.3233/JIFS-189380

[8] Meng-yue, C.; Dan, L.; Jun, W.: A Study of College English Culture Intelligence-Aided Teaching System and Teaching Pattern, English Language Teaching, 13(3), 2020, 77-83. https://doi.org/10.5539/elt.v13n3p77

[9] Pérez-Paredes, P.; Ordoñana Guillamón, C.; Aguado Jiménez, P.: Language teachers' perceptions on the use of OER language processing technologies in MALL, Computer Assisted Language Learning, 31(5-6), 2018, 522-545. https://doi.org/10.1080/09588221.2017.1418754

[10] Rinantanti, Y.; Bin-Tahir, S. Z.; Suriaman, A.: The Impact of EFL Senior High School Teachers' Performance in Papua, Indonesia toward the Students' English Learning Achievement, Asian EFL Journal, 23(3.3), 2019, 431-447.

[11] Shadiev, R.; Yang, M.: Review of studies on technology-enhanced language learning and teaching, Sustainability, 12(2), 2020, 524-532 https://doi.org/10.3390/su12020524

[12] Todd, R. W.: Teachers' perceptions of the shift from the classroom to online teaching, International Journal of TESOL Studies, 2(2), 2020, 4-16. https://doi.org/10.46451/ijts.2020.09.02

[13] Toto, G. A.; Limone, P.: Motivation stress and impact of online teaching on Italian teachers duringCOVID-19, Computers, 10(6), 2021, 75-94 https://doi.org/10.3390/computers10060075

[14] Tseng, S. S.; Yeh, H. C.: The impact of video and written feedback on student preferences of English-speaking practice, Language Learning & Technology, 23(2), 2019, 145-158. https://doi.org/10125/44687

[15] Xiao-Dong, L.; Hong-Hui, C.: Research on VR-supported flipped classroom based on blended learning—a case study in learning English through news, International Journal of Information and Education Technology, 10(2), 2020, 104-109. https://doi.org/10.18178/ijiet.2020.10.2.1347

[16] Xu, Z.; Shi, Y.: 2018, Application of constructivist theory in flipped classroom-take college English teaching as a case study, Theory and Practice in Language Studies, 8(7), 2018, 880-887. https://dx.doi.org/10.17507/tpls.0807.21

[17] Yu, L.; Wu, X.; Yang, Y.: An online education data classification model based on TrMAdaBoost algorithm, Chinese Journal of Electronics, 28(1), 2019, 21-28. https://doi.org/10.1049/cje.2018.06.006

[18] Zhao, H.; Liu, Z.; Yao, X.; Yang, Q.: A machine learning-based sentiment analysis of online product reviews with a novel term weighting and feature selection approach, Information Processing & Management, 58(5), 2021, 102656. https://doi.org/10.1016/j.ipm.2021.102656