April 2023

A METHOD AND SYSTEM FOR SECURE DOCUMENT SEARCH

Nicholas Kersting
VISA

Joydeep Mitra
VISA

Aman Mohanty
VISA

Miaomiao Liu
VISA

TITLE: "A METHOD AND SYSTEM FOR SECURE DOCUMENT SEARCH"

VISA

Nicholas Kersting
Joydeep Mitra
Aman Mohanty
Miaomiao Liu

TECHNICAL FIELD

[0001] This disclosure relates generally to the field of information security. More particularly, the disclosure discloses method for secure document search.

BACKGROUND

[0002] One of the existing technologies discloses an information retrieval system that converts unstructured ad-hoc search queries into search instructions that retrieve data from a structured relational database. A guided ad-hoc search engine allows most business users to express search requirements by just typing a few words. Data from the relational database, or from any other structured or unstructured data source, is uploaded into a distributed in-memory database system (database system). Tokens are automatically generated based on content, attributes, measures, and other metadata extracted from the relational database and any other structured or unstructured data sources. The tokens are identified and displayed in response to user inputs and may be any word, phrase, set of characters, symbols, etc. associated with data that exists in the database system.

[0003] Another existing technology discloses retrieving data from any one of a plurality of data sources, the data stored by each data source being arranged according to an associated data format. However, while storing the data related to the Personal Identifying Information (PII) in a database or static file location, the data must be encrypted. Further, when a user needs to retrieve the PII, the PII may be decrypted before applying usual database querying or search methods to isolate the records of interest, and this process is slow. Moreover, the decryption requirement exposes all the data and may leak more information than the application really requires and one cannot selectively retrieve records of interest without exposing all of data.

SUMMARY

[0004] According to some non-limiting embodiments, the present disclosure discloses a method for secure document search. The objective of the present disclosure is to avoid the need to decrypt the whole database while retrieving the necessary data.

[0005] The present disclosure provides homomorphic encryption which is a form of encryption that allows computations to be performed on encrypted data without having to decrypt it. The resulting computations are left in an encrypted form which, when decrypted, result in an output that is identical to the operations that may be performed on the unencrypted data. Homomorphic encryption can be used for privacy-preserving outsourced storage and computation. This allows data to be encrypted and outsourced to commercial cloud environments for processing, all while encrypted.

[0006] The present disclosure discloses a method for encrypting and searching documents using a combination of vectorization, hashing, and set intersection. The method includes defining a dictionary to map tokens to unique vectors, forming n-token combinations of the document, and hashing each combination using a nonlinear irreversible function such as a deep neural network. The output of the present disclosure is a set of D-dimensional vectors that represent the document.

[0007] In some embodiment, to query the database of documents, a query is similarly hashed to a set of D-dimensional vectors, and the relevance of the query to each document is measured via set intersection (or approximate intersection using cosine distance between vectors as a metric). The method has configurable security against brute force lookup table attacks, depending on the size of the dictionary lexicon L, the number of tokens n. If the limits of practical computation is expressed as M evaluations of the n– word inputs F In other words, when a user wants to query the database of documents, the query is similarly hashed into a set of D-dimensional vectors, and the similarity between the query and documents is measured using set intersection or cosine distance between vectors. The algorithm described in the present disclosure is highly secure against brute force lookup table attacks, depending on the size of the dictionary, the value of n, and the limits of practical computation.

[0008] These and other features and characteristics of the present invention, as well as the methods of operation and functions of the related elements of structures and the combination of parts and economies of manufacture, will become more apparent upon consideration of the following description with reference to the accompanying drawings, all of which form a part of this specification, wherein like reference numerals designate corresponding parts in the various figures. It is to be expressly understood, however, that the drawings are for the purpose of illustration and description only and are not intended as a definition of the limits of the invention. As used in the specification, the singular form of “a,” “an,” and “the” include plural referents unless the context clearly dictates otherwise.

