US011539661B2

(12) **United States Patent**
Lai et al.

(10) **Patent No.: US 11,539,661 B2**
(45) **Date of Patent: Dec. 27, 2022**

(54) **USING A LEARNING ALGORITHM TO SUGGEST DOMAIN NAMES**

(71) Applicant: **Go Daddy Operating Company, LLC**, Scottsdale, AZ (US)

(72) Inventors: **Wei-Cheng Lai**, Cupertino, CA (US); **Yu Tian**, Sunnyvale, CA (US); **Wenbo Wang**, Sunnyvale, CA (US); **Chungwei Yen**, Santa Clara, CA (US)

(73) Assignee: **Go Daddy Operating, LLC**, Scottsdale, AZ (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 976 days.

(21) Appl. No.: **16/259,808**

(22) Filed: **Jan. 28, 2019**

(65) **Prior Publication Data**

US 2020/0244622 A1 Jul. 30, 2020

(51) **Int. Cl.**

G06N 20/00	(2019.01)
H04L 61/30	(2022.01)
H04L 61/3015	(2022.01)
G06K 9/62	(2022.01)
G06F 16/33	(2019.01)

(52) **U.S. Cl.**

CPC ***H04L 61/3025*** (2013.01); ***G06F 16/3347*** (2019.01); ***G06K 9/6256*** (2013.01); ***G06N 20/00*** (2019.01)

(58) **Field of Classification Search**

CPC . H04L 61/3025; G06F 16/3347; G06N 20/00; G06K 9/6256

See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,884,295	A *	3/1999	Moll	G01V 3/38 702/14
7,620,613	B1 *	11/2009	Moore	H05K 7/20836 706/14
10,778,640	B1 *	9/2020	Cholleton	G06F 16/951
11,379,715	B2 *	7/2022	Tang	G06N 3/084
2015/0120737	A1 *	4/2015	Trahan	G06Q 10/00 707/738

FOREIGN PATENT DOCUMENTS

WO	WO-2012083371	A1 *	6/2012	G01N 27/00

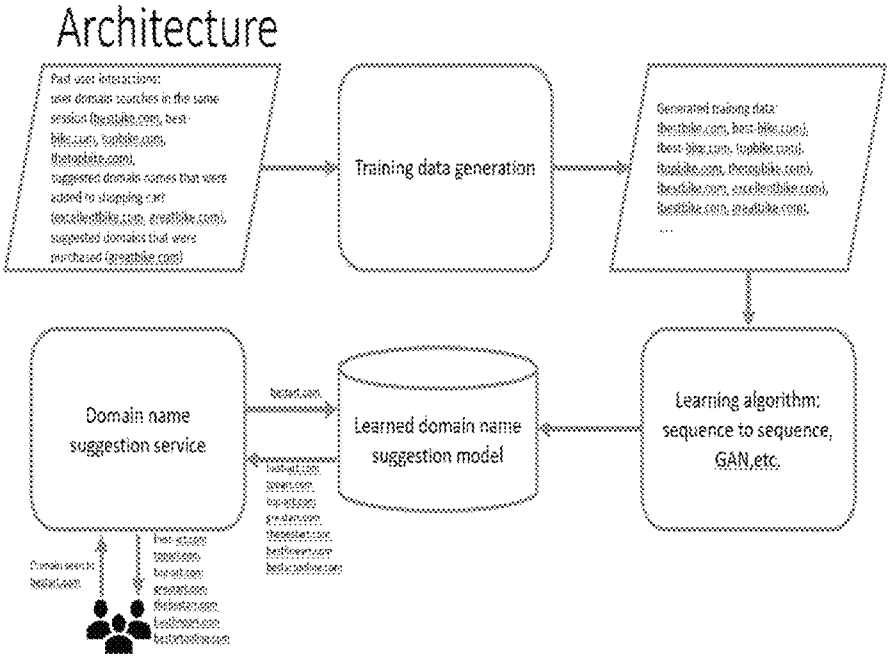
* cited by examiner

Primary Examiner — Paulinho E Smith
(74) *Attorney, Agent, or Firm* — Greenberg Traurig, LLP

(57) **ABSTRACT**

Methods are taught for creating training data for a learning algorithm, training the learning algorithm with the training data and using the trained learning algorithm to suggest domain names to users. A domain name registrar may store activities of a user on a registrar website. Preferably, domain name searches, selected suggested domain names and domain names registered to the user are stored as the training data in a training database. The training data may be stored so that earlier activities act as inputs to the learning algorithm while later activities are the expected outputs of the learning algorithm. Once trained, the learning algorithm may receive activities of other users and suggest domain names to the other users based on their activities.

