Privacy Preserving Multi Keyword Search over Outsourced Cloud Data Using N-Gram

V.Krishna Reddy, Ch.Srikala, Nishmai chowdary V, G. Sai Raghava, Roshan baig

Abstract*— With the increased use of cloud computing, cloud information owners are exasperated to reproduce the perplexing information management systems from local servers to the infinite open cloud for business purposes and reserved funds. In any scenario to make sure the information is secure and protected, all the sensitive data such as medical records, tax files must be cryptic before outsourcing it to the cloud. So, we can no longer depend on the customary plain keyword search to get the required information. By supporting the outsourced cloud information has a central future significance. When realizing the numerous numbers of files and records outsourced into the cloud by the information proprietors, it is a necessity to have a vast allowance of keywords so that any user in pursuit of a file can request them by their significant keywords. Most of the encrypted searches revolve around a single search or Boolean search and doesn't provide an effective search results. We here tackle this situation and use the multi-keyword ranked seeks over the scrambled data. We make use of the best encryption standards and other prerequisites to provide proper security. For the multi keyword search we make use of the 'cosine search similarity' and "coordinate matching". We initially discuss about an essential thought for the MRSE dependent on inward item calculation, and after that give fundamentally hybrid MRSE plans to carry out rigorous necessities in a diverse risk models. To enhance seek understanding of the information look benefit, we additionally stretch out these two plans to help more inquiry semantics with N-gram Features

Keywords: Cloud Computing, privacy, encryption, multiple keywords and ranked search technique.

I. INTRODUCTION

Distributed cloud computing will be processing utility where clients can temporarily store the requisite information into the cloud in order to appreciate a on- request top-notch application and administrations from a shared pool of figuring assets. It is a sort of registering that depends on redistributing and processing resources rather than having neighbourhood servers or individual gadgets to deal with applications. In basic terms, distributed computing implies accessing information that has been kept away and projects over the web rather than a PC's hard drive. Every day the importance of cloud computing is being realized globally and has been accepting the new developments in the cloud.

It is a budget friendly, adaptable as well as demonstrated conveyance stage giving business or purchaser IT benefits from the web. Anyways, distributed computing has an additional hazard as the fundamental administrations are frequently re-appropriated to an outsider, who guarantees secured and protected information in the cloud, bolster information and administration accessibility. Its extraordinary adaptability and financial reserve funds are inspiring two people and undertakings to re-appropriate their nearby mind boggling information administration framework.

To secure protection of information and contradict spontaneous retrieval from the cloud, sensitive information, like, personal messages, health records, individual photograph collections, tax and salary, information, et cetera, must always be converted into cryptic form by information proprietors before re- appropriating to the business open cloud; this, in any case, obsoletes the customary information use benefit dependent on plaintext catchphrase. The irregular deal of downloading every file in the file catalog and converting into ciphered text locally is unmistakably unrealistic, because of the substantial measure of transmission capacity cost. Additionally, besides taking out the nearby capacity administration, putting away information into the re-appropriated capacity doesn't fill any need except if they can be effectively looked and used.

Positioned seek can likewise exquisitely wipe out inane management activity by just giving back the most matched information to the keyword, which is profoundly easy. Consequently, investigating protection safeguarding and a viable hunt benefit over scrambled information is of extraordinary significance. To improve the query output, we require productive techniques to execute the likeness seek over a substantial measure of scrambled information. LSH (region delicate hashing) is broadly utilized for quick similitude seeks on plain information in the data recovery network. In our plan, we plan to use it with regards to the scrambled information.

Revised Version Manuscript Received on March 08, 2019.

Dr. V.Krishna Reddy, CSE Department, Koneru Lakshmaiah Education Foundation, Vaddeswaram, Guntur, Andhra Pradesh, India (e-mail: vkrishnareddy@kluniversity.in)

Ch.Srikala, CSE Department, Koneru Lakshmaiah Education Foundation, Vaddeswaram, Guntur, Andhra Pradesh, India (e-mail: srikalachallagundla@gmail.com,)

Dr. V.Krishna Reddy, CSE Department, Koneru Lakshmaiah Education Foundation, Vaddeswaram, Guntur, Andhra Pradesh, India (e-mail: nishmaichowdary9@gmail.com)

