# METHODS, DEVICES AND SYSTEMS OF TEXT STRING EMBEDDING AND AUTHENTICATION

## FIELD OF INVENTION

**[001]** The present specification relates broadly, but not exclusively, to methods, devices and systems for text string embedding and authentication.

## BACKGROUND

**[002]** Electronic-Know Your Customer (eKYC) is a digital due diligence process performed by a business to verify the identity of its clients and to assess potential risks of illegal intentions towards the business relationship. In eKYC processes, it is necessary for a business to conduct authentication to validate users' personal information. Users' personal information contains data in the form of text strings, e.g. users' addresses, users' names, etc.

**[003]** Current art uses text distance metrics based word comparison techniques, e.g. Edit Distance, for text string authentication. However, word comparison techniques do not capture semantic meaning of words and/or phrases, thus cannot recognise similarity between various expressions of a same or similar text string. For example, a text string "Bishan, Block 129, 04-01, Singapore 570129" is considered different to another text string "Bishan, Blk 129, 04-01, SG" under word comparison techniques in the current art, though they refer to the same address. The subsequent authentication is thus inaccurate.

**[004]** Word embedding techniques, e.g. word2vec, are able to extract semantic meaning from words or phrases. However, in the environment of eKYC, text strings (e.g. addresses, names, etc) are either captured via OCR techniques from photos of identification (ID) cards or entered by the users, and contain no context information for words in the text strings (e.g. address words in the addresses, words in the names). Thus, current word embedding techniques do not produce satisfying authentication results.

**[005]** A need therefore exists to provide methods, devices and systems that can generate accurate embedding for text strings so as to improve accuracy of authentication in eKYC processes.

## SUMMARY

**[006]** According to a first embodiment of the present specification, there is provided a method of text string embedding. The method comprises: accessing stored history data comprising text strings and device identifiers, wherein each of the device identifiers is associated with one or more of the text strings; and generating an embedding look-up table comprising a plurality of keys and a plurality of values, wherein the generating the embedding look-up table comprises: splitting the text strings in the history data into a plurality of words; generating a graph with the plurality of words and the device identifiers by: denoting each word in the plurality of words as a node of a first type; denoting each of the device identifiers as a node of a second type; and for each node of the second type, building edges by linking the node of the second type denoting one of the device identifiers to ones of the nodes of the first type, the linked ones of the nodes of the first type denoting words that are comprised in one or more of the text strings associated with the one of the device identifiers; training an embedding model using the generated graph; and constructing the embedding look-up table based on the trained embedding model, wherein each of the plurality of keys of the embedding look-up table comprises one of the nodes of the first type and the nodes of the second type, and wherein each of the plurality of values of the embedding look-up table comprises a vector corresponding to the one of the nodes of the first type and the nodes of the second type.

**[007]** According to a second embodiment of the present specification, there is provided a method of text string authentication. The method comprises: receiving a first text string from a user; splitting the first text string into a first plurality of words; and authenticating the first text string using one or more values in an embedding look-up table generated according to any one of claims 1 to 6 associated with keys in the embedding look-up table corresponding to each of the first plurality of words of the first text string.

**[008]** According to a third embodiment of the present specification, there is provided a text string embedding device. The text string embedding device comprises: a first storage device for storing history data comprising text strings and device identifiers, wherein each of the device identifiers is associated with one or more of the text strings; a training device coupled to the first storage device for generating an embedding look-up table comprising a plurality of keys and a plurality of values, the training device being configured to: access the stored history data; generate the embedding look-up table, wherein the generating the embedding look-up table comprises: splitting the text strings in the stored history data into a plurality of words; generating a graph with the plurality of words and the device identifiers by: denoting each word in the plurality of words as a node of a first type; denoting each of the device identifiers as a node of a second type; and for each node of the second type, building edges by linking the node of the second type denoting one of the device identifiers to ones of the nodes of the first type, the linked ones of the nodes of the first type denoting words that are comprised in one or more of the text strings associated with the one of the device identifiers; training an embedding model using the generated graph; and constructing the embedding look-up table based on the trained embedding model, wherein each of the plurality of keys of the embedding look-up table comprises one of the nodes of the first type and the nodes of the second type, and wherein each of the plurality of values of the embedding look-up table comprises a vector corresponding to the one of the nodes of the first type and the nodes of the second type; and a second storage device for storing the embedding look-up table.

**[009]** According to a fourth embodiment of the present specification, there is provided a text string authentication device coupled to an text string embedding device according to any one of claims 11 to 16, the text string embedding device including an embedding look-up table generated according to any one of claims 1 to 6, wherein the text string authentication device is configured to: receive a first text string; split the first text string into a first plurality of words; and authenticate the first text string using one or more values in the text string embedding look-up table associated with keys in the text string embedding look-up table corresponding to each of the first plurality of words of the first text string.

## BRIEF DESCRIPTION OF THE DRAWINGS

**[0010]** Embodiments and implementations are provided by way of example only, and will be better understood and readily apparent to one of ordinary skill in the art from the following written description, read in conjunction with the drawings, in which:

**[0011]** Figure 1 is a schematic diagram of a system for text string authentication according to an embodiment. As depicted in this embodiment, the system includes a text string embedding device and a text string authentication device coupled to the text string embedding device.

**[0012]** Figure 2 is a schematic diagram of a computing device according to an embodiment. The computing device can be implemented as a text string embedding device, a text string authentication device, or other components in the system for text string authentication, as described in the present specification.

**[0013]** Figure 3 is a flowchart showing steps in a method of text string embedding according to an embodiment.

**[0014]** Figure 4 is a flowchart showing steps in a method of text string authentication according to an embodiment.

**[0015]** Figure 5 illustrates history data that includes text strings and device identifiers, according to an embodiment. In this embodiment, the text strings are addresses. As shown in this embodiment, each of the addresses is associated with one of the device identifiers. Also shown in this embodiment, each of the addresses is split into a plurality of address words, and each of the plurality of address words is associated with one of the device identifiers.

**[0016]** Figure 6 shows an embodiment of a graph generated with the plurality of address words of the addresses and the device identifiers as illustrated in Figure 5.

**[0017]** Figure 7 shows an embodiment of an embedding look-up table. In this embodiment, the embedding look-up table is constructed based on an embedding model trained using the graph as shown in Figure 6.

**[0018]** Figure 8 shows a block diagram of a computer system suitable for use in executing at least some steps of a method of text string embedding or a method of text string authentication as described in the present specification. The computer system is also suitable for use as a text string embedding device, a text string authentication device, or a system for text string authentication in accordance with the embodiments shown in Figures 1 to 7.