18 Claims, 19 Drawing Sheets



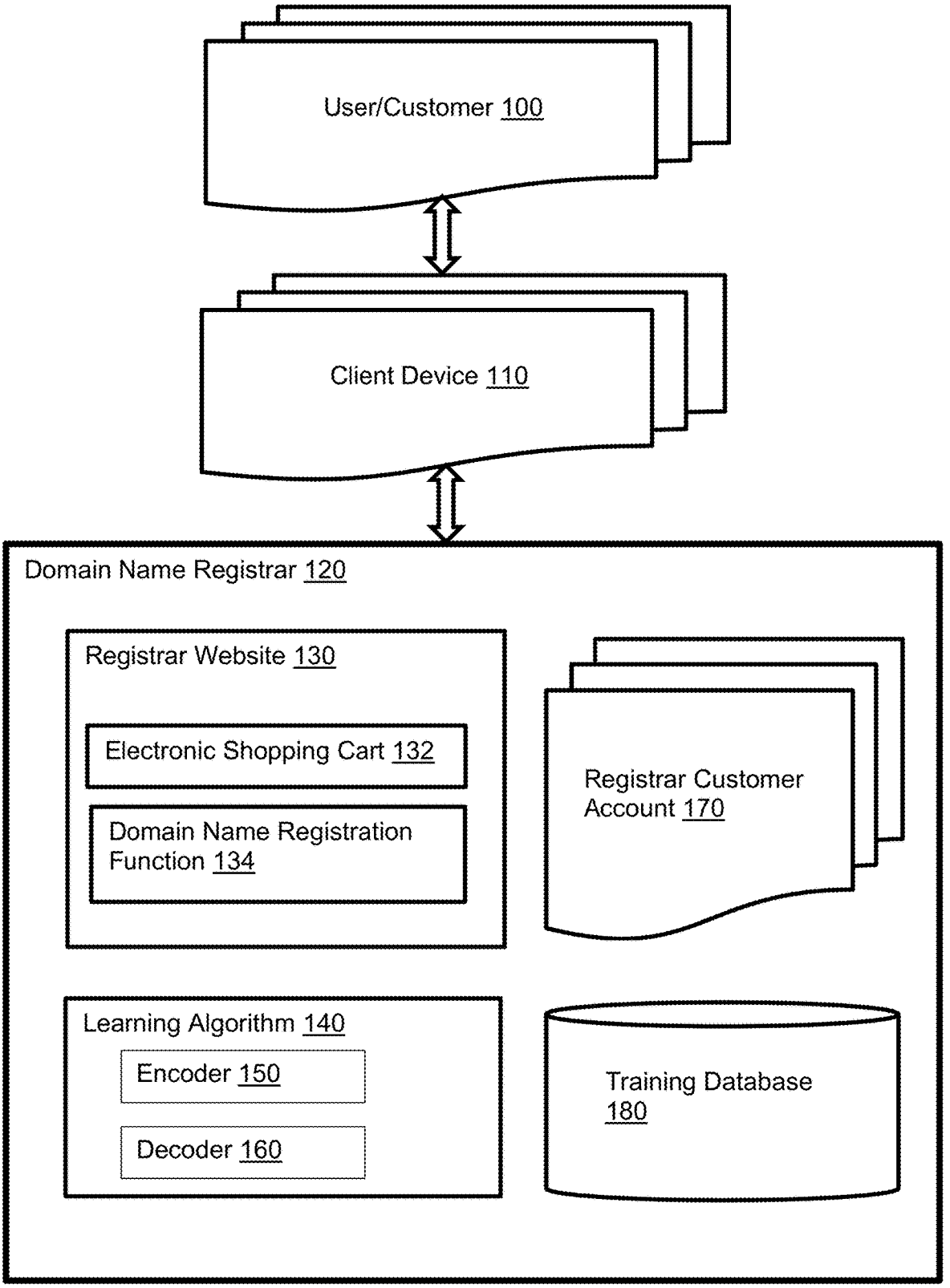

User/Customer 100
Client Device 110
Domain Name Registrar 120
Registrar Website 130
Electronic Shopping Cart 132
Domain Name Registration Function 134
Registrar Customer Account 170
Learning Algorithm 140
Encoder 150
Decoder 160
Training Database 180

FIG. 1

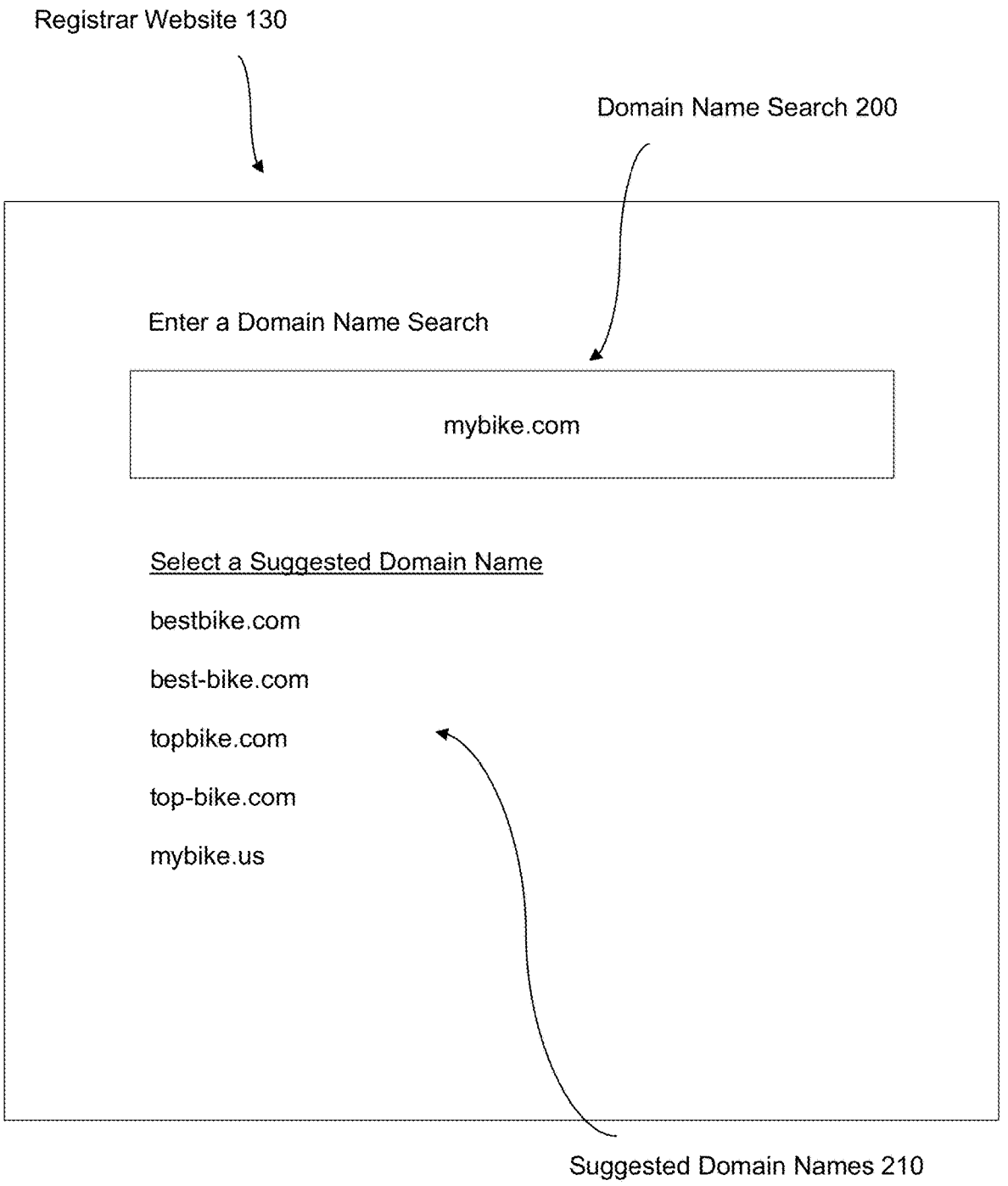

Registrar Website 130
Domain Name Search 200
Enter a Domain Name Search
mybike.com
Select a Suggested Domain Name
bestbike.com
best-bike.com
topbike.com
top-bike.com
mybike.us
Suggested Domain Names 210

FIG. 2

Training Database 180
Input Domain Search 300
Expected Domain Name Suggestion 310
OPTIONAL Related Searches 320
OPTIONAL Related Registered Domain Name(s) 330
mybike.com
bestbike.com
best-bike.com
topbike.com
hotbikes.com
great-bikes.com
new-york-pizza.com
pizza.ny
Italian-food.ny
best-ny-pizza.us
smith-family.com
smithfamily.com
smith-vacation.us
dogbreeders.us
canine.com
Training Data 340