Dr. V.Krishna Reddy, CSE Department, Koneru Lakshmaiah Education Foundation, Vaddeswaram, Guntur, Andhra Pradesh, India (e-mail: gsairaghava1998@gmail.com)

Ch.Srikala, CSE Department, Koneru Lakshmaiah Education Foundation, Vaddeswaram, Guntur, Andhra Pradesh, India (e-mail: roshanbaig017@gmail.com)

N-gram

N-Gram is a word expectation calculation utilizing probabilistic strategies to foresee the next prevail word in the wake of watching N-1 words. Along these lines, processing the likelihood of the following word is firmly identified with registering the likelihood of a succession of words

Simple N-gram (Unsmoothed)

The most basic and uncompelled probabilistic structure for guessing the words, it is like finding compatibility to each word. Let's assume that there are N words in a particular data set, for any word the likelihood of it being followed after another word would be 1/N. Nonetheless, this methodology disregards the way that a few words are more incessant than the other in dialects.

An upgrade to this structure is a model which is to simply appoint that the likelihood for a word 'Ai' being followed after the word 'Ai-1' is the probability of the word 'Ai'. For instance, a word "that" happens at a recurrence of 1/10.000. At that point, for any word, the likelihood of the following word being "that" is 7%. Nonetheless, this overlooks in a few settings, an event of the "fox" after a word is substantially more plausible than the event of "that". For example, "fox" following "arctic" appears to be considerable more legitimate than "the" accompanying "arctic"

Markov Assumption

The idea mentioned before validates that some specific words are most likely to pursue a word in specific settings. It is required to realize the exact same word till the word that we are endeavoring to foresee, however it is wasteful to acknowledge the whole history, since we can experience limitlessly several succession of sentences and the part of the past that we know would have never existed. In this way we scramble the history with a couple of words. Bi-gram, likewise known as Markov supposition, expect that we can predict the next words by considering the last word. We can also sum up bi-gram(looking at the last two words previously) to trigram, and to N-gram (looking at the N-1 words before). Along these lines, the general condition for the restrictive likelihood of the following word in a grouping would be:

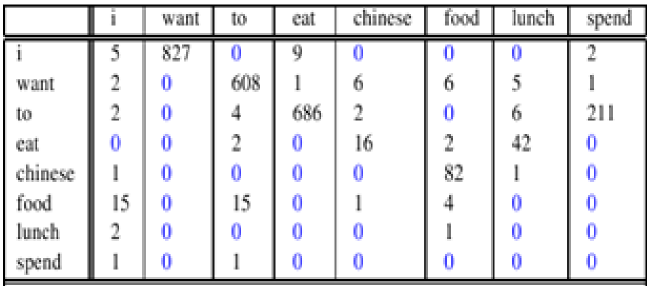
$P(w_n| w1_{n-1}) \approx P(w_n | w_{n-N+1}{}^{n-1})$,where word grouping w_1, w_2, ... , w_{n-1} is spoken to as $w1_{n-1}$.The most straightforward approach to appraise the probabilities is to utilize MLE that is the Maximum Likelihood Estimation.

$$P(w_n|w_{n-1}) = \frac{C(w_{n-1}w_n)}{C(w_{n-1})}$$

In light of taking tallies from the corpus and normalizing them to lie in the interim [0,1].

$$P(w_n|w_{n-N+1}^{n-1}) = \frac{C(w_{n-N+1}^{n-1}w_n)}{C(w_{n-N+1}^{n-1})}$$

The table shows an example for the number of Bi-gram counts:

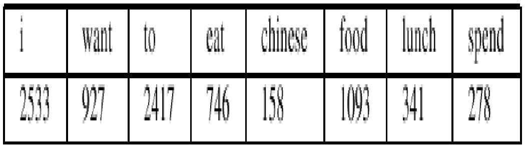


	i	want	to	eat	chinese	food	lunch	spend
i	5	827	0	9	0	0	0	2
want	2	0	608	1	6	6	5	1
to	2	0	4	686	2	0	6	211
eat	0	0	2	0	16	2	42	0
chinese	1	0	0	0	0	82	1	0
food	15	0	15	0	1	4	0	0
lunch	2	0	0	0	0	1	0	0
spend	1	0	1	0	0	0	0	0

Figure 4.1 Bigram counts for eight of the words (out of $V = 1446$) in the Berkeley Restaurant Project corpus of 9332 sentences.