**[0019]** Skilled artisans will appreciate that elements in the figures are illustrated for simplicity and clarity and have not necessarily been depicted to scale. For example, the dimensions of some of the elements in the illustrations, block diagrams or flowcharts may be exaggerated in respect to other elements to help to improve understanding of the present embodiments.

## DETAILED DESCRIPTION

**[0020]** Embodiments will be described, by way of example only, with reference to the drawings. Like reference numerals and characters in the drawings refer to like elements or equivalents.

**[0021]** Some portions of the description which follows are explicitly or implicitly presented in terms of algorithms and functional or symbolic representations of operations on data within a computer memory. These algorithmic descriptions and functional or symbolic representations are the means used by those skilled in the data processing arts to convey most effectively the substance of their work to others skilled in the art. An algorithm is here, and generally, conceived to be a self-consistent sequence of steps leading to a desired result. The steps are those requiring physical manipulations of physical quantities, such as electrical, magnetic or optical signals capable of being stored, transferred, combined, compared, and otherwise manipulated.

**[0022]** Unless specifically stated otherwise, and as apparent from the following, it will be appreciated that throughout the present specification, discussions utilizing terms such as “accessing”, “generating”, “splitting”, “denoting”, “building”, “linking”, “training”, “constructing”, “computing”, “applying”, “predicting”, “generating”, “executing”, “receiving”, ”splitting”, “fetching”, “authenticating” or the like, refer to the action and processes of a computer system, or similar electronic device, that

manipulates and transforms data represented as physical quantities within the computer system into other data similarly represented as physical quantities within the computer system or other information storage, transmission or display devices.

**[0023]** The present specification also discloses apparatus for performing the operations of the methods. Such apparatus may be specially constructed for the required purposes, or may comprise a computer or other device selectively activated or reconfigured by a computer program stored in the computer. The algorithms and displays presented herein are not inherently related to any particular computer or other apparatus. Various machines may be used with programs in accordance with the teachings herein. Alternatively, the construction of more specialized apparatus to perform the required method steps may be appropriate. The structure of a computer suitable for executing the various methods / processes described herein will appear from the description below. In the present specification, the terms “apparatus” and “device” are used interchangeably.

**[0024]** In addition, the present specification also implicitly discloses a computer program, in that it would be apparent to the person skilled in the art that the individual steps of the methods described herein may be put into effect by computer code. The computer program is not intended to be limited to any particular programming language and implementation thereof. It will be appreciated that a variety of programming languages and coding thereof may be used to implement the teachings of the disclosure contained herein. Moreover, the computer program is not intended to be limited to any particular control flow. There are many other variants of the computer program, which can use different control flows without departing from the scope of the specification.

**[0025]** Furthermore, one or more of the steps of the computer program may be performed in parallel rather than sequentially. Such a computer program may be stored on any computer readable medium. The computer readable medium may include storage devices such as magnetic or optical disks, memory chips, or other storage devices suitable for interfacing with a computer. The computer readable medium may also include a hard-wired medium such as exemplified in the Internet system, or wireless medium such as exemplified in the GSM mobile telephone system. The computer program when loaded and executed on such a computer

effectively results in an apparatus that implements the steps of any of the methods described herein.

**[0026]** It is observed in the present specification that when a same device is used to provide two or more addresses to an eKYC system of a service provider, the two or more addresses are likely to be the same, similar, or related addresses, as a sharing of device usually takes place among a cluster of users that share a same or related space, e.g. family members living in a same unit, neighbours living in a same or nearby apartment building, classmates sharing a classroom or a campus building, etc. In this manner, if address A and address A' are provided by a same device, address words in address A can be used to provide semantic meaning and/or context information for address words in address A', and vice versa.

**[0027]** In light of the above, embodiments of the present specification provide methods, devices, and systems to utilise device identifiers of users' devices to associate with address words so as to make semantic meaning and context information available for address words to facilitate generating an accurate embedding for addresses and improving accuracy of authentication in eKYC processes.

**[0028]** Furthermore, the present methods, devices, and systems generate a graph with device identifiers and the associated address words as nodes, and links between the nodes as edges. In this manner, the semantic meaning and context information of address words contained in the nodes and edges are learned by an embedding model to form an embedding for addresses. The embedding for addresses can be used for authentication as described herein. It will be appreciated by those skilled in the art that the embedding for addresses can also be used for similar address search, address comparison, etc.

**[0029]** In addition to addresses, the above observation of the present specification is also applicable to other text strings such as users' names, etc.

**[0030]** In addition or alternative to device identifiers of users' devices, other types of users personal information such as users' phone numbers, users' accounts with the service provider, etc., can also be used to associate with address words or words in

other text strings to provide context information for the address words or the words in other text strings.

**[0031]** In view of the above, the present methods, devices, and systems can further generate a graph with device identifiers, users' phone numbers, and/or users' accounts and the associated address words or the associated words in other text strings as nodes, and links between the nodes as edges. In this manner, the semantic meaning and context information of words in text strings (e.g. address words or words in other text strings) contained in the nodes and edges are learned by an embedding model to form an embedding for text strings. The embedding for text strings can be used for authentication of text strings (e.g. addresses, names, etc.) as described herein, or similarity quantification of the text strings.

**[0032]** Figure 1 shows a schematic diagram of a system 100 for text string authentication according to an embodiment of the present specification. The system 100 can be a portion of an eKYC system (not shown in Figure 1) of a service provider. It is also appreciable to those skilled in the art that the system 100 can be implemented as an eKYC system without substantial modification.

**[0033]** As depicted in this embodiment, the system 100 includes a text string embedding device 102 and a text string authentication device 112 coupled to the text string embedding device 102.

**[0034]** As shown in Figure 1, the embedding device 102 comprises a training device 106 coupled to a first storage device 104. The first storage device 104 stores history data comprising text strings such as users' addresses, users' names, etc. The history data also comprises data that is associated with the text strings, such as device identifiers, users' phone numbers, users' accounts with the service provider, etc. In some embodiments, a device identifier can be generated by the system 100 when previously receiving an address from a user's device. In this manner, each address in the history data is associated with a device identifier corresponding to the device that provides the address. The device identifier can be in the format of "Device 1", "Device 2", etc. It will be appreciated by those skilled in the art that the device identifier can be in other format.

**[0035]** Based on serial numbers of users' devices, different devices can be recognised and assigned with different device identifiers. Likewise, a device

providing multiple addresses at different points of time can be recognised as the same device and be assigned with the same device identifier. As such, one or more addresses provided by a same device can be associated with a same device identifier in the history data.

**[0036]** Alternative to the system 100, the device identifier can be generated by other components in an eKYC system (not shown in Figure 1) and transmitted to the first storage device 104 for storage. For the sake of simplicity, the generation of the device identifier is not discussed in further detail herein.