BRIEF DESCRIPTION OF THE DRAWINGS AND APPENDICES

[0009] Additional advantages and details of non-limiting embodiments are explained in greater detail below with reference to the exemplary embodiments that are illustrated in the accompanying schematic figures, in which:

[0010] FIG.1A discloses an architecture of computing system for secure document search, according to some principles of the present disclosure;

[0011] FIG.1B discloses an architecture of prior art for information retrieval, according to some principles of the present disclosure;

[0012] FIG.1C discloses a representation of secure document search, according to some principles of the present disclosure;

[0013] FIG.1D discloses an exemplary representation of secure document search, according to some principles of the present disclosure;

[0014] FIG.2A discloses a flow chart of N-F hashing for secure document search, according to some principles of the present disclosure;

[0015] FIG.2B discloses a flowchart describing N-f-Hashing algorithmic, according to some principles of the present disclosure;

[0016] FIG.3 disclose a detailed flowchart illustrating secure search document, according to some principles of the present disclosure.

DESCRIPTION OF THE DISCLOSURE

[0017] In the present document, the word "exemplary" is used herein to mean "serving as an example, instance, or illustration." Any embodiment or implementation of the present subject matter described herein as "exemplary" is not necessarily to be construed as preferred or advantageous over other embodiments.

[0018] While the disclosure is susceptible to various modifications and alternative forms, specific embodiment thereof has been shown by way of example in the drawings and will be described in detail below. It should be understood, however that it is not intended to limit the disclosure to the particular forms disclosed, but on the contrary, the disclosure is to cover all modifications, equivalents, and alternative falling within the spirit and the scope of the disclosure.

[0019] The terms “comprises”, “comprising”, or any other variations thereof, are intended to cover a non-exclusive inclusion, such that a setup, device or method that comprises a list of components or steps does not include only those components or steps but may include other components or steps not expressly listed or inherent to such setup or device or method. In other words, one or more elements in a device or system or apparatus proceeded by “comprises… a” does not, without more constraints, preclude the existence of other elements or additional elements in the device or system or apparatus.

[0020] The terms "an embodiment", "embodiment", "embodiments", "the embodiment", "the embodiments", "one or more embodiments", "some embodiments", and "one embodiment" mean "one or more (but not all) embodiments of the invention(s)" unless expressly specified otherwise.

[0021] The terms "including", "comprising", “having” and variations thereof mean "including but not limited to", unless expressly specified otherwise.

[0022] As used herein, the terms “communication” and “communicate” may refer to the reception, receipt, transmission, transfer, provision, and/or the like of information (e.g., data, signals, messages, instructions, commands, and/or the like). For one unit (e.g., a device, a system, a component of a device or system, combinations thereof, and/or the like) to be in communication with another unit means that the one unit is able to directly or indirectly receive information from and/or transmit information to the other unit. This may refer to a direct or indirect connection (e.g., a direct communication connection, an indirect communication connection, and/or the like) that is wired and/or wireless in nature. Additionally, two units may be in communication with each other even though the information transmitted may be modified, processed, relayed, and/or routed between the first and second unit. For example, a first unit may be in communication with a second unit even though the first unit passively receives information and does not actively transmit information to the second unit. As another example, a first unit may be in communication with a second unit if at least one intermediary unit (e.g., a third unit located between the first unit and the second unit) processes information received from the first unit and communicates the processed information to the second unit. In some non-limiting embodiments, a message may refer to a network packet (e.g., a data packet and/or the like) that includes data. It will be appreciated that numerous other arrangements are possible.

[0023] As used herein, the term “computing device” may refer to one or more electronic devices that are configured to directly or indirectly communicate with or over one or more networks. A computing device may be a mobile or portable computing device, a desktop computer, a server, and/or the like. Furthermore, the term “computer” may refer to any computing device that includes the necessary components to receive, process, and output data, and normally includes a display, a processor, a memory, an input device, and a network interface. A “computing system” may include one or more computing devices or computers.

[0024] It will be apparent that systems and/or methods, described herein, can be implemented in different forms of hardware, software, or a combination of hardware and software. The actual specialized control hardware or software code used to implement these systems and/or methods is not limiting of the implementations. Thus, the operation and behavior of the systems and/or methods are described herein without reference to specific software code, it being understood that software and hardware can be designed to implement the systems and/or methods based on the description herein.

[0025] **FIG.1A** discloses an architecture of computing system for secure document search, according to some principles of the present disclosure.

