Research on Fuzzy Retrieval Method of Blockchain Information Based on B+Tree Index Structure

Jia-hua Li(✉)

School of Information Engineering, Guangzhou Vocational and Technical University of Science and Technology, Guangzhou 510550, China
lijiahua21223@163.com

Absrtact. Due to the large scale of blockchain information, some traditional retrieval methods are difficult to play a role, so it is necessary to use distributed retrieval method for information retrieval to effectively meet the needs of big data retrieval. Therefore, a blockchain information fuzzy retrieval method based on B+tree index structure is studied. This method first analyzes the problems of blockchain information retrieval, and then uses B+tree index structure to build an index about the target text, and then expands the key words to achieve the fuzziness of the key words and improve the comprehensiveness of the retrieval. Finally, the similarity between the key words and the index is calculated to achieve fuzzy information retrieval. The experimental results show that the retrieval accuracy and comprehensiveness of the research method have been effectively improved, and the research goal has been achieved, and the method can effectively meet the needs of big data retrieval.

Keywords: B+tree index structure · Blockchain · Fuzzy retrieval method

1 Introduction

With the rapid development of information technology, more and more data needs to be stored on cloud servers, but the existing cloud storage servers are not completely trusted. In order to realize the confidentiality of data, the data owner encrypts the data and sends the data to the cloud storage server in the form of ciphertext, so that the cloud server cannot obtain the content of the plaintext data, but how to retrieve the information on the cloud server is a difficult problem [1]. Information retrieval is a method by which the user retrieves the required information from the data set. The user generates the query request according to his own needs and sends the query request to the server that has the database, and then the server returns one or more relevant results to the user. With the popularization of the Internet, online information retrieval has become very common in people's life. At the same time, the user's retrieval privacy is more difficult to be protected. Content providers refine their profiles by keeping a

S. Wang et al. (Eds.): IoTCare 2021, LNICST 414, pp. 308–325, 2022.
https://doi.org/10.1007/978-3-030-94185-7_21

record of what users have searched for. These ISPs can use these user profiles to make money by providing targeted advertising or content recommendations. In some cases, users want their searches to be private. For example, investors looking at the current market value of certain stocks in a stock market database may be reluctant to disclose that they are interested in that stock because this could inadvertently affect the stock price. When a user submits a query request to the server, he hopes that his query information will not be disclosed. Therefore, in order to solve these problems, a variety of new information retrieval methods have been proposed in related fields. For example, some scholars proposed symmetric searchable encryption algorithm for the first time to solve the problem of ciphertext search, but the search efficiency of this scheme is very low based on linear scan. Some scholars put forward the definition of safe index, and use Bloon filter to construct safe index, but there are some errors in the search results. Some scholars have also proposed a searchable scheme of linked keywords, which allows users to retrieve multiple keywords, but the location of keywords needs to be specified in the scheme. In order to further improve the ability of information retrieval, some scholars proposed a single keyword searchable encryption scheme on the blockchain, which realized the data retrieval on the blockchain, but the search results were inaccurate and the search efficiency was low. In order to solve this problem, some scholars proposed a multi-keyword sorting search scheme which supported dynamic updating. In this scheme, the balanced binary tree was used to build the index and the greedy depth-first search algorithm was used to sort the retrieval results.

Based on the predecessors' research experience, based on the current information sharing as the low efficiency of data search and search results are biased, put forward a kind of based on B+tree index structure blockchain fuzzy information retrieval method, in order to use B+tree index structure improve the efficiency of the cipher text search, fully to solve the problems existing in the traditional method, can be widely applied to medical data, and other fields, it has certain contribution to the further development of information technology.

2 Problems in Blockchain Information Retrieval

This section focuses on the organization, storage and management of blockchain data, and takes Ethereum as an example to analyze and study the blockchain query mode, from the underlying storage to the abstract data structure, and then to the specific data fields; Then three solutions are summarized and analyzed from different angles; Finally, some ideas for the future development of Block DataBase are put forward, in order to provide some reference for the development of this field. The problems faced by blockchain information retrieval are shown in Fig. 1.