**[0037]** In some embodiments, when a user provides an address to the service provider from the user's device, a name of the user is also provided to accompany the address. The name is linked to the address in the history data, and can also be subject to the text string embedding as described herein.

**[0038]** In some embodiments, in addition to or alternative to a device identifier corresponding to the user's device, the user's phone number and/or the user's account with the service provider can also be associated with the address and/or name to provide context information for words in these text strings.

**[0039]** The text string embedding device 102 is configured to perform a method of text string embedding. Figure 3 shows an embodiment 300 of a method of text string embedding. As shown in Figure 3, the text string embedding device 102 is configured at least to perform the following steps in the method 300 of text string embedding. Each of the steps of the method may be performed sequentially; or when applicable, in parallel or in any order.

**[0040]** The method 300 includes step 302 of accessing stored history data comprising text strings and device identifiers, wherein each of the device identifiers is associated with one or more of the text strings. In some embodiments, the text strings comprise addresses. In these embodiments, each of the device identifiers is associated with one or more of the addresses.

**[0041]** The method 300 further includes step 304 of generating an embedding look-up table comprising a plurality of keys and a plurality of values. The step 304 of generating the embedding look-up table comprises step 304a of splitting the text strings in the history data into a plurality of words; step 304b of generating a graph with the plurality of

words and the device identifiers; step 304c of training an embedding model using the generated graph; and step 304d of constructing the embedding look-up table based on the trained embedding model, wherein each of the plurality of keys of the embedding look-up table comprises one of the nodes of the first type and the nodes of the second type, and wherein each of the plurality of values of the embedding look-up table comprises a vector corresponding to the one of the nodes of the first type and the nodes of the second type.

**[0042]** The step 304b of generating a graph with the plurality of words and the device identifiers comprises: denoting each word in the plurality of words as a node of a first type; denoting each of the device identifiers as a node of a second type; and for each node of the second type, building edges by linking the node of the second type denoting one of the device identifiers to ones of the nodes of the first type, the linked ones of the nodes of the first type denoting words that are comprised in one or more of the text strings associated with the one of the device identifiers.

**[0043]** In some embodiments where the text strings comprise addresses, the plurality of words comprise a plurality of address words. In these embodiments, in step 304a, the text string embedding device 102 is configured to split the addresses in the history data into a plurality of address words. In step 304b, the text string embedding device 102 is configured to generate a graph with the plurality of address words and the device identifiers by: denoting each address word in the plurality of address words as a node of a first type; denoting each of the device identifiers as a node of a second type; and for each node of the second type, building edges by linking the node of the second type denoting one of the device identifiers to ones of the nodes of the first type, the linked ones of the nodes of the first type denoting address words that are comprised in one or more of the addresses associated with the one of the device identifiers.

**[0044]** In some embodiments where the text strings comprise names, the plurality of words comprise a plurality of name words. In these embodiments, in step 304a, the text string embedding device 102 is configured to split the names in the history data into a plurality of name words. In step 304b, the text string embedding device 102 is configured to denote each name word in the plurality of name words as a node of a first type; denoting each of the device identifiers as a node of a second type; and for each node of the second type, building edges by linking the node of the second type denoting one of the device identifiers to ones of the nodes of the first type, the linked ones of the

nodes of the first type denoting name words that are comprised in one or more of the names associated with the one of the device identifiers.

**[0045]** In some embodiments where the text strings comprise addresses and names, the plurality of words comprise a plurality of address words and name words. In these embodiments, in step 304a, the text string embedding device 102 is configured to split the addresses and the names in the history data into a plurality of address words and name words. In step 304b, the text string embedding device 102 is configured to generate a graph with the plurality of address words and name words and the device identifiers by: denoting each address or name word in the plurality of address words and name words as a node of a first type; denoting each of the device identifiers as a node of a second type; and for each node of the second type, building edges by linking the node of the second type denoting one of the device identifiers to ones of the nodes of the first type, the linked ones of the nodes of the first type denoting address words that are comprised in one or more of the addresses associated with the one of the device identifiers or denoting name words that are comprised in one or more of the names associated with the one of the device identifiers.

**[0046]** In some embodiments, in addition to or alternative to the device identifiers, users' phone numbers and/or users' accounts with the service provider can also be associated with addresses and/or names in the text strings to provide context information for words in the text strings. In these embodiments, in step 304b, the text string embedding device 102 is configured to generate a graph with the plurality of address words and name words and the device identifiers by: denoting each address or name word in the plurality of address words and name words as a node of a first type; denoting each of the device identifiers, users' phone numbers and/or users' accounts as a node of a second type; and for each node of the second type, building edges by linking the node of the second type denoting one of the device identifiers, users' phone numbers and/or users' accounts to ones of the nodes of the first type, the linked ones of the nodes of the first type denoting address words that are comprised in one or more of the addresses associated with the one of the device identifiers, users' phone numbers and/or users' accounts or denoting name words that are comprised in one or more of the names associated with the one of the device identifiers, users' phone numbers and/or users' accounts.

**[0047]** Steps 302 and 304 are described below with reference to Figures 5 to 7, which depict embodiments in which the text strings comprise only addresses. That is, embodiments in which the text strings are addresses. In some other embodiments, the text strings can further comprise names, etc..

**[0048]** In some embodiments of step 302, the training device 106 in the text string embedding device 102 is configured to access history data stored in the first storage device 104 of the text string embedding device.

**[0049]** The history data comprises text strings and device identifiers as described above. An embodiment of the history data 500 stored in the first storage device 104 is depicted in Figure 5. In this embodiment, the text strings comprise only addresses. In this embodiment, an exemplified portion of the history data 500 include: address #1 “Bishan Block 129 Singapore 570129” and its associated device identifier “device 1”; address #2 “Bishan Block 131 06-02 Singapore 570131” and its associated device identifier “device 2”; and address #3 “Bishan Blk 129 SG” and its associated device identifier “device 1”.

**[0050]** In the history data as shown in Figure 5, each of the device identifiers is associated with one or more of the addresses. For example, in the embodiment 500 of history data, the device identifier “device 2” is associated with address #2, whereas the device identifier “device 1” is associated with address #1 and address #3. It can be seen that address #1 and address #3 are associated with a shared device denoted by the device identifier “device 1”. Such information, i.e. information of multiple addresses being associated with a shared device, is advantageously used in the present specification to build context information and provide semantic meaning for address words in the address. For example, as address #1 and address #3 are associated with the same device, the address “Bishan Block 129 Singapore 570129” and the address “Bishan Blk 129 SG” are of a high possibility to be the same or related, as they can be provided by family members living in a same unit, neighbours living in a same or nearby apartment building, classmates sharing a same classroom or a campus building, etc. In this manner, semantic meaning and context information of one address are rendered by other address(es) associated to the same device identifier.