FIG. 3

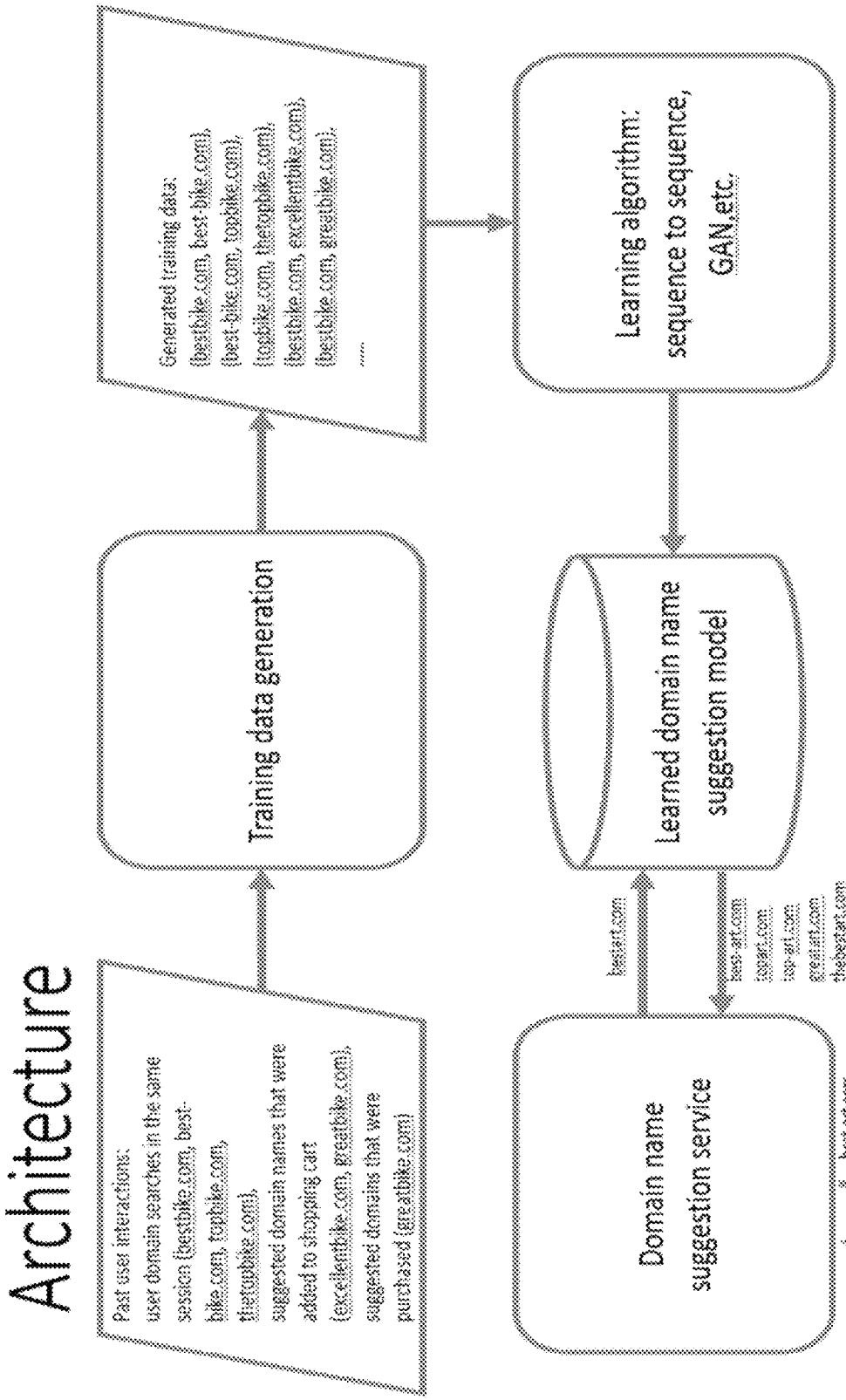

Architecture
Past user interactions:
user domain searches in the same session (bestbike.com, best-bike.com, topbike.com, thetopbike.com),
suggested domain names that were added to shopping cart (excellentbike.com, greatbike.com),
suggested domains that were purchased (greatbike.com)
Training data generation
Generated training data:
(bestbike.com, best-bike.com),
(best-bike.com, topbike.com),
(topbike.com, thetopbike.com),
(bestbike.com, excellentbike.com),
(bestbike.com, greatbike.com),
.....
Learning algorithm:
sequence to sequence,
GAN,etc.
Learned domain name suggestion model
Domain name suggestion service
bestart.com
best-art.com
toart.com
top-art.com
greatart.com
thebestart.com
bestfineart.com
bestartonline.com
Domain search:
bestart.com

FIG. 4

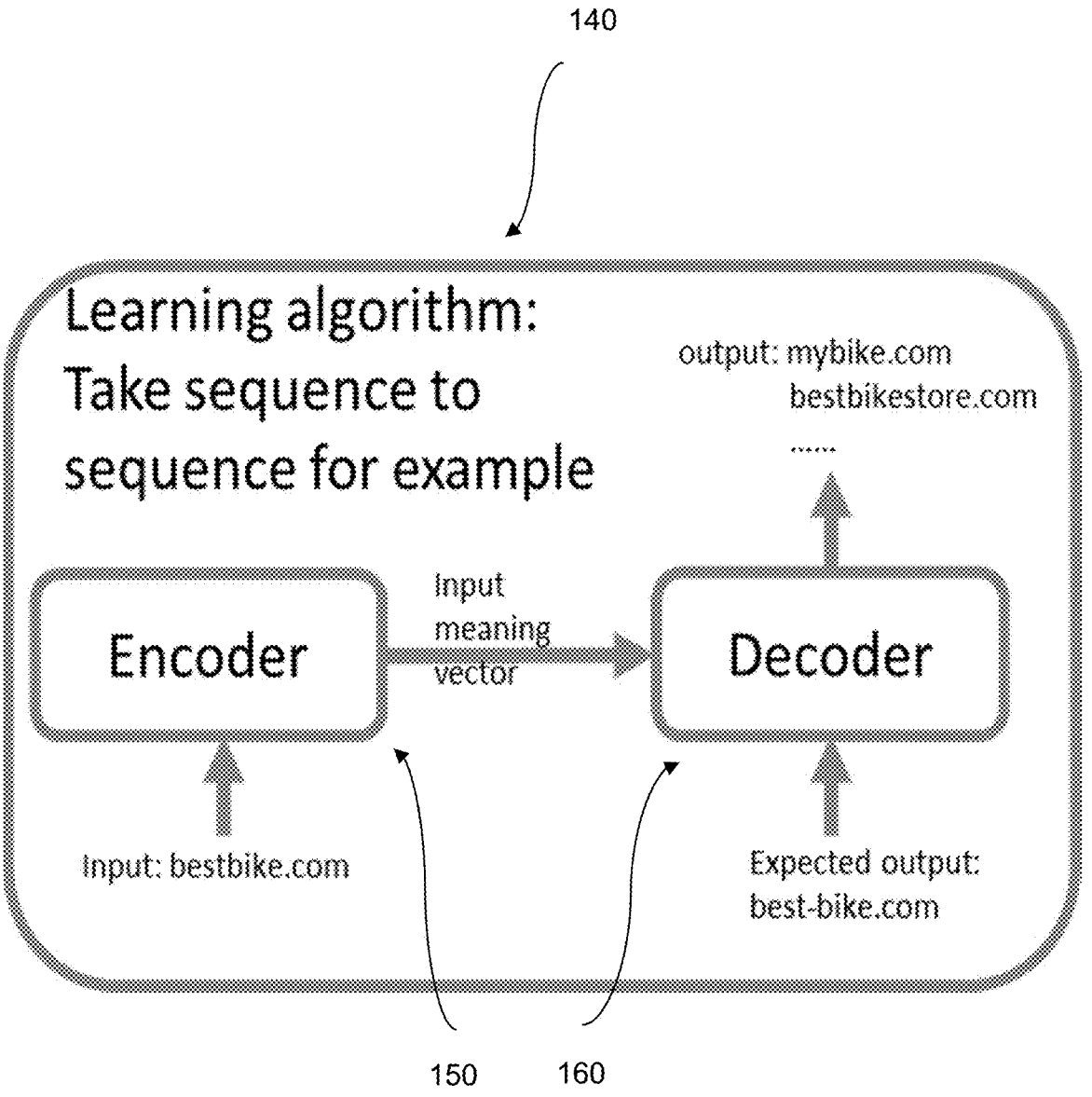

140
Learning algorithm:
Take sequence to
sequence for example
output: mybike.com
bestbikestore.com
.....
Encoder
Input
meaning
vector
Decoder
Input: bestbike.com
Expected output:
best-bike.com
150
160