Count of the occurrence of the words in corpora are;

Rather than altering both the numerator and the denominator, it is more flexible to introduce a firsthand



i	want	to	eat	chinese	food	lunch	spend
2533	927	2417	746	158	1093	341	278

The numbers of words occurring in corpora are: Two things can be watched from the above table;

- The bi-gram ("i", "want") happened multiple times in the corpora. So as indicated by the recipe, the likelihood of "I" followed by "need" is 827/2533.
- The lattice is meager, which indicates that the lattice has a great deal of zeros in it, even the words chose are rational with one another.

Presently we can discover the likelihood of a sentence with the use of bi-grams;

P (<s> I need English sustenance </s>) = P (I | <s>). P (want | I). P (English | need). P (food | English). P (</s> | nourishment)

Note: <s> signifies beginning of a sentence and </s> indicates end of a sentence.Smoothing

From the table above it is well observed that the majority are occupied by zeroes. Since our corpora is constrained, most word groupings are doled out to zero likelihood even they ought to have a non-zero likelihood. The MLE estimation gives precise outcomes when the arrangements are visit in our preparation information, however it doesn't give great outcomes in zero likelihood groupings and the successions with low recurrence. Another issue is perplexity, a metric utilized for assessment of N-grams, does not work when there are zero likelihood arrangements in our information. Thusly, we will adjust MLE to gather some likelihood mass from high recurrence groupings, and disperse it to zero recurrence successions. This adjustment is called Smoothing

Add-One Smoothing

In add-one smoothing one is added to the numerator and denominator, and then the probability is calculated. They include one smoothing is an extremely straightforward smoothing calculation, that augments all frequencies by one. Anyways, rather than altering both the numerator as well as denominator, it is more flexible to define a new count

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

variable, to find the probability by dividing with N in the usual way. This technique is one of the most trifling technique among all.

Typically, the likelihood of the event of a term is given underneath, where ci represents the occurrence of the word, while N represents the aggregate number of terms in an expression;count variable, by allotting N in a habitual way the probability of the term can be found.

When applied to bi-gram the form of the equation of the count is as follows:

Again we should partition the add-one tally to C(wn-1), the quantity of events of the final term, to discover the likelihood P*(wn | wn-1), which means the likelihood of wn succeeding the term wn-1. The issue includes one marking down in the majority of the odds are reduced unreasonably, in some cases to a factor of 10.

Good-Turing Discounting

This technique utilises the frequency of the number of count of occurrences of N-Grams to calculate the maximum likelihood estimation. The inspiration driving By observing the tally once that we have never seen before, we try to reduce the good turning gauge, which is called singletons. The Great Turing generally utilizes the recurrence of the singletons to assess the likelihood mass that will be doled out to zero-check bi-grams. With the end goal to process the recurrence of singletons, we have to tally Nc which is the check of all N-grams that happen c count of times. Along these lines, N0 compares the count of bi-grams that are not observed, N1 relates to the count of singletons et cetera.

The MLE means the Nc is c. In Good-Turing gauge, a new check is a component Nc+1:

The equation mentioned before expect that we know N0, that is the quantity of bi-grams we have never observed. As a matter of fact, we can figure it by the accompanying way. Assume that the vocabulary is of size V, at that point the quantity o every conceivable bi-gram with this vocabulary is V2. We know what number of bi-grams we have seen, so V2-(number of bi-grams known) is the N0.

When the count of the words is incremented by 1, then the equation must be equalized accordingly. To normalize the equation, as we incremented some V number of distinct words and Vocabulary, then we must add it to the denominator.

By and by, the Good-Turing tally isn't utilized for all tallies in light of he fact that more often than not, regularly observed words gives dependable probabilities. In this manner, a limit k is characterized and c* is utilized just for the c's that are littler than k. At that point, the right condition with the k esteem progresses toward becoming;

This acquaintance of k additionally implies with treat tallies lower than k like zero tally bi-grams. Be that as it may, Good-Turing isn't utilized exclusively in N-gramusage, it is by and large utilized with back-off and insertion calculations.