[0026] In some embodiments, the architecture comprises a computing system and a database associated with the computing system. Generally, Personal Identifying Information (PII) may be stored in a database or static file location which may be encrypted when an application needs access to the PII. Thus, the PII is first decrypted before applying usual database querying or search methods to isolate the records of interest. The decryption requirement exposes all the data and may leak more information than the application really requires as shown in **FIG 1B** (prior art). The present disclosure focuses on solving the above-mentioned problem with the help of a homomorphic encryption. Homomorphic encryption is a form of encryption with an additional evaluation capability for computing over encrypted data without accessing the whole database. The result of such a computation remains encrypted.

[0027] In some embodiments, the computing system may include but not limited to, a processor 111, a memory 113 and an Input/Output (I/O) interface 115. When the user wishes to retrieve the PII from the database 103, the processor 111 may perform the following steps: (1) vectorizing the document tokens in which the user wishes to retrieve the details, (2) forming n-token combinations, and (3)'encrypting’ via feeding these n-token combinations to a nonlinear irreversible function. The ‘document’ is any collection (ordered or not) of tokens. For example, a text document is an ordered collection of words. Another example may be medical biometrics record that may be an unordered collection of chemical concentrations in the bloodstream. Initially, the processor 111 may define a ‘dictionary' which maps each token to a unique vector in some pre-determined number of D dimensions. The dictionaries are used to store data values in key value pairs. The dictionary is a collection which is ordered, changeable and does not allow duplicates. For instance, the words in a text document can be mapped to 300-dimensional glove embeddings, chemical concentrations can be mapped to the coordinates of some higher-dimensional manifold. Further, the processor **111** may consider all the n-token combinations of the document. In this way obtaining a “tensor” of shape of encoding the document may be performed. Finally, the processor **111** may feed the n-tensors to non-linear function such as neural network. As a result, every n-token combination of the document is thus hashed to a D- dimensional vector, the whole document forming a set of such vectors. Now, when a user wants to query a database of documents, i.e. find all documents which are relevant to the query, the query is similarly bashed to a set of D-dimensional vectors, and query-document relevance is measured via set intersection (or approximate intersection, using cosine distance between vectors as a metric). In other words, when a user wants to query the database of documents, the query is similarly hashed into a set of D-dimensional vectors, and the similarity between the query and documents is measured using set intersection or cosine distance between vectors as shown in **FIG. 1C**.

[0028] For instance, when a user wants to query a database of documents, the processor **111** may define a ‘dictionary' which maps each token to a unique vector in some pre-determined number of D dimensions. The dictionaries are used to store data values in key value pairs. As shown in **FIG. 1D.** The database contains IDs associated with the tokens. For example, phrase “fish tacos dinner” may be associated with the ID-1 and the phrase may be divided into n-token and each token are associated with a key value. For instance, key value for token “fish” may be (3,4), key value for token “tacos” may be (-4,6) and key value for token “dinner” may be (0,8). When the user search for the phrase “fish tacos dinner”, the value (-1, 18) may be displayed and thus the user may search and retrieve the values without decrypting the whole document.

[0029] **FIG. 2A** discloses a detailed flowchart hashing process, according to some principles of the present disclosure.

[0030] In some embodiments, when the user wishes to retrieve the data from a document, the user may select the particular document and the processor **111** may vectorize the words in the document. Once the whole document is converted to vectorized form, the processor **111** may use the combinations of the converted vector. In other words, the one or more phrases in the document may be divided into n-token and each token are associated with a key value. For instance, key value for token “fish” may be (3,4) in such manner all n-token combinations of the document (thus N-choose-n = NI/n! /(N-n)! combinations for a document with N tokens), a ‘tensor’ of shape (N-choose-n, n, D） encoding the document. The n and D are therefore configurable parameters of the algorithm. Finally, the encryption or more appropriately hashing step feeds each of the N-choose-n (n, D)-tensors to a highly nonlinear function F (e.g., deep neural network) which must be practically irreversible, and the output is itself a D- dimensional vector. Therefore, every n-token combination of the document is thus hashed to a D-dimensional vector, the whole document forming a set of such vectors as shown in **FIG. 2A.**

In some embodiments, the order of the hashing may be chosen as N from the set {1, 2, 3, ... infty}, which indicates how many words from the query/document will be processed at a time through the hashing function. If N=2, for example, then pairs of words will be hashed. If N=infinity, then all the words, regardless of how many, will be sent together to the hashing function. B. Decide on the functional hash F. The simplest possibility may be to sum the word embeddings of all the functions' inputs. A more complex possibility would be sending the inputs to a BERT encoder as shown in **FIG.2B**. The output of F will in general be a vector in some abstract space. The following is the procedure for hashing:

1. Start with a M-word query or document.

2. If N=infty, send all the words to the hashing function F, the output of which is the hashed value.
3. If N<infty, find all N-word combinations (MNC of these) and send each to F.
4. For N<infty, the hashed value is the set of all outputs of F for each N-word combination.