FIG. 5

Learning algorithm 1: sequence-to-sequence neural networks implementation

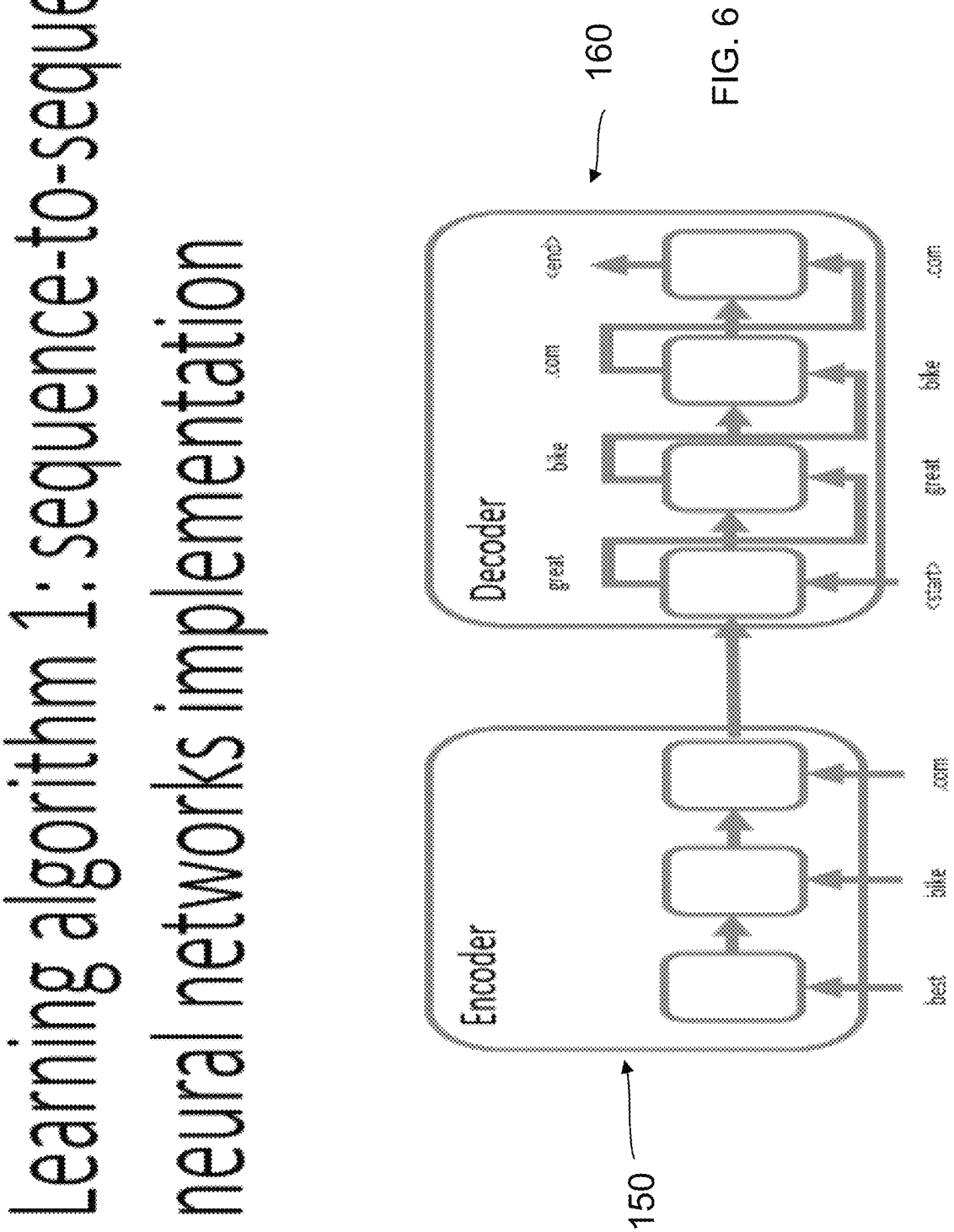


FIG. 6

Learning algorithm 2: generative adversarial networks (GAN) implementation

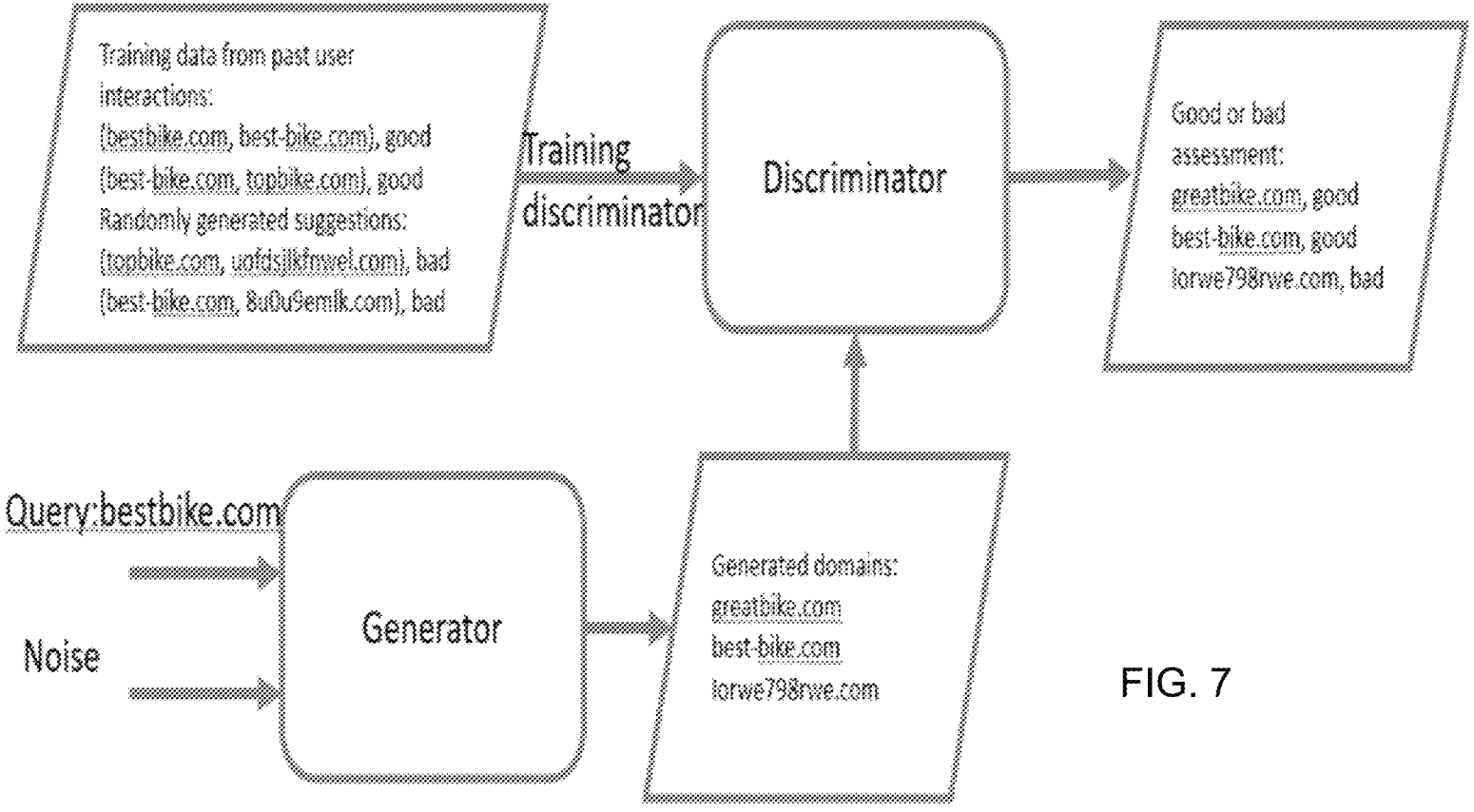


FIG. 7