**[0051]** In step 304, the training device 106 in the text string embedding device 102 is configured to generate a text string embedding look-up table for text string embedding based on the addresses and the device identifiers in the history data. The embedding look-up table comprises a plurality of keys and a plurality of values. The step 304 of generating the embedding look-up table includes steps 304a, 304b, 304c and 304d as mentioned above and described below.

**[0052]** In step 304a, the training device 106 in the text string embedding device 102 is configured to split the addresses in the history data 500 into a plurality of address words.

**[0053]** With reference to the embodiment of history data 500 in Figure 5, the training device 106 splits the address #1, address #2 and address #3 in the history data 500 into a plurality of address words. The address words include words, abbreviations of words, numbers, etc. For example, the address “Bishan Block 129 Singapore 570129” is split into address words: “Bishan”, “Block”, “129”, “Singapore”, “570129”; the address “Bishan Block 131 06-02 Singapore 570131” is split into address words: “Bishan”, “Block”, “131”, “06-02”, “Singapore”, “570131”; and the address “Bishan Blk 129 SG” is split into address words: “Bishan”, “Blk”, “129”, “SG”. In this manner, a plurality of address words “Bishan”, “Block”, “129”, “Singapore”, “570129”, “131”, “06-02”, “Blk”, “129”, “SG” are split from the addresses. In some embodiments, a duplicate entry of the same address word can be deleted from the plurality of address words for the sake of simplicity. It can be seen that address words “Singapore”, “Bishan” and “Block” have appeared in multiple addresses. That is, the address word “Singapore” is comprised in two addresses, i.e. address #1 and address #2; the address word “Bishan” is comprised in all of the three addresses, i.e. address #1, address #2, and address #3; the address word “Block” is comprised in two addresses, i.e. address #1 and address #2.

**[0054]** As the addresses are associated with the respective device identifiers in the history data 500, it is inherent that the address words split from the addresses are also associated with the respective device identifiers. For example, each of the address words “Bishan”, “Block”, “129”, “Singapore” and “570129” split from the address “Bishan Block 129 Singapore 570129” is associated with the device identifier “device 1”. Likewise, each of the address words “Bishan”, “Block”, “131”, “06-02”, “Singapore” and “570131” split from the address “Bishan Block 131 06-02 Singapore 570131” is associated with the device identifier “device 2”, whereas each

of the address words “Bishan”, “Blk”, “129” and “SG” split from the address “Bishan Blk 129 SG” is associated with the device identifier “device 1”.

**[0055]** For address words (e.g. “Singapore”, “Bishan” and “Block”) that are comprised in multiple addresses (e.g. address #1 and address #2) of the addresses, as the multiple addresses are associated with respective device identifiers (e.g. “address 1” is associated with “device 1”, “address 2” is associated with “device 2”), it is inherent that these address words are also associated with the respective multiple device identifiers. For example, each of the address words “Singapore”, “Bishan”, and “Block” is associated with two device identifiers “device 1” and “device 2”.

**[0056]** As described above, the address words “Bishan”, “Block”, “129”, “Singapore” and “570129” split from address # 1 and the address words “Bishan”, “Blk”, “129” and “SG” split from address # 1, which are associated with the same device identifier “device 1”, are considered to have the same or related semantic and geographic meaning. Such a relationship, if learned, can advantageously provide semantic meaning and context information for address words. The present specification reserves such relationships along with the addresses and device identifiers in a graph as generated in step 304b to achieve a holistic and accurate text string embedding.

**[0057]** In step 304b, the training device 106 in the text string embedding device 102 is configured to generate a graph with the plurality of address words of the addresses as split in step 304a and the device identifiers associated with the plurality of address words. Since the address words and the device identifiers have different attributes, they are captured as different types of nodes in the graph.

**[0058]** An embodiment of such a graph 600 is shown in Figure 6. Referring to Figure 6, the step 304b of generating the graph 600 comprises the following steps:

**[0059]** First, the training device 106 in the text string embedding device 102 denotes each address word in the plurality of address words of each of the addresses as a node of a first type. As shown in Figure 6, each address word “Bishan”, “Block”, “129”, “Singapore”, “570129” of the address “Bishan Block 129 Singapore 570129” is denoted as a node of a first type. Likewise, each address word “Bishan”, “Block”, “131”, “06-02”, “Singapore”, “570131” of the address “Bishan Block 131 06-02

Singapore 570131" is denoted as a node of the first type; each address word "Bishan", "Blk", "129", "SG" of the address "Bishan Blk 129 SG" is also denoted as a node of the first type.

**[0060]** As described above, in some embodiments, an address word can appear multiple times in multiple addresses. Such an address word is only captured once in the graph 600 and denoted as one node of the first type. For example, the address word "Singapore" comprised in two of the addresses "Bishan Block 129 Singapore 570129" and "Bishan Block 131 06-02 Singapore 570131" is denoted as one node of the first type in the graph 600. As shown in Figure 6, the graph 600 includes 10 nodes of the first type, i.e. node 602 denoting "Singapore", node 604 denoting "SG", node 606 denoting "Bishan", node 608 denoting "Blk", node 610 denoting "Block", node 612 denoting "129", node 614 denoting "570129", node 616 denoting "06-02", node 618 denoting "131", and node 620 denoting "570131".

**[0061]** Second, the training device 106 in the text string embedding device 102 denotes each of the device identifiers as a node of a second type. As shown in Figure 6, the graph 600 includes 2 nodes of the second type, i.e. node 622 denoting the device identifier "device 1" and node 624 denoting the device identifier "device 2".

**[0062]** As describe above, each of the address words "Singapore", "Bishan" and "Block" that are comprised in multiple addresses is associated with the respective device identifiers. In this regard, among the above described 10 nodes of the first type, the node 602 denoting "Singapore" is associated with two nodes of the second type, i.e. the node 622 denoting the device identifier "device 1" and the node 624 denoting the device identifier "device 2". Similarly, each of the node 606 denoting "Bishan" and the node 610 denoting "Block" is also associated with the node 622 denoting the device identifier "device 1" and the node 624 denoting the device identifier "device 2".