To compare two hashes' similarity, use e.g., min or avg cosine distance between set members.

[0031] **FIG. 3** illustrates the flowchart of for secure document search, according to some principles of the present disclosure.

[0032] In some embodiments, if secure database already exists, the user may retrieve the data from the documents. In an alternative embodiment, the processor **111** associated with the computing system may start with a corpus of documents to hash. The corpus or text corpus is a language resource consisting of a large and structured set of texts.

[0033] In some embodiments, the processor **111** may hash each document with N-f-hashing scheme and store in efficient database.

[0034] In some embodiments, the processor **111** retains a copy of the original document and hashed document correspondences for use later.

[0035] In some embodiments, the processor **111** may match the query against the database when user wishes to perform search or retrieve documents.

[0036] In some embodiments, the processor **111** hashes query with same N-f-hashing scheme used by database.

[0037] In some embodiments, the processor **111** compares the hashed query to each of the hashed documents to determine similarity scores (e.g. cosine distance between vectors).

[0038] In some embodiments, the processor **111** obtains ranked list of most similar documents indices. This may already be sufficient for join purposes.

[0039] In some embodiments, if the user needs original documents' contents, the user may apply for permissions to see only those retrieved indices' documents. As a result, every n-token combination of the document is thus hashed to a D- dimensional vector, the whole document forming a set of such vectors. Now, when a user wants to query a database of documents, i.e. find all documents which are relevant to the query.

[0040] The above description is illustrative and is not restrictive. Many variations of the invention may become apparent to those skilled in the art upon review of the disclosure.

[0041] One or more features from any embodiment may be combined with one or more features of any other embodiment without departing from the scope of the invention.

[0042] A recitation of "a", "an" or "the" is intended to mean "one or more" unless specifically indicated to the contrary.

[0043] All patents, patent applications, publications, and descriptions mentioned above are herein incorporated by reference in their entirety for all purposes. None is admitted to be prior art.

[0044] Although the invention has been described in detail for the purpose of illustration based on what is currently considered to be the most practical and preferred embodiments, it is to be understood that such detail is solely for that purpose and that the invention is not limited to the disclosed embodiments, but, on the contrary, is intended to cover modifications and equivalent arrangements that are within the spirit and scope of the invention. For example, it is to be understood that the present invention contemplates that, to the extent possible, one or more features of any embodiment can be combined with one or more features of any other embodiment.

ABSTRACT

A METHOD AND SYSTEM FOR SECURE DOCUMENT SEARCH

[0045] The present disclosure discloses a method for secure document search. The objective of the present disclosure focuses on avoiding the need to decrypt the whole database while retrieving the necessary data. The method discloses a method for encrypting and searching documents using a combination of vectorization, hashing, and set intersection. The method includes defining a dictionary to map tokens to unique vectors, forming n-token combinations of the document, and hashing each combination using a nonlinear irreversible function such as a deep neural network. The output of the present disclosure is a set of D-dimensional vectors that represent the document.