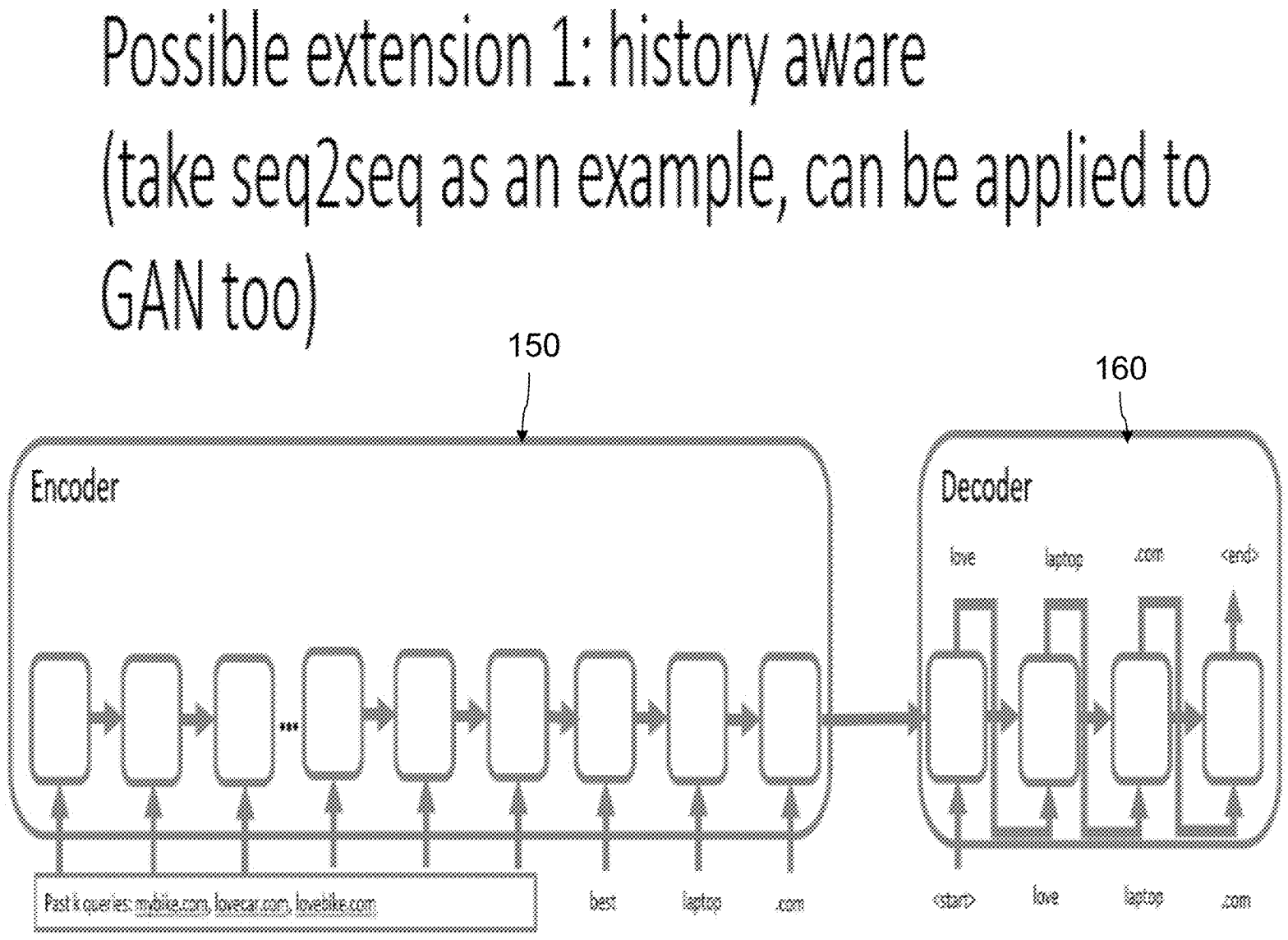

Possible extension 1: history aware
(take seq2seq as an example, can be applied to GAN too)
150
160
Encoder
Decoder
love
laptop
.com
<end>
Past k queries: mybike.com, lovecar.com, lovebike.com
best
laptop
.com
<start>
love
laptop
.com