**[0063]** Third, for each node of the second type, the training device 106 in the text string embedding device 102 builds edges by linking the node of the second type denoting one of the device identifiers to ones of the nodes of the first type, the linked ones of the nodes of the first type denoting address words that are comprised in one or more of the addresses associated with the one of the device identifiers. As shown in

Figure 6, for node 622 of the second type denoting the device identifier “device 1”, the training device 106 builds edges 626, 628, 630, 632, 634, 636, 638 by linking the node 622 of the second type to the nodes 602, 604, 606, 608, 610, 612, 614 of the first type denoting address words “Singapore”, “SG”, “Bishan”, “Blk”, “Block”, “129”, “570129” that are comprised in one or more of the addresses (i.e. address #1 and address #2) associated with the device identifier “device 1” denoted by the node 622. Likewise, for node 624 of the second type denoting the device identifier “device 2”, the training device 106 builds edges 640, 642, 646, 648, 650, 652 by linking the node 624 of the second type to the nodes 602, 606, 610, 616, 618, 620 of the first type denoting address words “Singapore”, “Bishan”, “Block”, “06-02”, “131”, “570131” that are comprised in one or more of the addresses (i.e. address #2) associated with the device identifier “device 2” denoted by the node 624.

**[0064]** The graph 600 generated in the above manner contains address words denoted by nodes of the first type, device identifiers denoted by nodes of the second type, relationships between device identifiers and address words denoted by edges, and relationships between address words of one address and address words of another address linked by a shared device identifier. As described above, the relationships between device identifiers and address words denoted by edges and the relationships between address words of one address and address words of another address linked by a shared device identifier advantageously facilitate the present specification to build semantic meaning and context information for the address words.

**[0065]** In step 304c, the training device 106 in the text string embedding device 102 trains an embedding model using the generated graph 600.

**[0066]** In some embodiments, the step 304c of training the embedding model using the generated graph 600 comprises the following sub-steps.

**[0067]** First, in sub-step a, for each node 602, 604, 606, 608, 610, 612, 614, 616, 618, 620, 622, 624 in the generated graph 600, the training device 106 can compute a vector based on information of the node’s neighbours, wherein the node’s neighbours comprise one or more of nodes linked to the node within a predetermined number of edges. The predetermined number of edges can be decided based on the accuracy

requirements for text string authentication or other downstream applications utilising the text string embedding. For example, the predetermined number of edges can be 5.

**[0068]** In some embodiments, the device identifiers can include geographic information of the respective devices as attributes. The geographic information can include GPS information of the respective devices. Such geographic information can be captured by the system 100 or other components in an eKYC system. In these embodiments, the information of the node's neighbours includes geographic information corresponding to one or more of the device identifiers "device 1", "device 2" denoted by one or more of the nodes 622, 624 of the second type in the nodes' neighbours.

**[0069]** In some embodiments, when the training device 106 computes a vector for a node of the first type denoting an address word that has an appearance frequency lower than a threshold, the training device 106 can compute the vector for the node as a vector for unknown words. For example, the threshold can be 5% or a predetermined value based on the practical needs. The computing and training of the vector for unknown words are readily understandable to those skilled in the art.

**[0070]** Subsequently, in sub-step b, the training device 106 can apply random walk on all the nodes 602, 604, 606, 608, 610, 612, 614, 616, 618, 620, 622, 624 in the generated graph 600 to generate node sequences.

**[0071]** Thereafter, in sub-step c, for every node in a node sequence of the generated node sequences, the training device 106 can predict the node's neighbours in the node sequence to form a predicted node distribution.

**[0072]** Subsequently, in sub-step d, the training device 106 can train the embedding model based on the predicted node distribution and a true node distribution. The true node distribution is available from real node sequences.

**[0073]** The above sub-steps a, c and d of step 304c can be iterated for all the nodes 602, 604, 606, 608, 610, 612, 614, 616, 618, 620, 622, 624 in the generated graph 600 to train the embedding model for all the address words in the generated graph 600.

**[0074]** In an embodiment of an iteration process, the training device 106 updates the vector of each node by information of the node's neighbours. Subsequently, the training device 106 predicts the node's neighbours in the node sequences. The node's

neighbours in the node sequences are different from the node's neighbours in the graph 600. Thereafter, the training device 106 trains the embedding model based on the predicted node distribution and the true node distribution so as to update the vector of every node.

**[0075]** In some embodiments, the graph 600 generated in step 304c can be trained by a heterogeneous neural network since the address words and the device identifiers in the graph 600 are treated as two types of nodes. In this regard, the embedding model can be a general attributed multiplex heterogeneous network embedding (GATNE) model, a hierarchical attention network (HAN) model, or a heterogeneous graph neural network (HetGNN) model. It is understandable to those skilled in the art that other graph neural networks can also be used for training the embedding model of the present specification.

**[0076]** In step 304d, the training device 106 in the text string embedding device 102 constructs the embedding look-up table based on the trained embedding model. Each of the plurality of keys of the embedding look-up table comprises one of the nodes of the first type and the nodes of the second type. Each of the plurality of values of the embedding look-up table comprises a vector corresponding to the one of the nodes of the first type and the nodes of the second type. The vector corresponding to the one of the nodes of the first type and the nodes of the second type is trained in step 304c as described above.

**[0077]** An embodiment 700 of the embedding look-up table is shown in Figure 7. The embedding look-up table 700 comprises a plurality of keys and a plurality of values. As shown in Figure 7, each of the plurality of keys of the embedding look-up table 700 comprises one of the nodes of the first type denoting the address words "Singapore", "SG", "Bishan", "Blk", "Block", "129", "131", "570129", "570131", "06-02" and the nodes of the second type denoting the device identifiers "device 1", "device 2".

**[0078]** As shown in Figure 7, each of the plurality of values of the embedding look-up table 700 comprises a vector corresponding to the one of the nodes of the first type and the nodes of the second type. In some embodiments, the vector is trained in step 304c as described above.

**[0079]** The embedding look-up table 700 can also include a key for address words having a low appearance frequency. The low appearance frequency can be an appearance frequency lower than a threshold. The threshold can be 5% or a

predetermined value based on the practical needs. The key for address words having a low appearance frequency can be treated as a key for unknown words. The key for unknown words has a corresponding vector as its value, e.g. “vector_UNK”. The vector “vector_UNK” is trained together with other vectors in step 304c. Based on practical needs in different scenario, the vector “vector_UNK” may be equal to 0, or other various values.

**[0080]** The embedding look-up table as generated in the above described embodiments of the method provides a more accurate text string embedding compared to the current arts.

**[0081]** As shown in Figure 1, the embedding look-up table generated by the training device 106 can be transmitted to a second storage device 108 for storage. The second storage device 108 is coupled to the training device 106 and a text string authentication device 112 for authentication of new text strings (e.g. new addresses, new names, etc) in downstream applications.