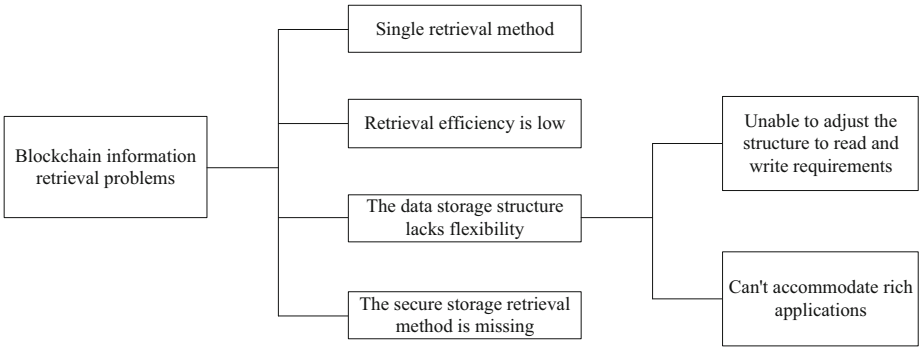

Fig. 1. Problems faced by blockchain information retrieval

2.1 Single Retrieval Method

The current blockchain system has a single retrieval method, which only supports a limited number of fixed fields as the key matching search, can not support complex queries, is not suitable for rich applications, and can not meet the needs of complex data analysis. Ethereum query specific flow: first, the query request is parsed to get the key; Then we call the Get (key) method of Leve1DB to get the corresponding Value. Finally, the RLP deserialization and decoding of the Value are carried out, and then the Json code is encapsulated and returned to the client.

At present, the data field design of blockchain does not consider the query requirements of data analysis at the beginning. It only supports a limited number of fields to meet the basic functions, and does not support user-defined fields. The type is fixed and relatively single, and there is basically no scalability. Moreover, it can not support complex analytical queries such as Range query, Top-k query or fuzzy query. If you want to meet other complex application scenarios or complex query requirements, you need to encapsulate the retrieval layer and build the corresponding index structure in the upper layer of Leve1DB, which reduces the attraction of blockchain system to rich applications.

2.2 Low Retrieval Efficiency

At present, the underlying storage databases used by mainstream blockchain platforms are mostly key value databases such as leve1db, which are suitable for write intensive business requirements, and the read performance is not excellent. However, the write concurrency of current blockchain systems is not large. For example, the write concurrency of bitcoin transactions is only about 1 transaction per second, and that of Ethereum is about 7 transactions per second. The data storage structure lacks flexibility. When the data in the blockchain system is increasing and the application is expanding, the query demand will increase. At this time, the defects of excessive write performance and insufficient read performance of the underlying storage system become the key factors limiting the query performance.

2.3 Data Storage Structure Lacks Flexibility

(1) Unable to structure for read and write requirements
The current situation of the Internet is that both the reading and writing needs are taken into account, but the reading and writing of the blockchain system is unbalanced, and it can not meet the flexible adjustment according to the situation. Due to the numerous application fields of blockchain system, their requirements for read-write performance are different. For example, in e-commerce payment system, the transaction throughput will be very large, which requires frequent write operations and relatively few query requests; In the bank's settlement system, the number of transactions per unit time is limited, so the pressure on the write performance will be reduced. However, due to the large number of user queries, the complex audit process and frequent analysis and operation of the salesman, the query is needed, which puts forward higher requirements for the read performance. However, the performance of the existing blockchain system is difficult to meet the requirements of read-write performance adjustment.

(2) Unable to adapt to rich applications
At present, the number of fields supported by blockchain data is small, the structure is relatively fixed, and the query processing logic is relatively simple. With the expansion of the scope of application, the data structure of the application will be more complex. The current system can not add and manage user-defined fields, and it is difficult to provide effective support for new applications.

2.4 Lack of Secure Storage Retrieval Method

At present, the data of blockchain system is mainly divided into two categories: one is completely open, which can be accessed by anyone and stored in clear text, such as bitcoin, Ethereum and other public chains; One is authorized access, and the data that can be obtained is also stored in clear text. Whether it is internal leakage or hackers' intrusion, the data will be directly exposed.

3 Design of Blockchain Information Fuzzy Retrieval Method

With the unique advantages of de trust, weak centralization, data traceability and anti tampering, blockchain is gradually becoming a popular platform for data storage and sharing. Reasonable data organization and management and perfect data privacy protection mechanism are the key factors for the data platform to be recognized. However, due to the decentralized digital currency at the beginning of the design, the data organization form of blockchain is solidified, resulting in single retrieval method and lack of effective data privacy protection design, which hinders the implementation of blockchain in related fields with strong data security requirements and data query requirements, such as smart medicine, financial technology and e-government [2]. The searchable encryption technology can make the data meet the requirements of secure storage and support efficient retrieval, but at present, most of the researches are focused on the searchable encryption in the cloud environment, and there is a lack of searchable Encryption Research for the blockchain data platform. Therefore, this paper studies a

method to improve the efficiency and accuracy of keyword retrieval while achieving the security of ciphertext retrieval. In terms of process, the method starts from the data provider hosting the document data to the cloud server, and then the cloud server returns the query results to the user. There are five steps to complete a cycle of data hosting and query. The specific work flow is as follows:

(1) The data provider uploads the encrypted data file and the encrypted index corresponding to the data file. The data owner needs to extract the keywords from the file set F, use the edit distance algorithm to generate the fuzzy keyword set, and construct it as an index file. Then the data provider encrypts the file set F and stores it in the cloud computing server in the form of ciphertext to protect the data provider's document privacy information. The encrypted document set is E. In order to retrieve the encrypted document set E, the data provider needs to establish a retrievable encryption index for F before encrypting the document set F. The retrievable encryption index refers to I. Then the data provider hosts the encrypted document set E and the corresponding retrievable encryption index I to the cloud computing server [3].
(2) When the user queries, the search request W_a is sent to the data provider through the encrypted connection. In order to query the file data information, the corresponding query encryption index needs to be constructed. Only the data provider can build the retrievable encrypted index and query encrypted index of the document, so the user needs to send the query keywords to the data provider [4]. In order to ensure the security of users' query privacy transmission on the network, data providers and data users communicate directly through encrypted connections.
(3) When the data provider receives the transmitted search request W_a, it returns the encrypted query index. The data provider constructs the corresponding query encrypted search request T_W according to the query keywords sent by the user, and returns it to the data user.
(4) The user sends the encrypted search request T_W to the cloud computing server. The user receives the encrypted query request T_W returned by the data provider and directly sends it to the cloud computing server for query request without further operation.
(5) The cloud computing server returns the query result. After receiving the user's search request T_W, the cloud computing server matches the user's search request T_W with the retrievable encryption index I under the encryption condition, and then returns the encrypted file name sorted by relevance to the user [5].

The keyword oriented ciphertext retrieval scheme can be roughly divided into three stages: index structure establishment; The key words are fuzzy; Matching retrieval is realized. The following is a detailed analysis of these three processes.

3.1 Index Structure Establishment

Index technology is one of the key technologies of modern information retrieval, search application and data mining. Inverted index and B+tree index are two widely used index technologies. Among them, inverted index technology has the advantages of relatively simple implementation, fast query speed and easy to support synonym query. It is widely used in large-scale document set retrieval and other information retrieval. In addition, B+tree is a variant of B tree, which is suitable for the application environment of random and sequential processing, and it has a wide range of applications in database system, file system indexing and so on [6]. In theory, it can index the attributes of any data type, and keep the index level corresponding to the size of the data file. Search has stable I/O overhead, good support for index update, and can bear a variety of workload.

Design Idea of Index Structure

The index object contains two kinds of data, namely character data and numerical data. This paper adopts the idea of layering in the design of index structure. That is to say, the first layer of the index structure indexes the attributes to which the data belongs, and the second layer indexes the attribute values corresponding to the data attributes of the first layer. At this time, if the attribute value belongs to character type data, the inverted index is established. If the attribute value belongs to numerical type data, the B+tree index is established. In this way, it is different from the traditional B+tree structure to build B+tree index for all different types of data. The index structure does not establish B+tree index for all character data, so the time cost caused by node splitting can be avoided, and the extra storage space occupied by temporary nodes in the process of node splitting is also relatively reduced. Therefore, the index structure improves the index creation speed and space utilization. Inverted index has good performance for character data retrieval, but it can not meet the requirements of cross data range retrieval for numerical data. B+tree has advantages in numerical data retrieval because of the order of leaf nodes. Therefore, the hybrid index structure in this paper adopts the idea of layered and building different index structures for different types of data to complete the cross data range retrieval, that is, all numerical data are built B+tree index in the second layer. When the cross data range retrieval is needed after the index structure is built, the task will be completed by the B+tree index part of the second layer. The hybrid index structure makes use of and inherits the advantages of the two structures, while discarding the disadvantages of the two structures. In order to achieve efficient index creation speed and space efficiency at the same time, it can complete the requirements of cross data range retrieval. The hybrid index structure is shown in Fig. 2.

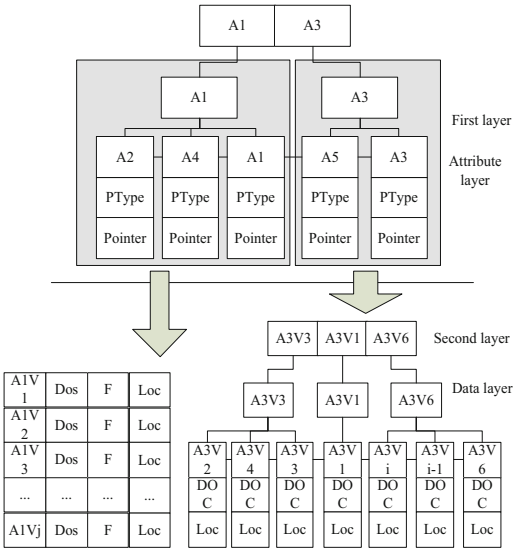

Fig. 2. Hybrid index structure

In Fig. 2, the first layer is to build a B+tree index structure for the attribute of the index object. In the B+tree index structure of this layer, all non leaf nodes store specific attributes, and all leaf nodes contain three parts of information $\langle Ai, Ptype, Pointer \rangle$:

(1) Ai is the attribute of the indexed data, $i \in [1, n]$ and n are the number of all attributes.
(2) $Ptype$ is the pointer type $Ptype \in \{B+tree, Inver\_index\}$ to the layer 2 structure.
(3) $Pointer$ is the pointer to the root node or inverted header of layer 2 B+tree, that is, according to different types of attributes, the pointer points to different index structures.

The second layer is the inverted table or B+tree index structure of the attribute values corresponding to the first layer attributes. The index structure of the inverted table contains four parts of information $\langle AiVj, Doc, F, Loc \rangle$:

(1) $AiVj$ is the j attribute value of the i attribute, $i \in [1, n_1]$, $j \in [1, m]$, n_1 are the number of character attributes, and m is the number of attribute values contained in the i attribute.
(2) Doc is the file number of the attribute value, and each file number is unique.
(3) F is the frequency of attribute values appearing in the file.
(4) Loc is the file location information of the attribute value.

The non leaf nodes of the B+tree index structure in the second layer only contain attribute values, and the leaf nodes are all ordered, and each leaf node contains three parts of information $\langle A_R V_S, Doc, F, Loc \rangle$:

(1) $A_R V_S$ is the S attribute value of the R attribute, $R \in [1, n_2]$, $S \in [1, p]$, n_2 are the number of numerical attributes in the file, and p is the number of the R attribute.
(2) *Doc* is the file number of the attribute value, and each file number is unique.
(3) *Loc* is the location information of the file where the attribute value is located.

Index Structure Creation Process

The creation process of hybrid index structure is as follows:

(1) If the file to be indexed is a new attribute of the file, a new node is created for the attribute in the first layer of the hybrid index structure.
(2) The index structure of the second layer is determined according to the attribute value type of the attribute. If it is a numeric attribute value, a B+tree index is established for the attribute value; If it is a character type property value, an inverted index is established for the property value.
(3) Repeat step (1). If the current attribute is the same as the previously indexed attribute, no new attribute node is added in the first layer of the hybrid index, and only the attribute value of the attribute is added to the corresponding index structure of the second layer.
(4) If the current attribute is different from the previously indexed attribute, a new attribute node is added in the first layer of the hybrid index structure, and then according to step (2) to determine what type of index structure needs to be established for the attribute value. Repeat the above steps until all data are indexed.

3.2 Fuzziness of Key Words Based on Query Expansion

At present, there are mainly two methods for query expansion, as shown in Table 1.

Table 1. Comparison of query expansion methods

Aspect	Middleware method of external database	Built in index optimization method
Query performance	Speed is generally better than the current system	Slightly slower than the current system
Space consumption	At least twice the extra space (full data copy+index structure)	The extra space added can be ignored (index structure of specific fields)
Write performance	Write performance is unchanged because internal and external data is asynchronous	Write performance will decline because it takes time to update the secondary index

(*continued*)

Table 1. (*continued*)

Aspect	Middleware method of external database	Built in index optimization method
Scalability	It is easy to extend and has good external operability	The internal expansion is relatively complex and needs to modify the blockchain structure, which is not friendly to the old system
Consistency maintenance and fault tolerance	Because of asynchronous operation, consistency is difficult to guarantee; System crash is prone to inconsistency	Compared with native system, consistency maintenance only needs to increase the time of building secondary index

The principles of the above retrieval methods are generally through the accurate matching of keywords to achieve retrieval, that is, the user query keywords need to be completely consistent with the server-side index keywords to retrieve the corresponding documents [7]. In order to improve the flexibility and usability of search, some scholars further proposed the keyword based fuzzy retrieval technology. These fuzzy retrieval schemes are mainly based on the editing distance to measure the similarity of keywords, which can solve the problems of user input errors and inconsistent keyword formats, such as users want to input the keyword "million" but mistakenly input "million", or input the keyword "data mining" as "data mining". To sum up, the existing retrieval schemes do not consider some words related to keyword semantics. When the retrieval user does not know much about the domain to be retrieved, the keywords submitted by the user are difficult to express the user's actual retrieval intention, and some documents containing semantic related words are missed, resulting in incomplete retrieval results and reducing the user's satisfaction with the retrieval results [8].