FIG. 8

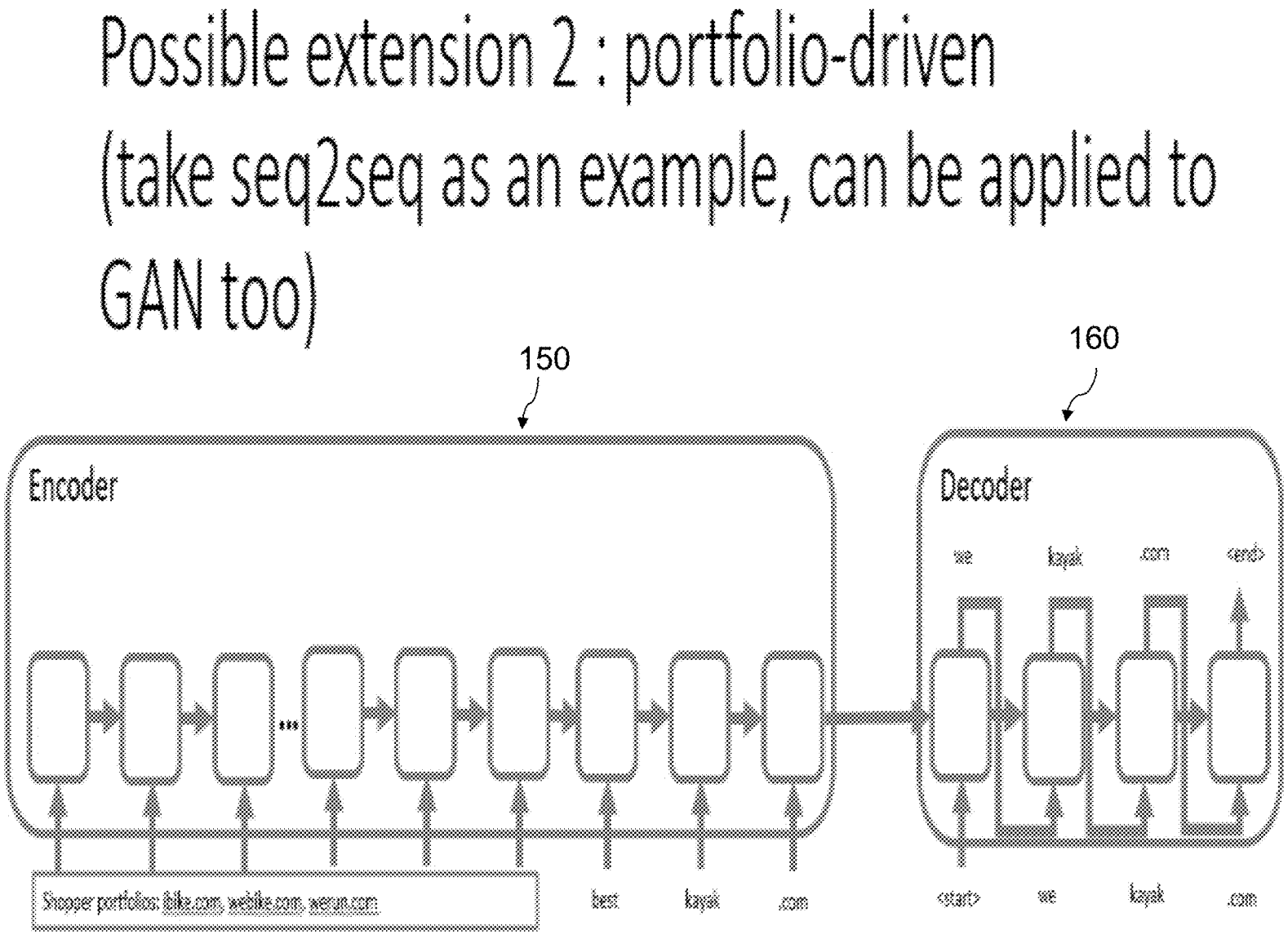

Possible extension 2 : portfolio-driven
(take seq2seq as an example, can be applied to
GAN too)
150
160
Encoder
Decoder
we
kayak
.com
<end>
Shopper portfolios: ibike.com, webike.com, werun.com
best
kayak
.com
<start>
we
kayak
.com

FIG. 9

Neural Network (NN)