EXPERIMENTAL DETAILS

2.1. Problem Statement:

The pliability and lucrative budgets of deployed memory are inspiring both users and owners of the cloud to contract out their local sophisticated data administration system. To preserve data seclusion and overcome gratuitous access in the cloud and further, protective information and records must be ciphered by cloud possessors before uploading data into the cloud.

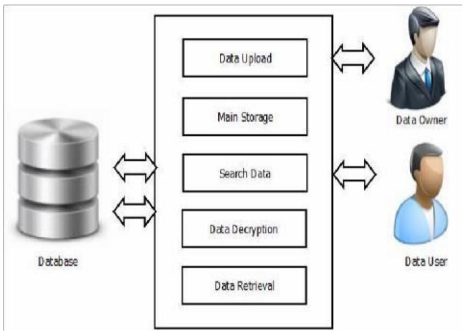


Figure 1 Construction Diagram

Projected Work

To achieve competent resemblance pursuit, cloud owner constructs a private directory and subcontract it to the cloud server along with enciphered data items. Server operates search on the catalog accordingly with respect to the enquiries of the data cloud handlers without any rough knowledge of the data other than the data cloud owner consents an antagonist to acquire.

Data Proprietor's Segments:

Login Segment
Data Transmit
View Documents
File Segment

Login Segment:

Information Proprietor has to login to the cloud with the end motivation to handover the information on to the cloud.

Data Transmit:

This segment is solely for the purpose of syncing the data on to the server. Cloud proprietor has the ability to randomly pick out the data file from the indigenous machine and outsource it to the server.

View Documents:

If the cloud proprietor desires to pursuit any explicit folder from the cloud, this segments the finest preference. The return from pursuit task gives the rundown for the documents; these segments likewise obligate the choice for cancellation in a particular record.

File Segment:

This segment gives an alternative to document allotment. The segment, in one way or another relies upon the View document segment. Subsequently after the search process of the files, the data cloud user can handpick specific file or files

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

and send request to the owner to access the files.

Data Consumes Segment:
Registration
Analysing Documents
Downloading required Documents

Registration:

With the end goal to be a piece of area, the client needs to enrol on the entryway. With the end goal to enrol on the entryway, client needs to give some peculiar information that will be put away at back-end. When the time comes for the user to login using the credentials, this information further utilized to substantiate the user.

Analysing Documents:

It is a main and central segment for the consumer. In this segment client have the choice to seek particular records (wanted).Client can search for a specific file by using appropriate keywords in the search bar, if there is a match and a folder exists for the specific keyword specified, the document label will appear on the particular webpage. On account of unfitting keywords inclusion aimed at hunt, the resulting yield will be invalid.

Downloading required Files:

This segments a follow up module to the Search module. Subsequent to seeking records if a few documents lists are returned, If the user is satisfied with the documents turned up, then, there is a choice for downloading the records. Client can choose particular record and download it by getting an authenticated private key (it sent to the user by the information proprietor)

REVIEW OF THE RESULTS

Proposed Algorithm

In the paper we used following algorithm:

Amid File Uploading following Steps will execute.

1. After uploading the files into the cloud, the data in the file will be split word by word (depending on the spacing).
2. Disposing of the Stop words (e.g.:- and, or, an etc.)
3. Implements Stemming. (e.g.: -If there are two words print, printed. By stemming ‘ed’ is detached from the succeeding word.) These consecutive three phases are called pre-processing phases.
4. Reckon term frequency, the term frequency-inverse folder frequency. The values calculated for every keyword in the file are stored in the database.
5. For every five files uploaded consecutively, the following steps are performed.
6. For the next set of five files uploaded, additional steps will be performed along with the existing ones.
7. An additional document vector is created with array values of TF, TF-IDF of all the keywords of the uploading files.
8. In the same way another document vector is created for the other uploaded files also.
9. After that cosine similarity is performed on the uploading documents along with all other documents in the database. So the cosine searches similarity for the vectors P = (p1, p2) and Q = (q1, q2) will be:

(p1 q1 + p2 q2) / sqrt(p1^2 + p2^2) sqrt(q1^2 + q2^2) Where p1, p2, q1, q2 are term frequency-catalog term frequency values.