For example, if a user wants to search for articles related to football, but he is not familiar with this field, he can only submit the keyword "football" for retrieval, but in the following case, it will cause the problem of missing.

Article 1 is about the content of football transfer market, the key words extracted are "transfer, value, make up".

Article 2 is about the football game situation, the key words extracted are "score, referee, foul".

These two articles are related to football, which is also what users want to know. However, the keywords extracted from these documents do not contain the keyword "football", and these documents will be missed in the retrieval process.

In view of the above shortcomings, this paper realizes the fuzziness of search words based on query expansion, so as to improve the comprehensiveness of retrieval. That is to say, according to the co-occurrence probability of keywords in the document, the semantic relationship database of keywords is constructed. In the retrieval process, the cloud service provider extends the semantic of query keywords submitted by users, Then, the retrieval is performed according to the expanded keyword set, and finally the retrieval results are sorted according to the comprehensive relevance value of the

retrieved documents, and the sorted retrieval results are returned according to the user's requirements (if the user requires to return the first k, only the first k documents are returned) [9]. The basic process of keyword fuzziness based on query expansion is shown in Fig. 3.

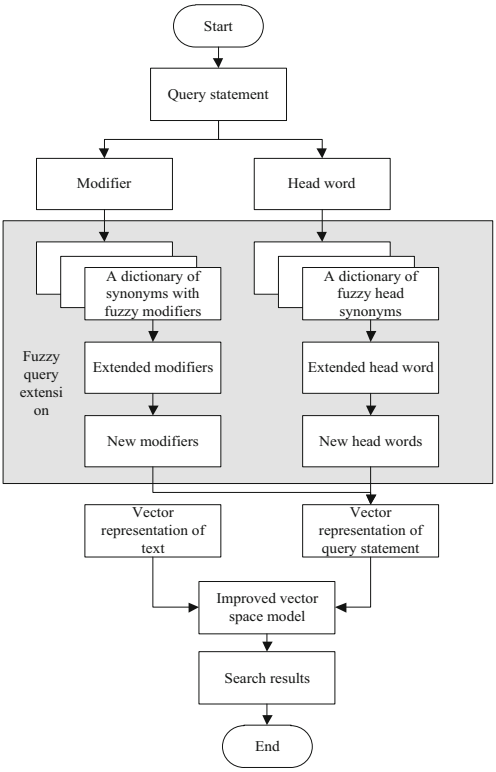

Fig. 3. Basic process of keyword fuzziness based on query expansion

The specific description is based on the query statement given by the user. Usually it contains the compound phrase with modifier modifier, which is marked by the word character and synonym is used.

In dictionary, we can get the synonyms of modifier and central word respectively. By re combining these extended modifier and central word in statistical sense, we can get new synthetic phrases. The paper constructs vector space model by using the phrase, calculates the correlation between text vector and query vector, sorts the text according to the value of correlation degree, and finally, gets the text related to the semantic of query statement.

3.3 Implementation of Fuzzy Retrieval Based on Query Expansion

Boolean retrieval model is the most traditional and mature retrieval model, which has a wide range of applications in the field of information retrieval. Fuzzy retrieval model is an improved product based on Boolean retrieval model and fuzzy set theory. It defines the fuzzy relationship between the query statement and the related literature. The fuzzy retrieval model relates the text information and the query data to a certain extent, and assumes that there is a set of fuzzy text information related to each word in the query statement. In other words, each query word is defined as a fuzzy set, and the elements in the set are the text information for retrieval. Each independent text in the retrieval text information set has a membership degree for each query word in each query statement.

Fuzzy Retrieval Based on Word Incidence Matrix

(1) Constructing keyword matrix

The word incidence matrix is a word matrix composed of the semantic relationship values between keywords and query words extracted from relevant text information, that is, the elements in the set composed of keywords and query words are taken as rows and columns. Assuming that the word incidence matrix is represented by $W_{k\times k}$, k represents the number of elements in the set. The element w_{ij} in the matrix corresponds to the semantic relationship values between words i and j, It represents the semantic similarity between two words. In order to make its value range within the interval $[0, 1]$, formula (1) is used for calculation:

$$w_{ij} = \begin{cases} \frac{N_{ij}}{N_i + N_j - N_{ij}}, & i \neq j \\ 1, & i = j, \end{cases} \tag{1}$$

Among them, N_{ij} represents the number of documents containing both i and j, N_i and N represent the number of documents containing i and j respectively. When w_{ij} is 0, it means that there is almost no semantic correlation between the two words; When w_{ij} is 1, it means that there is the strongest correlation between the two words.