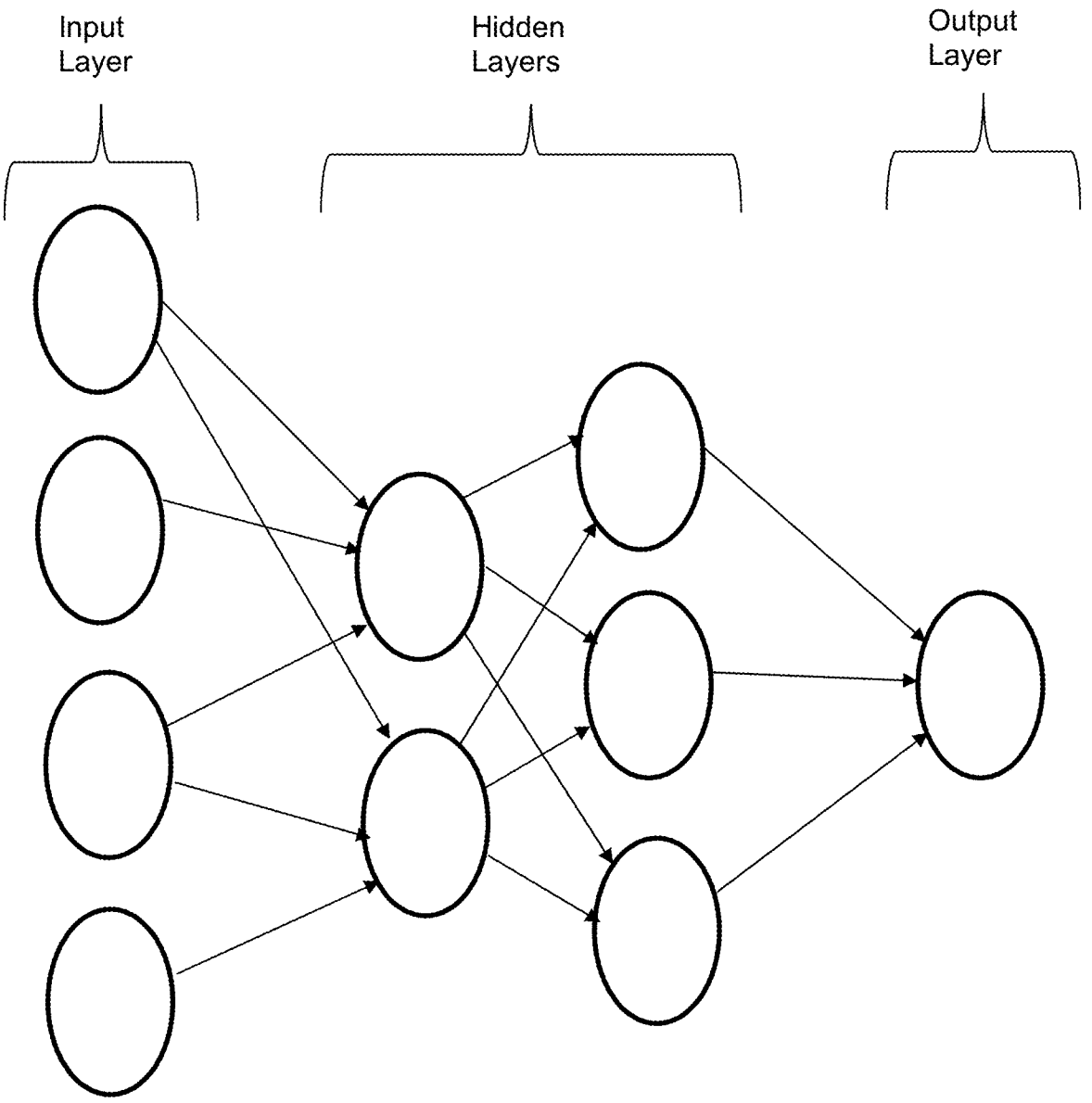


FIG. 10

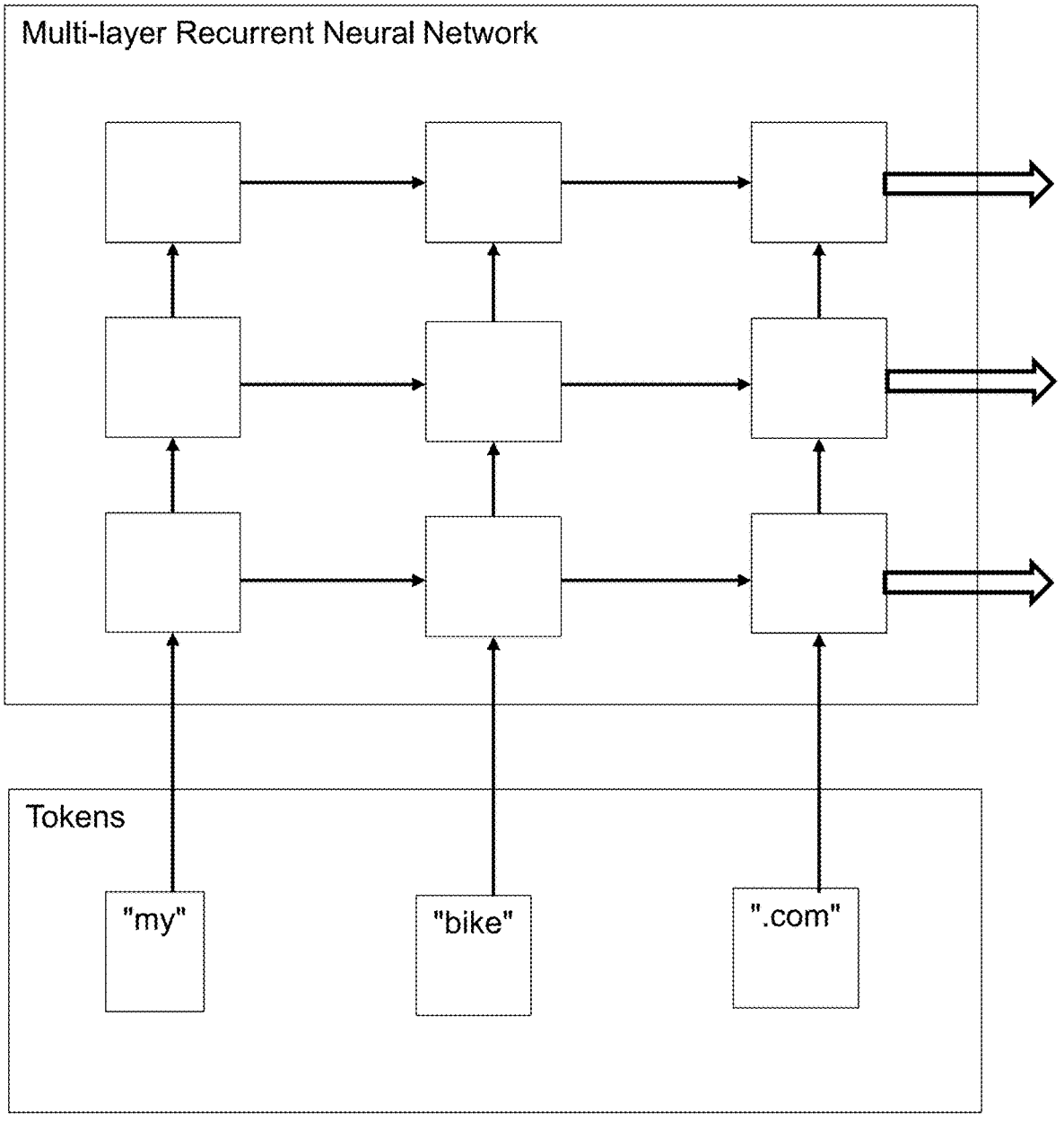


FIG. 11

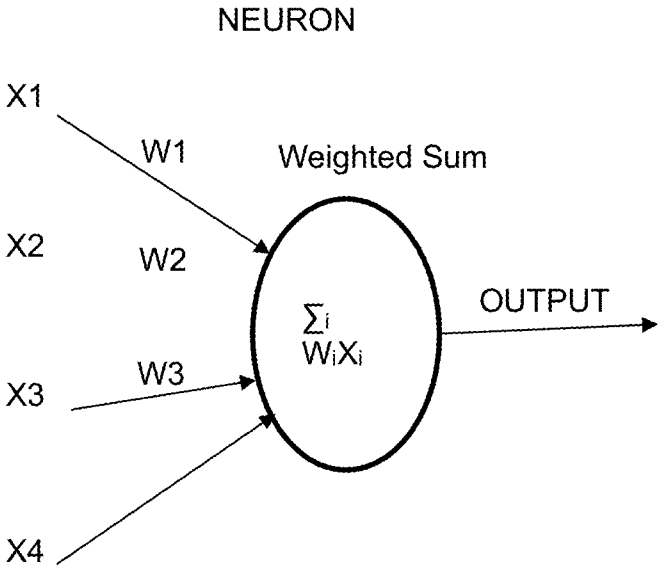


FIG. 12

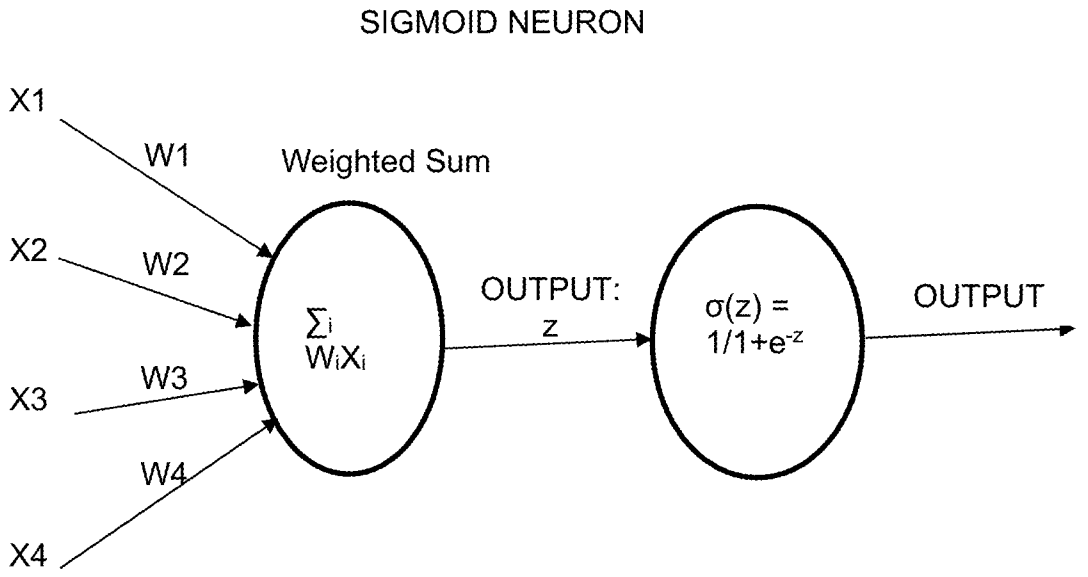

SIGMOID NEURON
X1
W1
Weighted Sum
X2
W2
OUTPUT:
z
σ(z) =
1/1+e-z
OUTPUT
W3
X3
W4
X4