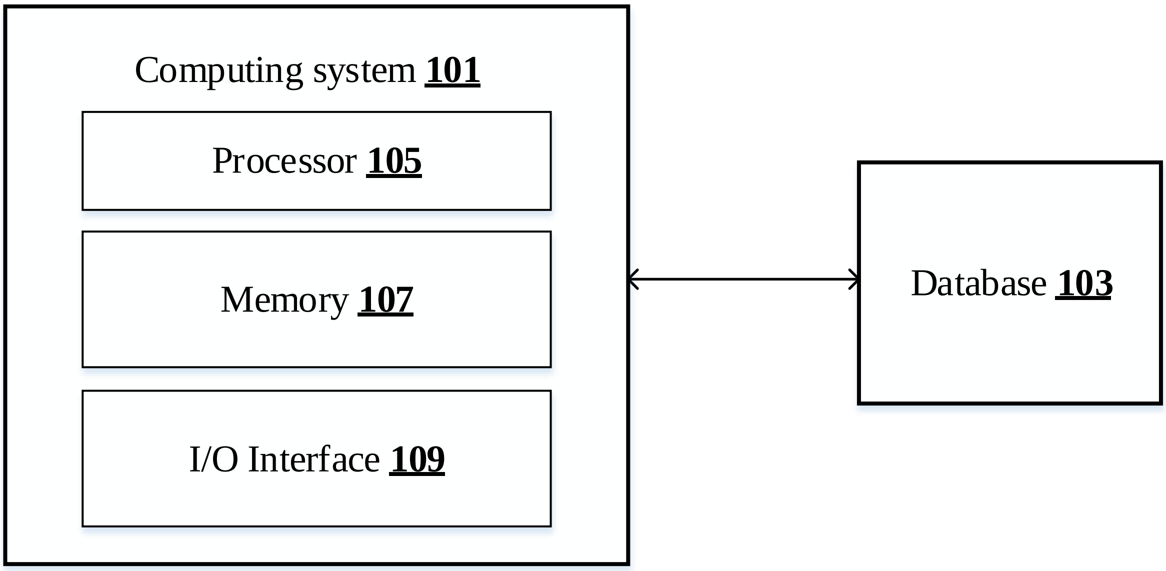

Computing system 101
Processor 105
Memory 107
I/O Interface 109
Database 103

FIG.1A

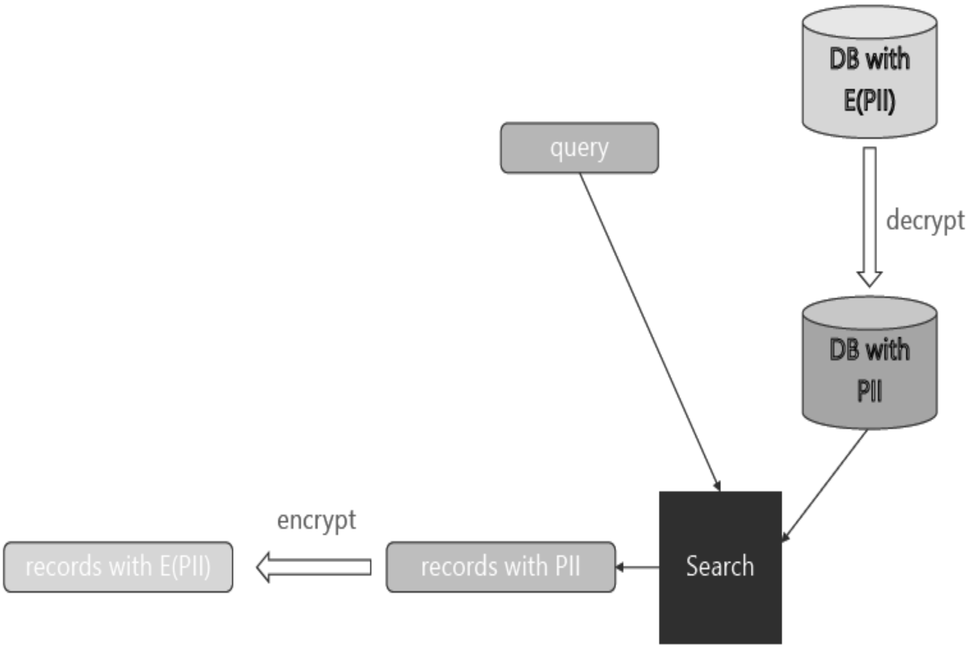

DB with E(PII)
decrypt
DB with PII
query
Search
records with PII
encrypt
records with E(PII)