**[0082]** In the embodiment shown in Figure 1, the first storage device 104 and the second storage device 108 are comprised in the text string embedding device 102. It is understandable to those skilled in the art that, in some other embodiments, the first storage device 104 and the second storage device 108 may be external hardware components coupled to the text string embedding device 102. In addition, in some alternative embodiments, the first storage device 104 and the second storage device 108 may be implemented by a single storage device, either as a hardware component comprised in the text string embedding device 102 or as an external hardware component communicable to the text string embedding device 102.

**[0083]** As shown in Figure 1, the system 100 can further comprise an input device 110 for receiving new text strings. The input device 100 is coupled to the first storage device 104 for storing the new text strings. The input device 100 is also coupled to the text string authentication device 112. The text string authentication device 112 is coupled to the second storage device 108 that stores the embedding look-up table.

**[0084]** The text string authentication device 112 is configured to perform a method of text string authentication. Figure 4 shows an embodiment 400 of a method of text string authentication. As shown in Figure 4, the text string authentication device 112 is configured at least to perform the following steps in the method 400 of text string

authentication. Each of the steps of the method may be performed sequentially; or when applicable, in parallel or in any order.

**[0085]** The method 400 includes step 402 of receiving a first text string from a user.

**[0086]** The method 400 further includes step 404 of splitting the first text string into a first plurality of words.

**[0087]** The method 400 further includes step 406 of authenticating the first text string using one or more values in an embedding look-up table generated as described herein associated with keys in the embedding look-up table corresponding to each of the first plurality of words of the first text string.

**[0088]** In some embodiments, in step 402, the text string authentication device 112 is configured to receive a first text string 116 transmitted from the input device 100 that is coupled to the text string authentication device 112. In the embodiments, the first text string 116 is a first address 116. The first address 116 can be provided by a user.

**[0089]** In some embodiments, in step 404, the text string authentication device 112 is configured to split the first address 116 into a first plurality of address words.

**[0090]** In some embodiments, in step 406, the text string authentication device 112 is configured to authenticate the first address 116 using one or more values in the embedding look-up table 800 associated with keys in the embedding look-up table 800 corresponding to each of the first plurality of address words of the first address 116.

**[0091]** In some embodiments, the step 406 of authenticating the first address 116 further includes the following sub-steps. Each of the sub-steps of the step 406 may be performed sequentially; or when applicable, in parallel or in any order.

**[0092]** The step 406 includes sub-step 406a of fetching one or more values in the embedding look-up table associated with keys in the embedding look-up table corresponding to each of the first plurality of address words of the first address 116, wherein any unknown word in the first plurality of address words of the first address 116 is mapped to a key for unknown words having a vector for unknown words as a value.

**[0093]** The step 406 further includes sub-step 406b of generating a vector for the first address 116 by a sum calculation or an average calculation of the fetched values of the first plurality of address words.

**[0094]** In some embodiments, the step 406 of authenticating the first address 116 further includes the following sub-steps. Each of the sub-steps of the step 406 may be performed sequentially; or when applicable, in parallel or in any order.

**[0095]** The step 406 further includes sub-step 406c of receiving a second text string 118 of the user from an official source.

**[0096]** The step 406 further includes sub-step 406d of splitting the second text string 118 into a second plurality of words.

**[0097]** The step 406 further includes sub-step 406e of fetching one or more values in the embedding look-up table associated with keys in the embedding look-up table corresponding to each of the second plurality of words of the second text string 118, wherein any unknown word in the second plurality of words of the second text string 118 is mapped to a key for unknown words having a vector for unknown words as a value.

**[0098]** The step 406 further includes sub-step 406f of generating a vector for the second text string 118 by a sum calculation or an average calculation of the fetched values of the second plurality of words.

**[0099]** In some embodiments, the second text string 118 is a second address 118. The second plurality of words are a second plurality of address words.

**[00100]** In some embodiments, the step 406 of authenticating the first address 116 further includes a sub-step 406g: authenticating the first address 118 based on a similarity between the vector for the first address 116 and the vector for the second address 118. In some embodiments, in step 406g, the similarity between the between the vector for the first address 116 and the vector for the second address 118 can be based on a cosine similarity calculation. In some examples, if the similarity is greater than a threshold value, the first address 116 can be authenticated as a true address. The threshold value can be 0.85 or other predetermined value based on the practical requirements.

**[00101]** In some embodiments, the official source mentioned in the sub-step 406c comprises a government database or a third party database authorised by a government that provide official personal information of users, including identification (ID) numbers, names, addresses, etc. The second address received from the official source can be obtained based on the user's ID number provided in an eKYC process to the system 100.

**[00102]** The above paragraphs describe embodiments of steps 402, 404 and 406 in scenarios where the text strings are addresses. Based on these embodiments, it is understandable to those skilled in the art that the steps 402, 404, 406 are similar in scenarios where the text strings are names.

**[00103]** By virtue of the text string embedding achieved in the present specification, the text string authentication between a text string (e.g. an address) provided from a user and an official text string (e.g. address) of the user from an official source can achieve a more accurate result.

**[00104]** Furthermore, by virtue of the text string embedding achieved in the present specification, the above described method of text string authentication can be modified to be a method of identifying potential fraudsters by calculating a similarity between a vector of an address of a known fraudster and a vector of another address of a suspect fraudster based on the text string embedding look-up table as constructed in the present specification. It is understandable to those skilled in the art that the text string embedding can also be used for downstream applications such as similar address search, address comparison, etc.

**[00105]** As shown in Figure 1, the system 100 can further comprise an output device 114 coupled to the text string authentication device 112. A result of the text string authentication can be transmitted from the text string authentication device 112 to the output device 114 for display. Alternatively or additionally, the result of the text string authentication can be transmitted from the text string authentication device 112 to a transmitter for transmitting to users' devices.

**[00106]** Figure 2 shows a schematic diagram of a device 200. The device 200 can be implemented as the text string embedding device 102, the text string authentication device 112, or other components in the system 100 for text string authentication, as described in the present specification.

**[00107]** The device 200 at least includes a processor 202 and a memory 204. The processor 202 and the memory 204 are interconnected. The memory 204 includes computer program code (not shown in Figure 2). The memory 204 and the computer program code are configured to, with the processor 202, cause the device 200 to perform the steps for text string embedding or text string authentication as described in the present specification.

**[00108]** Figure 8 shows a block diagram of a computer system 800 suitable for use as the text string embedding device 102, the text string authentication device 112, or the system 100 for text string authentication. The following description of the computer system / computing device 800 is provided by way of example only and is not intended to be limiting.