(2) Using keyword matrix to calculate membership degree

For fuzzy retrieval model, each query word corresponds to a set containing fuzzy text information. Let D_i denote the fuzzy text information set related to word V, then for any independent text information d_j, its membership degree R_{ij} belonging to set D_i is calculated by formula (2):

$$R_{ij} = \underset{k\in d_j}{\oplus} W_{ik} \tag{2}$$

Among them, W_{ik} represents the semantic relation value of word i and word j, k is the word in independent text d_j, and $\oplus$ represents fuzzy operator:

$$\underset{i}{\oplus} X_i = 1 - \min_{\mathrm{i}}(1 - \mathrm{X}_i) \tag{3}$$

Then the membership degree R_{ij} of any independent text information d_j, which belongs to set D_i, can be reduced to:

$$R_{ij} = 1 - \min_{k \in d_j}(1 - W_{ik}) \tag{4}$$

(3) Convert user query

Usually, a query statement containing multiple query words is input to the computer during retrieval. The traditional Boolean retrieval replaces the relationship between these query words with Boolean logic expression. The fuzzy retrieval model uses "truth table method" to transform the query statement into a main disjunctive paradigm composed of minimal items, Let $Q(h)^-$ denote the set of negative query words and $Q(h)^+$ denote the set of positive query words in the query statement, then the membership degree $T_i(h)$ of any independent text information d_i belonging to the whole query statement is:

$$T_{\mathrm{i}}(\mathrm{h}) = 1 - \left(\prod_{j \in Q(h)+} R_{ij}\right)\left(\prod_{j \in Q(h)-} R_{ij}\right) = 1 - \left(\prod_{j \in Q(h)+} R_{ij}\right)\left\{\prod_{j \in Q(h)+} (1 - R_{ij})\right\} \tag{5}$$

where R_{ij} is the membership of independent text information d_i to the corresponding query word:

Word Vector

Word vector technology is a kind of word representation method through distributed expression of words in the language model. In the training process, words are transformed into dense vectors, and the distance between the corresponding vectors is relatively close for similar words. Therefore, word vector can be used to calculate the similarity between words. Cbow model is a training word vector model proposed by mikolov in 2013. The network structure of cbow model consists of three layers: input layer, projection layer and output layer. The input layer is one hot coding corresponding to the input context. The projection layer sums up the input initial vectors, and the output layer corresponds to a tree structure, which is a tree structure constructed with the words above and below as leaf nodes, and the number of times each word appears in the corpus information as weight. The core idea of cbow model is to predict the word w_t when the context w_{t-2}, w_{t-1}, w_{t+1} and w_{t+2} of the current word wt are known.

Query Extension

As we all know, in natural language, a word may express several meanings, and the same several different words may express the same meaning. When searching, the computer may not return documents with the same meaning as the query words but different words. Query expansion is to improve query efficiency by adding words with similar semantics to query words [10]. In this paper, we use the word vector trained by CBOW model to calculate the similar words of the query term and expand the query term.

(1) Word segmentation and part of speech tagging are carried out for the input query sentences. In order to avoid the interference of useless information, the stop words are removed from the sentence. The adjectives, adverbs and the query core words modified by them are retained to form the keyword set of the query item.

(2) The word vector of Related words in the keyword set is extracted. The cosine distance of space vector is used to calculate the similarity between words.

(3) Take the first N words with the largest similarity with the corresponding words as the extension.

Based on the above research, this chapter studies the implementation of fuzzy retrieval based on query expansion, that is, the retrieval is completed by calculating the corresponding similarity between keywords and index words [11, 12]. The specific process is as follows:

Step 1: build a large-scale unlabeled data set, and preprocess it by word segmentation, part of speech tagging and filtering stop words. On this basis, train the processed data set with word vector model (CBOW model) to get the corresponding word vector;

Step 2: construct the text information data set as the retrieval content, and extract 20 keywords from each document (these keywords can generally summarize the content of the article) after the preprocessing operations of word segmentation [13], part of speech tagging and filtering stop words;

Step 3: preprocess the query statement, calculate the similarity according to the cosine distance, expand each word in the query item by N similar words, and transform the query statement into the main disjunctive normal form of the minimal item;

Step 4: use the extracted keywords and the expanded query words to form the word incidence matrix, get the relationship value between words, go through the whole data set, and calculate the membership of each article belonging to the query sentence;

Step 5: output the corresponding search results after sorting.

4 Simulation Experiment

4.1 Experimental Environment

This paper uses Java and go language; The experimental results are stored in the file, and the data is drawn into a chart by MATLAB. All the experimental procedures are implemented under the operating system of Ubuntu 16.04 (64bit). The processor platform is Intel (Xeon es-2630 v4.2 GHz * 40), and the memory size is 250 GB. However, according to the characteristics of memory configuration of Java virtual machine and the system overhead, some key parameters of Java virtual machine are configured as follows: the minimum memory size of -Xms2048m is set to 2G, and the maximum memory size of-Xmx64000 is set to 64G.