FIG.1B

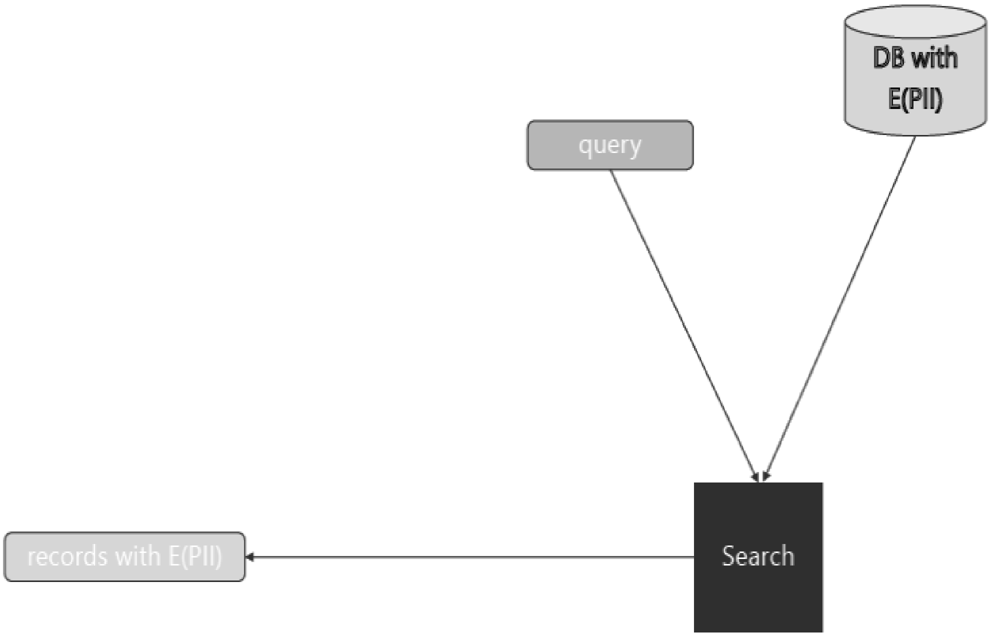
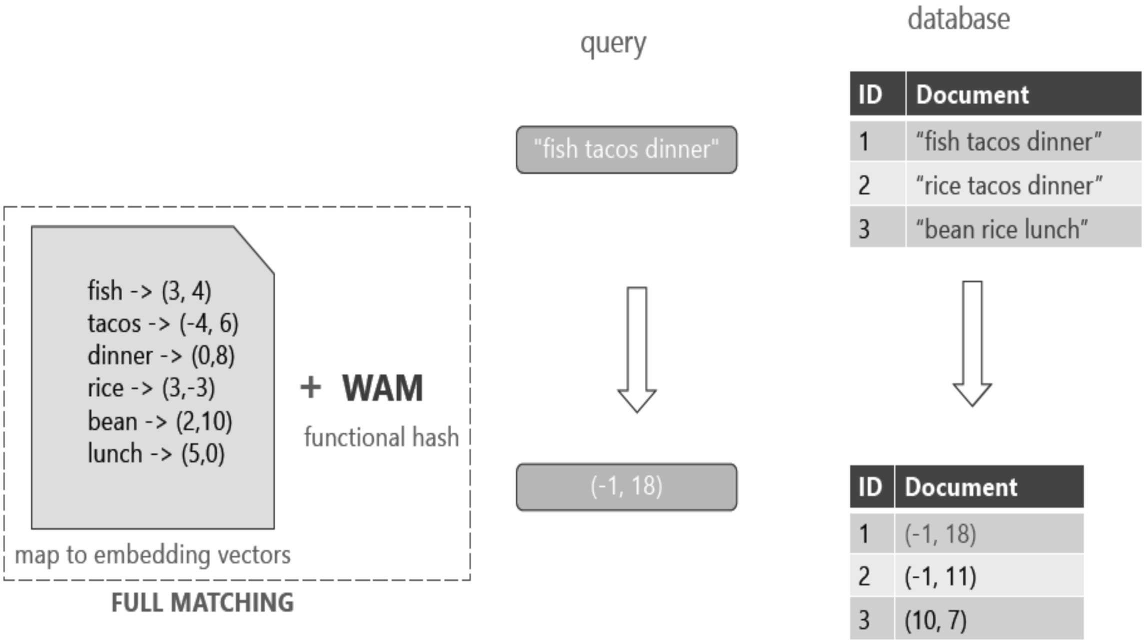


FIG.1C



ID	Document
1	“fish tacos dinner”
2	“rice tacos dinner”
3	“bean rice lunch”

ID	Document
1	(-1, 18)
2	(-1, 11)
3	(10, 7)

H(query) = F(fish, tacos, dinner) = (3,4) + (-4,6) + (0,8) = (-1, 18)