**[00109]** As shown in Figure 8, the example computing device 800 includes a processor 504 for executing software routines. Although a single processor is shown for the sake of clarity, the computing device 800 may also include a multi-processor system. The processor 804 is connected to a communication infrastructure 806 for communication with other components of the computing device 800. The communication infrastructure 806 may include, for example, a communications bus, cross-bar, or network.

**[00110]** The computing device 800 further includes a main memory 808, such as a random access memory (RAM), and a secondary memory 810. The secondary memory 810 may include, for example, a hard disk drive 812 and/or a removable storage drive 814, which may include a magnetic tape drive, an optical disk drive, or the like. The removable storage drive 814 reads from and/or writes to a removable storage unit 818 in a well-known manner. The removable storage unit 818 may include a magnetic tape, optical disk, or the like, which is read by and written to by removable storage drive 814. As will be appreciated by persons skilled in the relevant art(s), the removable storage unit 818 includes a computer readable storage medium having stored therein computer executable program code instructions and/or data.

**[00111]** In an alternative implementation, the secondary memory 810 may additionally or alternatively include other similar means for allowing computer programs or other instructions to be loaded into the computing device 800. Such means can include, for example, a removable storage unit 822 and an interface 820. Examples of a

removable storage unit 822 and interface 820 include a removable memory chip (such as an EPROM or PROM) and associated socket, and other removable storage units 822 and interfaces 820 which allow software and data to be transferred from the removable storage unit 822 to the computer system 800.

**[00112]** The computing device 800 also includes at least one communication interface 824. The communication interface 824 allows software and data to be transferred between computing device 800 and external devices via a communication path 826. In various embodiments, the communication interface 824 permits data to be transferred between the computing device 800 and a data communication network, such as a public data or private data communication network. The communication interface 824 may be used to exchange data between different computing devices 800 which such computing devices 800 form part an interconnected computer network. Examples of a communication interface 824 can include a modem, a network interface (such as an Ethernet card), a communication port, an antenna with associated circuitry and the like. The communication interface 824 may be wired or may be wireless. Software and data transferred via the communication interface 824 are in the form of signals which can be electronic, electromagnetic, optical or other signals capable of being received by communication interface 824. These signals are provided to the communication interface via the communication path 826.

**[00113]** Optionally, the computing device 800 further includes a display interface 802 which performs operations for rendering images to an associated display 830 and an audio interface 832 for performing operations for playing audio content via associated speaker(s) 834.

**[00114]** As used herein, the term "computer program product" may refer, in part, to removable storage unit 818, removable storage unit 822, a hard disk installed in hard disk drive 812, or a carrier wave carrying software over communication path 826 (wireless link or cable) to communication interface 824. Computer readable storage media refers to any non-transitory tangible storage medium that provides recorded instructions and/or data to the computing device 800 for execution and/or processing. Examples of such storage media include floppy disks, magnetic tape, CD-ROM, DVD, Blu-rayTM Disc, a hard disk drive, a ROM or integrated circuit, USB memory, a magneto-optical disk, or a computer readable card such as a PCMCIA card and the like, whether or not such devices are internal or external of the computing device 800.

Examples of transitory or non-tangible computer readable transmission media that may also participate in the provision of software, application programs, instructions and/or data to the computing device 800 include radio or infra-red transmission channels as well as a network connection to another computer or networked device, and the Internet or Intranets including e-mail transmissions and information recorded on Websites and the like.

**[00115]** The computer programs (also called computer program code) are stored in main memory 808 and/or secondary memory 810. Computer programs can also be received via the communication interface 824. Such computer programs, when executed, enable the computing device 800 to perform one or more features of embodiments discussed herein. In various embodiments, the computer programs, when executed, enable the processor 804 to perform features of the above-described embodiments. Accordingly, such computer programs represent controllers of the computer system 800.

**[00116]** Software may be stored in a computer program product and loaded into the computing device 800 using the removable storage drive 814, the hard disk drive 812, or the interface 820. Alternatively, the computer program product may be downloaded to the computer system 800 over the communications path 826. The software, when executed by the processor 804, causes the computing device 800 to perform functions of embodiments described herein.

**[00117]** It is to be understood that the embodiment of Figure 8 is presented merely by way of example. Therefore, in some embodiments one or more features of the computing device 800 may be omitted. Also, in some embodiments, one or more features of the computing device 800 may be combined together. Additionally, in some embodiments, one or more features of the computing device 800 may be split into one or more component parts.

**[00118]** The techniques described in this specification produce one or more technical effects. In particular, the present disclosure advantageously utilises information of multiple addresses being associated with a shared device to build context information and provide semantic meaning for address words in the addresses. With this information, a plurality of address words and another plurality of address words associated with a same device identifier are considered to have the

same or related semantic and geographic meaning. In addition to addresses, the above observation of the present specification is also applicable to other text strings such as users' names, etc..

**[00119]** In addition or alternative to device identifiers of users' devices, other types of users personal information such as users' phone numbers, users' accounts with the service provider, etc., can also be used to associate with address words or words in other text strings to provide context information for the address words or the words in other text strings.

**[00120]** The present specification reserves the above relationships in a graph to train an embedding model so as to achieve a holistic and accurate text string embedding. Such a text string embedding is then used in an text string authentication described herein. It will be appreciated by those skilled in the art that the text string embedding can also be used for downstream applications such as similar address search, address comparison, etc. Compared to conventional text comparison or word embedding based text string embedding and text string authentication, the present devices, methods and systems produce improved text string embedding accuracy, which in turn enhances text string authentication accuracy in eKYC processes.

**[00121]** It will be appreciated by a person skilled in the art that numerous variations and/or modifications may be made to the present disclosure as shown in the specific embodiments without departing from the scope of the specification as broadly described. The present embodiments are, therefore, to be considered in all respects to be illustrative and not restrictive.

## CLAIMS

What is claimed is:

1. A method of text string embedding comprising:

accessing stored history data comprising text strings and device identifiers, wherein each of the device identifiers is associated with one or more of the text strings; and

generating an embedding look-up table comprising a plurality of keys and a plurality of values, wherein the generating the embedding look-up table comprises:

splitting the text strings in the history data into a plurality of words;

generating a graph with the plurality of words and the device identifiers by:

denoting each word in the plurality of words as a node of a first type;

denoting each of the device identifiers as a node of a second type; and

for each node of the second type, building edges by linking the node of the second type denoting one of the device identifiers to ones of the nodes of the first type, the linked ones of the nodes of the first type denoting words that are comprised in one or more of the text strings associated with the one of the device identifiers;

training an embedding model using the generated graph; and

constructing the embedding look-up table based on the trained embedding model, wherein each of the plurality of keys of the embedding look-up table comprises one of the nodes of the first type and the nodes of the second type, and wherein each of the plurality of values of the embedding look-up table comprises a vector corresponding to the one of the nodes of the first type and the nodes of the second type.