4.2 Experimental Data

In order to ensure the reliability of the experiment, the 3.2G Wikipedia Chinese corpus, 17901 articles and news corpora including politics, economy, culture, medicine, history and other aspects crawled on the Internet are used as the corpus training set of CBOW

model. 150 articles, totally 1350 articles, are selected from 9 fields of politics, military affairs, economy, culture, medicine, history, sports and so on, retrieve as a text data set.

4.3 Evaluation Criterion

Generally, precision and recall are used to evaluate the performance of information retrieval system

(1) Precision ratio refers to the percentage of the number of relevant documents in the total number of documents, which reflects the retrieval accuracy, and its complement is the false detection rate;

$$\text{Precision ratio} : q = \frac{\alpha}{\beta} \tag{6}$$

where q is the precision ratio; α is the amount of relevant information retrieved; β is the total amount of information retrieved.

(2) Recall rate refers to the percentage of the number of relevant documents detected in the system, which reflects the comprehensiveness of retrieval, and its complement is the missing rate;

$$\text{Recall rate} : p = \frac{\chi}{\gamma} \tag{7}$$

Where p is the recall rate; χ is the amount of relevant information retrieved; γ is the total amount of relevant information in the system.

It can be seen from formula (6) and formula (7) that recall and precision are mutually exclusive. When the recall rate is increased, the precision rate will decrease accordingly; When the recall is reduced, the precision will increase accordingly. In order to judge the retrieval efficiency more accurately, the harmonic average of recall and precision is added here.

$$H = \frac{2 \times q \times p}{q + p} \tag{8}$$

In order to get the experimental data and calculate the relevant proportion, 20 experimenters were asked to select the search results respectively, and the relevant documents of each query were obtained by synthesizing the opinions of 20 people.

4.4 Word Vector Training

In this paper, the CBOW model is used to train. The dimension of the word vector is set to 100, the size of the context window is set to 5, the words whose frequency is greater than 1 are retained, and 0.025 is taken as the learning rate of the model. The corresponding word vector is obtained by 100 times of iterative training. Table 2 is the sample of the word vector training results.

Table 2. Sampling of word vector training results

Input	Similar words	Vector cosine distance
Happiness	Happy	0.794 743 53
	Happy	0.777 624 21
	Fine	0.761 178 97
	Unforgettable	0.728 432 14
	Pleasure	0.698 920 33
China	Our country	0.852 053 22
	U.S.A	0.774 880 76
	The republic of korea	0.749 371 11
	Britain	0.734 025 57
	Japan	0.728 140 57
Sugar urine sugar	Blood sugar	0.691 834 22
	Pancreatin	0.659 388 69
	Hyperlipidemia	0.598 150 42
	Heart disease	0.532 443 88
	Hypertension	0_512 551 30

It can be seen from Table 2 that by calculating the cosine distance between word vectors, words with similar semantic environment can be found more accurately.

4.5 Comprehensive Comparative Analysis

In the data set of 1350 articles in 9 different fields, the recall rate, precision rate and their harmonic average value are taken as the evaluation indexes, and the threshold value is taken as 1. Fuzzy retrieval based on word association matrix and word vector query expansion are used respectively, and the number of expansion items is added to carry out the retrieval experiment. The experimental results are shown in Table 3.

Table 3. Comparison results of comprehensive experiments

Project	Word incidence matrix	Word vector extension
q	0.2168	0.2817
p	0.5971	0.6122
Harmonic average	0.3181	0.3859

It can be seen from Table 3 that the evaluation result of the improved fuzzy retrieval model by query expansion is better than that of the fuzzy retrieval model based on word incidence matrix, and the recall, precision and the harmonic average of the two are increased by 1.51%, 6.49% and 6.78% respectively, indicating that the query expansion based on word vector can improve the query effect and overcome the problem of "word mismatch" to a certain extent.

4.6 Analysis of Query Expansion Number N

In this paper, the key step is to use the word vector to calculate the similar words of the query term, and use the similar words to expand the query term. How to get the most appropriate value of the query expansion number n to ensure the optimal retrieval efficiency is $N = 0, 2, 4, 5, 6, 8, 10, 12, 14, 15, 16, 18, 20, 22, 24, 26, 28, 30$ problem worthy of discussion, the experimental results are shown in Table 4.