FIG.1D

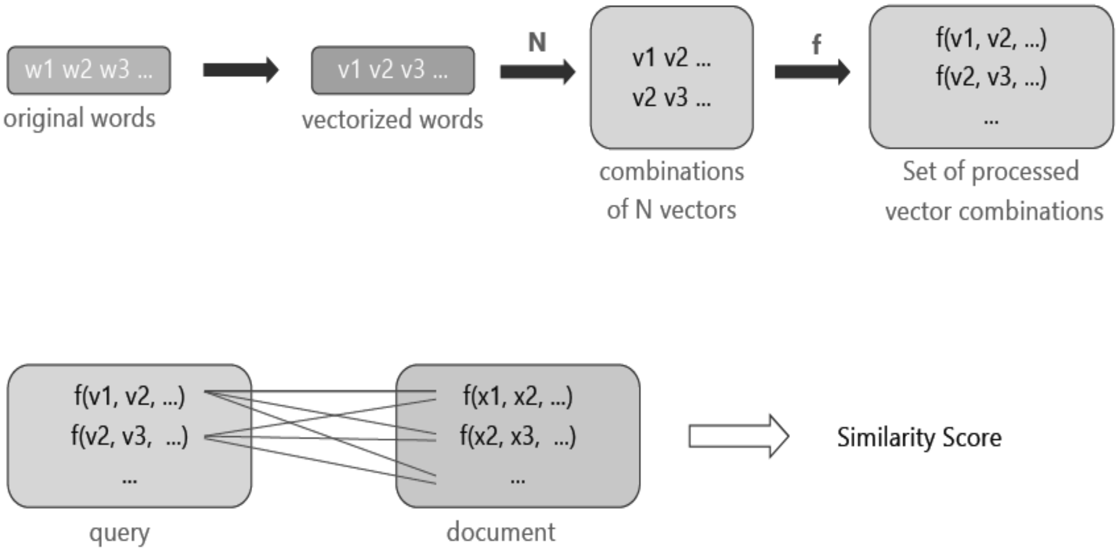

w1 w2 w3 ...
original words
v1 v2 v3 ...
vectorized words
N
v1 v2 ...
v2 v3 ...
combinations of N vectors
f
f(v1, v2, ...)
f(v2, v3, ...)
...
Set of processed vector combinations
f(v1, v2, ...)
f(v2, v3, ...)
...
query
f(x1, x2, ...)
f(x2, x3, ...)
...
document
Similarity Score