2. The method according to claim 1, wherein the training of the embedding model using the generated graph comprises:

for each node in the generated graph, computing the vector based on information of the node's neighbours, wherein the node's neighbours comprise one or more of nodes linked to the node within a predetermined number of edges;

applying random walk on all the nodes in the generated graph to generate node sequences;

for every node in a node sequence of the generated node sequences, predicting the node's neighbours in the node sequence to form a predicted node distribution; and

training the embedding model based on the predicted node distribution and a true node distribution.

3. The method according to claim 1 or 2, wherein the text strings comprise addresses, and wherein the plurality of words comprise a plurality of address words.

4. The method according to claim 3, wherein when the node's neighbours comprises one or more nodes of the second type, the information of the node's neighbours includes geographic information corresponding to one or more of the device identifiers denoted by the one or more of the nodes of the second type.

5. The method according to claim 2, wherein the computing of the vector for each node in the generated graph further comprises:

if the node is a node of the first type denoting a word that has an appearance frequency lower than a threshold, computing the vector for the node as a vector for unknown words.

6. The method according to any one of claims 1 to 5, wherein the embedding model comprises a general attributed multiplex heterogeneous network embedding (GATNE) model, a hierarchical attention network (HAN) model, or a heterogeneous graph neural network (HetGNN) model.

7. A method of text string authentication comprising:

receiving a first text string from a user;

splitting the first text string into a first plurality of words; and

authenticating the first text string using one or more values in an embedding look-up table generated according to any one of claims 1 to 6 associated with keys in the embedding look-up table corresponding to each of the first plurality of words of the first text string.

8. The method according to claim 7, wherein the authenticating of the first text string comprises:

fetching one or more values in the embedding look-up table associated with keys in the embedding look-up table corresponding to each of the first plurality of words of the

first text string, wherein any unknown word in the first plurality of words of the first text string is mapped to a key for unknown words having a vector for unknown words as a value; and

generating a vector for the first text string by a sum calculation or an average calculation of the fetched values of the first plurality of words.

9. The method according to claim 8, wherein the authenticating of the first text string further comprises:

receiving a second text string of the user from an official source;

splitting the second text string into a second plurality of words;

fetching one or more values in the embedding look-up table associated with keys in the embedding look-up table corresponding to each of the second plurality of words of the second text string, wherein any unknown word in the second plurality of words of the second text string is mapped to the key for unknown words having the vector for unknown words as a value; and

generating a vector for the second text string by a sum calculation or an average calculation of the fetched values of the second plurality of words.

10. The method according to claim 9, further comprising:

authenticating the first text string based on a similarity between the vector for the first text string and the vector for the second text string.

11. A text string embedding device comprising:

a first storage device for storing history data comprising text strings and device identifiers, wherein each of the device identifiers is associated with one or more of the text strings;

a training device coupled to the first storage device for generating an embedding look-up table comprising a plurality of keys and a plurality of values, the training device being configured to:

access the stored history data;

generate the embedding look-up table, wherein the generating the embedding look-up table comprises:

splitting the text strings in the stored history data into a plurality of words;

generating a graph with the plurality of words and the device identifiers by:

denoting each word in the plurality of words as a node of a first type;

denoting each of the device identifiers as a node of a second type; and

for each node of the second type, building edges by linking the node of the second type denoting one of the device identifiers to ones of the nodes of the first type, the linked ones of the nodes of the first type denoting words that are comprised in one or more of the text strings associated with the one of the device identifiers;

training an embedding model using the generated graph; and

constructing the embedding look-up table based on the trained embedding model, wherein each of the plurality of keys of the embedding look-up table comprises one of the nodes of the first type and the nodes of the second type, and wherein each of the plurality of values of the embedding look-up table comprises a vector corresponding to the one of the nodes of the first type and the nodes of the second type; and

a second storage device for storing the embedding look-up table.

12. The device according to claim 11, wherein when training the embedding model using the generated graph, the training device is configured to:

for each node in the generated graph, compute the vector based on information of the node's neighbours, wherein the node's neighbours comprise one or more of nodes linked to the node within a predetermined number of edges;

apply random walk on all the nodes in the generated graph to generate node sequences;

for every node in a node sequence of the generated node sequences, predict the node's neighbours in the node sequence to form a predicted node distribution; and

train the embedding model based on the predicted node distribution and a true node distribution.

13. The device according to claim 11 or 12, wherein the text strings comprise addresses, and wherein the plurality of words comprise a plurality of address words.

14. The device according to claim 13, wherein when the node's neighbours comprises one or more nodes of the second type, the information of the node's neighbours includes

geographic information corresponding to one or more of the device identifiers denoted by the one or more of the nodes of the second type.

15. The device according to claim 12, wherein the training device when computing the vector for each node in the generated graph is further configured to:

if the node is a node of the first type denoting a word that has an appearance frequency lower than a threshold, compute the vector for the node as a vector for unknown words.

16. The device according to any one of claims 11 to 15, wherein the embedding model comprises a general attributed multiplex heterogeneous network embedding (GATNE) model, a hierarchical attention network (HAN) model, or a heterogeneous graph neural network (HetGNN) model.

17. A text string authentication device coupled to a text string embedding device according to any one of claims 11 to 16, the text string embedding device including an embedding look-up table generated according to any one of claims 1 to 6, wherein the text string authentication device is configured to:

receive a first text string;

split the first text string into a first plurality of words; and

authenticate the first text string using one or more values in the embedding look-up table associated with keys in the embedding look-up table corresponding to each of the first plurality of words of the first text string.

18. The device according to claim 17, wherein the text string authentication device is configured to:

fetch one or more values in the embedding look-up table associated with keys in the embedding look-up table corresponding to each of the first plurality of words of the first text string, wherein any unknown word in the first plurality of words of the first text string is mapped to a key for unknown words having a vector for unknown words as a value; and

generate a vector for the first text string by a sum calculation or an average calculation of the fetched values of the first plurality of words.

19. The device according to claim 18, wherein the text string authentication device is further configured to:

receive a second text string of the user from an official source;

split the second text string into a second plurality of words;

fetch one or more values in the embedding look-up table associated with keys in the embedding look-up table corresponding to each of the second plurality of words of the second text string, wherein any unknown word in the second plurality of words of the second text string is mapped to the key for unknown words having the vector for unknown words as a value; and

generate a vector for the second text string by a sum calculation or an average calculation of the fetched values of the second plurality of words.

20. The device according to claim 19, wherein the text string authentication device is further configured to:

authenticate the first text string based on a similarity between the vector for the first text string and the vector for the second text string.