For example, If a 6th file is being uploaded cosine search similarity will be performed with 5 values to find the similarity.

The files are then divided into clusters based on the average of the 5 cosine similarity values. The cosine similarities that are greater than the average are formed into one cluster.

The highest frequency keywords maximum of 10 are chosen from each documents and stored in the catalog.

Catalog Creation

For every set cluster, constructs
an individual catalog.

1. Search and find files in that cluster.
2. From the catalog of that cluster retrieve highest frequency keywords of all files. 3.
3. Find most repeated keywords keeping the count to the maximum of 4
   Example: CDAF
4. For every top 10 high frequency keywords in a particular cluster find the values of TF-IDF.
   E.g.:-TF-IDF*1000 ❼ all high frequency keywords
   (people) ❼ zi ❼ z1*z2*z3*....z10
   ❼ 13467

Perform concatenation of CDAF1367

Performing search using N-Gram:

During the search process these steps are performed

1. The user performs search operation by giving an input of some keywords in the text bar.
2. Based on the search the nearest or similarly matched keywords are retrieved.
3. Find which cluster these keywords belong to and retrieve its respective catalog.
4. There are two parts in the catalog.
5. Example: searches for → hello → 123, world → 1452, 13467 % 123 == 0 && 13467 % 145 == 0
6. Based on the catalog, character analysis is performed.
7. Selects the cluster with more character similarity to the searched keyword.
8. The catalogs that are identical to the TF, TF-TDF values are selected.
9. Identify watchwords that are associated to catalog and identical to the search keywords.

The catalog related files are displayed.

B. Ranking

Example: - ball, tree, ground and grass-are the searched keywords.

TF-IDF ARE:

BALL-10, TREE-8, GROUND-11, GRASS-4

Docx2 contains ball and tree->10+8=18 Docx1 contains tree, ground → 8 + 11 = 19 Docx3 contains grass → 4. Order of files will be displayed as, Doc1, Doc2, Doc3

CONCLUSIONS

In our paper, we have proposed a proficient similarity ranked searchable encryption scheme using N-Gram. With the help of locality, sensitive hashing technique, which is also utilized for an effective and fast similar search in highly complex spaces for decrypted data? We also took advantage of the bloom filter catalog for enabling fast search with respect to the encrypted data. For such specific scenarios, it gets extremely difficult not to forfeit secrecy of the delicate information stored in cloud while giving usefulness. We gave a thorough security and demonstrated the security of the projected plan in the given description to guarantee the secrecy.

REFERENCES

1. Chi Chen, Xiaojie Zhu PeisongShen, Jiankun Hu, Song Guo, ZahirTari, and Albert Y. Zomaya, “An Efficient Privacy Preserving Ranked Keyword Search Method” , IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, APRIL 2016
2. Ning Cao, Cong Wang, Ming Li, KuiRen, and Wenjing Lou, Worcester Polytechnic Institute,
3. Illinois Institute of Technology, “Privacy Preserving Multi Keyword Ranked Search Over Encrypted Cloud Data.
4. Keikoas rhizume,David.G.Rosado,Eduardo Fernández Medin and Eduardo B Fernandez, “An analysis of security issues for cloud computing”, “Journal of Internet Services and Applications” 2013
5. Cong Wang, Ning Cao, Jin Lin, KuiRen†, and Wenjing “Secured ranked Search over Encrypted Data” 2010 International Conference on Distributed Computing Systems.
6. W. K. Wong,The University of Hong Kong “Secure kNN Computation on Encrypted Databases”
7. Cong Wang, , Qian Wang, ,KuiRen, Ning Cao, and Wenjing Lou, “Toward Secure and Dependable Storage Services in Cloud Computing”,
8. T. Jothi Neela1 and N. Saravanan,” Privacy Preserving Approaches in Cloud”, May 2013
9. Yong Kim, Yoo-Kang Ji and Sun Park,” Big Text Data Clustering using Class Labels and Semantic Feature Based on Hadoop of Cloud Computing”, International Journal of Software Engineering and Its Applications
10. Muhammad Yasir S shabir, AsifI iqbal, Zahid Mahmood_, and Ata Ullah Ghafoor, ” Analysis of Classical Encryption Techniques in Cloud Computing”,February2011