FIG.2A

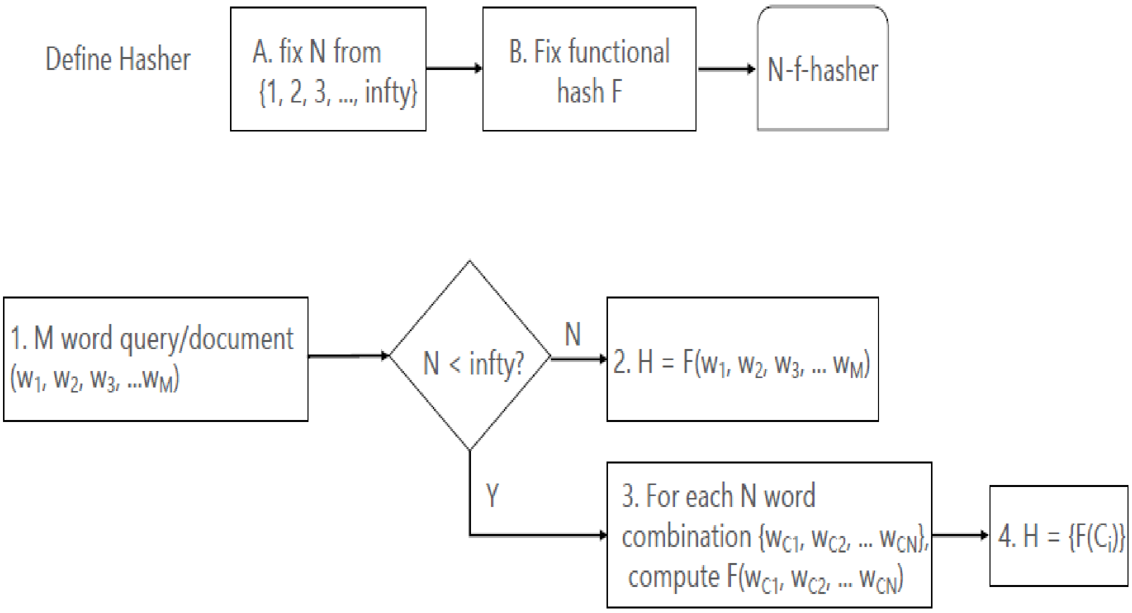

Define Hasher
A. fix N from {1, 2, 3, ..., infty}
B. Fix functional hash F
N-f-hasher
1. M word query/document (w_1, w_2, w_3, ...w_M)
N < infty?
N
Y
2. H = F(w_1, w_2, w_3, ... w_M)
3. For each N word combination {w_{C1}, w_{C2}, ... w_{CN}}, compute F(w_{C1}, w_{C2}, ... w_{CN})
4. H = {F(C_i)}

FIG.2B

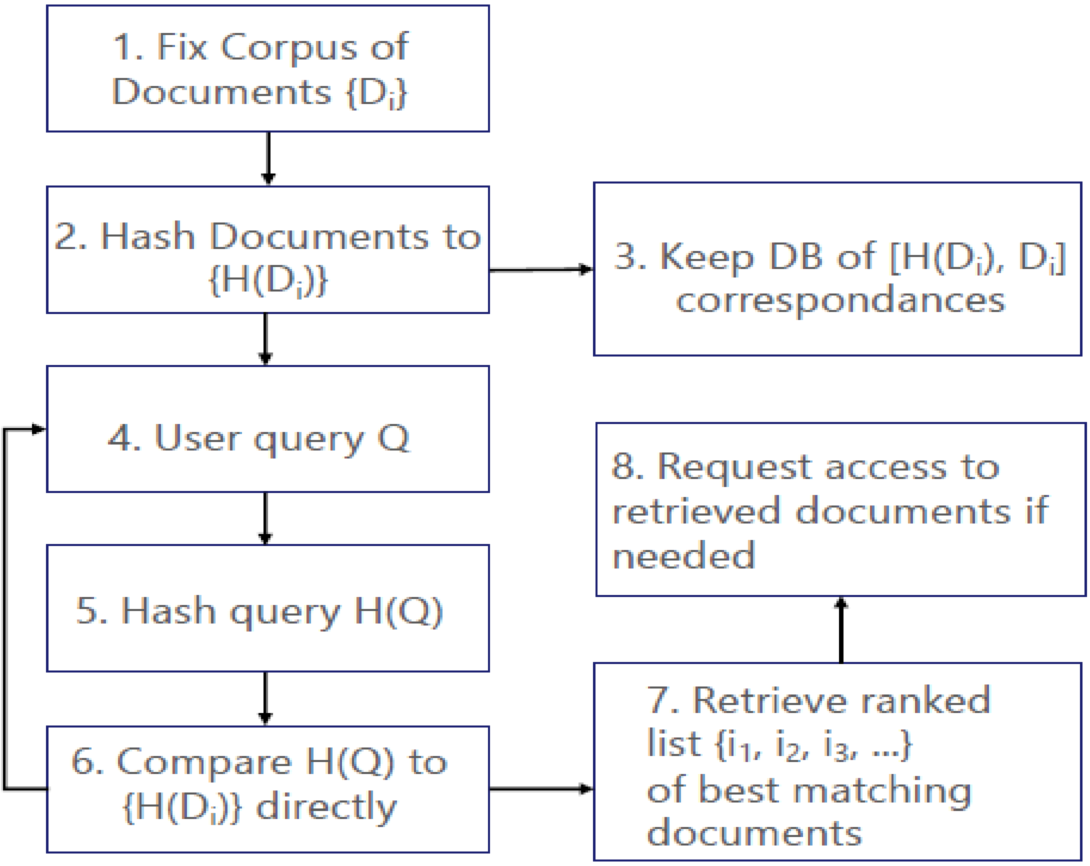

1. Fix Corpus of Documents $\{D_i\}$
2. Hash Documents to $\{H(D_i)\}$
3. Keep DB of $[H(D_i), D_i]$ correspondances
4. User query Q
5. Hash query H(Q)
6. Compare H(Q) to $\{H(D_i)\}$ directly
7. Retrieve ranked list $\{i_1, i_2, i_3, \ldots\}$ of best matching documents
8. Request access to retrieved documents if needed

FIG.3