Table 4. Analysis of expansion number N

Number of extensions N	Harmonic average
0	0.32
2	0.34
3	0.37
6	0.40
7	0.41
10	0.42
12	0.42
14	0.44
16	0.45
18	0.46
20	0.47
22	0.46
24	0.46
26	0.45
28	0.45
30	0.45

It can be seen from Table 4 that with the change of parameter N, the harmonic average of recall and precision also changes $N \in [0, 15]$; When the query expansion number $N \in [15, 20]$, the relative harmonic average value gradually reaches the maximum value; When $N > 20$, the harmonic average is relatively lower, but basically remains stable. This is because with the increasing of N, the number of extensions is increasing, but the similarity between the extensions and the corresponding query items is beginning to decrease. Therefore, it is better to take $N \in [15, 20]$ as the number of the corresponding extensions.

5 Concluding Remarks

Because blockchain is designed for decentralized digital currency at the beginning, its data organization form is solidified, resulting in a single retrieval method and lack of effective data privacy protection design, which impedes the implementation of blockchain in relevant fields with strong data security requirements and data query

requirements. For this reason, this paper studies the safe storage and safe and efficient retrieval of sensitive data on blockchain, and proposes a fuzzy retrieval method of blockchain information based on B+tree index structure. The experiment shows that, compared with the traditional retrieval model, the retrieval model studied in this paper improves the fuzzy retrieval effect, can fully solve the problems existing in the traditional methods, promote the development of multiple fields represented by medical big data, and has a certain contribution to the further development of the information technology field. This experiment is a preliminary exploration of this method, and many links need to be further improved. In the future, we will try to take different thresholds for text retrieval on the basis of this paper, expand the training corpus, improve the extraction method of keywords, and improve the retrieval efficiency.

Fund Projects. 1. Key scientific research platforms and projects of ordinary universities in Guangdong Province in 2020-Research on cross-border E-commerce blockchain information security technology based on B+search tree algorithm (Item No.:2020ZDZX3103)

2. "Innovation Research on Cross-border E-commerce Shopping Guide Platform Based on Big Data and AI Technology", Funded by Ministry of Education Humanities and Social Sciences Research and Planning Fund (18YJAZH042).

References

1. Vale, T., Almeida, E.D.: Experimenting with information retrieval methods in the recovery of feature-code SPL traces. Empir. Softw. Eng. **24**(3), 1328–1368 (2019)
2. Marcos-Pablos, S., García-Peñalvo, F.: Information retrieval methodology for aiding scientific database search. Soft Comput. **24**(8), 5551–5560 (2018). https://doi.org/10.1007/s00500-018-3568-0
3. Kaur, P., Pannu, H.S., Malhi, A.K.: Comparative analysis on cross-modal information retrieval: a review. Comput. Sci. Rev. **39**(2), 100336 (2021)
4. Alexandrescu, A.: Optimization and security in information retrieval, extraction, processing, and presentation on a cloud platform. Inf. (Switzerland) **10**(6), 200 (2019)
5. Vaz, G.J., Barbedo, J.: An information retrieval system based on multiple portlets: communication between its components. Int. J. Web Portals **13**(1), 74–86 (2021)
6. Huang, T., Weng, Z., Liu, G., et al.: HD-Tree: an efficient high-dimensional virtual index structure using a half decomposition strategy. Algorithms **13**(12), 338 (2020)
7. Banerjee, P.S., Chakraborty, B., Tripathi, D., Gupta, H., Kumar, S.S.: A information retrieval based on question and answering and NER for unstructured information without using SQL. Wirel. Pers. Commun. **108**(3), 1909–1931 (2019). https://doi.org/10.1007/s11277-019-06501-z
8. Yang, Z., Yu, H., Tang, J., et al.: Toward keyword extraction in constrained information retrieval in vehicle social network. IEEE Trans. Veh. Technol. **68**(5), 4285–4294 (2019)
9. Surya, S., Sumitra, P.: An innovative information retrieval model implementing particle swarm optimization technique. J. Comput. Theor. Nanosci. **17**(12), 5613–5617 (2020)
10. Parali, U., Zontul, M., Ertugrul, D.C.: Information retrieval using the reduced row Echelon form of a term-document matrix. J. Internet Technol. **20**(4), 1037–1046 (2019)
11. Huang, M.X., Lu, S.D., Xu, H.: Cross-language information retrieval based on weighted association patterns and rule consequent expansion. Data Anal. Knowl. Discov. **3**(9), 77–87 (2019)

12. Xie, X.Z., Wang, Z.K., Li, Y.X., et al.: Cross-domain semantic information retrieval based on convolutional neural network. Comput. Appl. Softw. **35**(8), 73–78 (2018)
13. Shen, X.T., Lv, L.H.: Research on information retrieval location based on cloud computing. Comput. Knowl. Technol. **14**(08), 210–211 (2018)