FIG. 13

Creating Training Data for a Learning Algorithm to Suggest a Domain Name

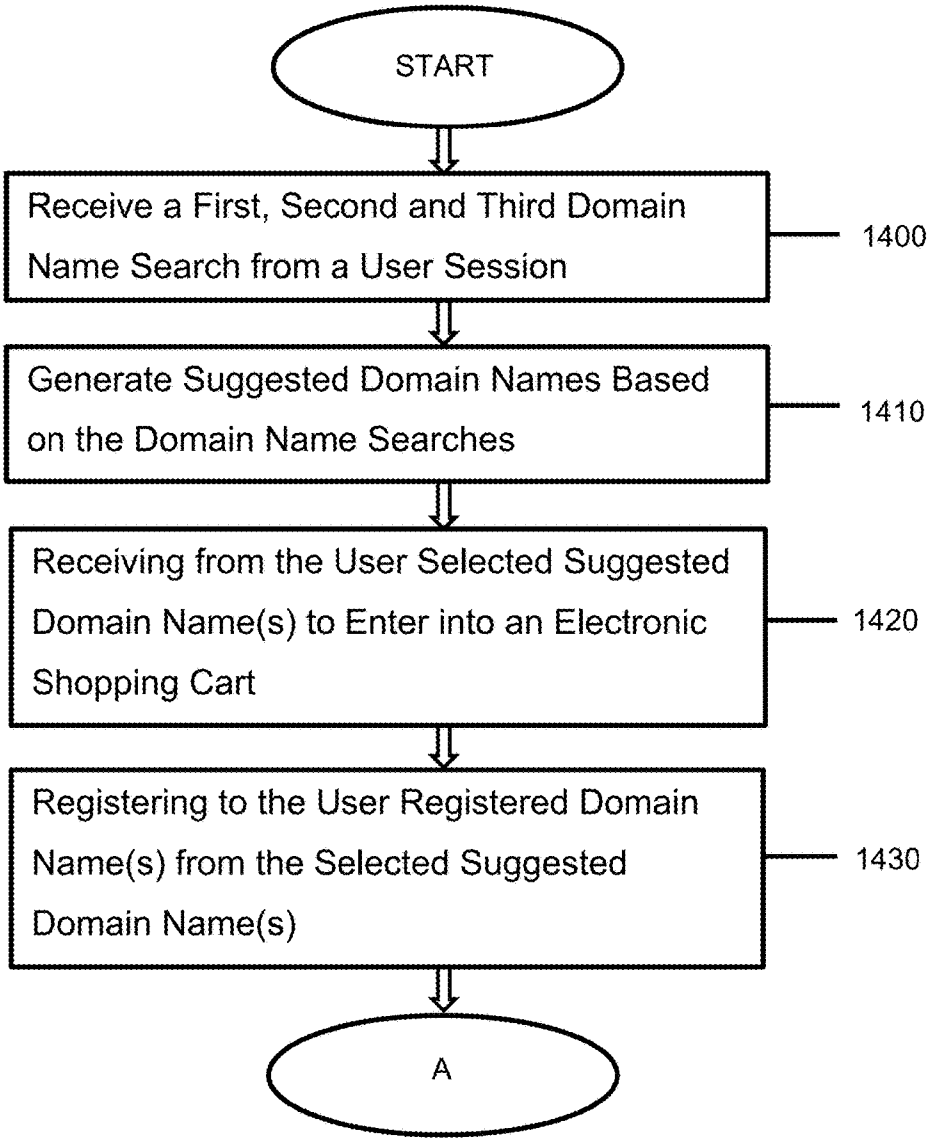


FIG. 14

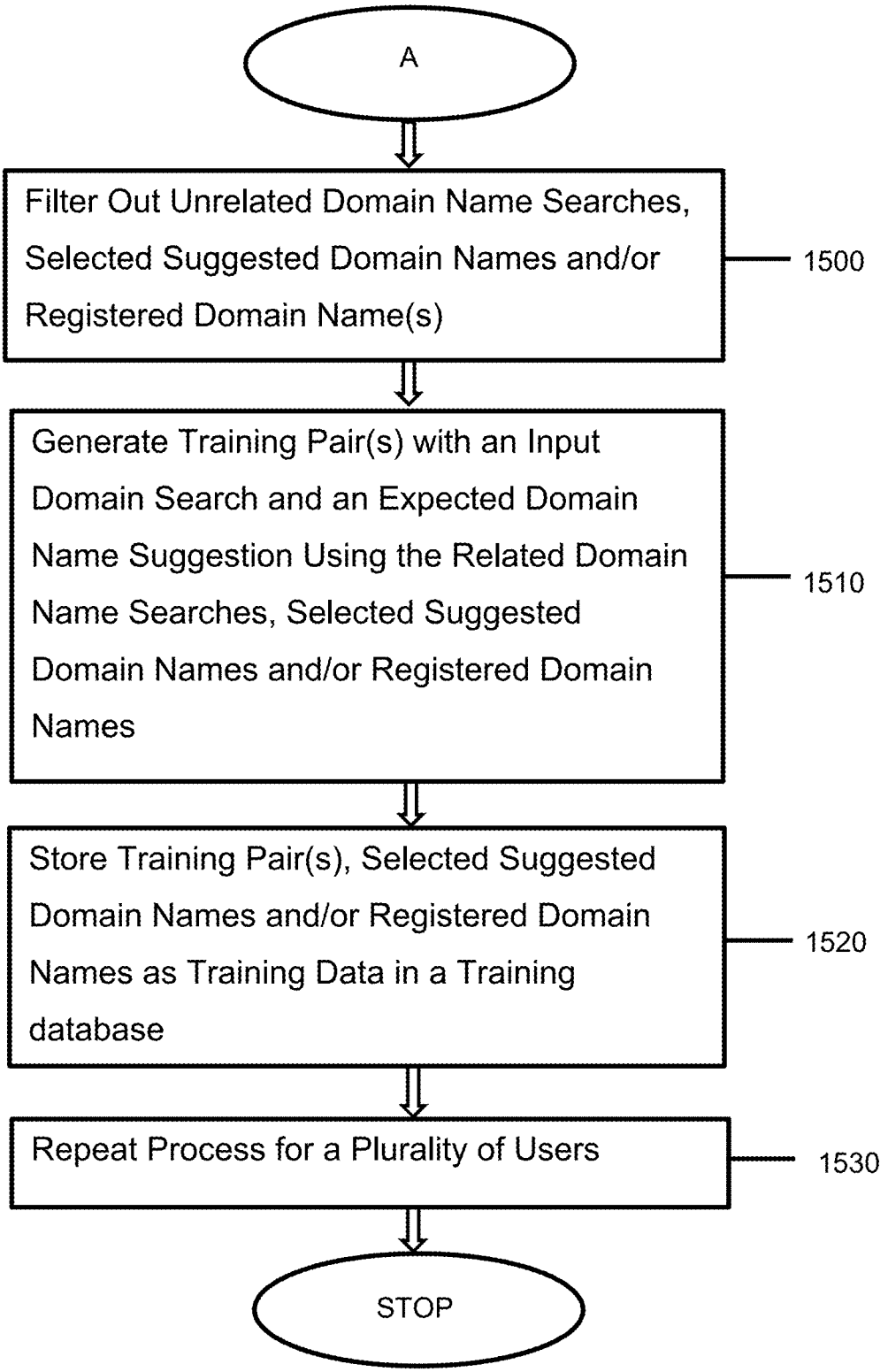

A
Filter Out Unrelated Domain Name Searches, Selected Suggested Domain Names and/or Registered Domain Name(s)
1500
Generate Training Pair(s) with an Input Domain Search and an Expected Domain Name Suggestion Using the Related Domain Name Searches, Selected Suggested Domain Names and/or Registered Domain Names
1510
Store Training Pair(s), Selected Suggested Domain Names and/or Registered Domain Names as Training Data in a Training database
1520
Repeat Process for a Plurality of Users
1530
STOP

FIG. 15

Training a Learning Algorithm to Suggest a Domain Name

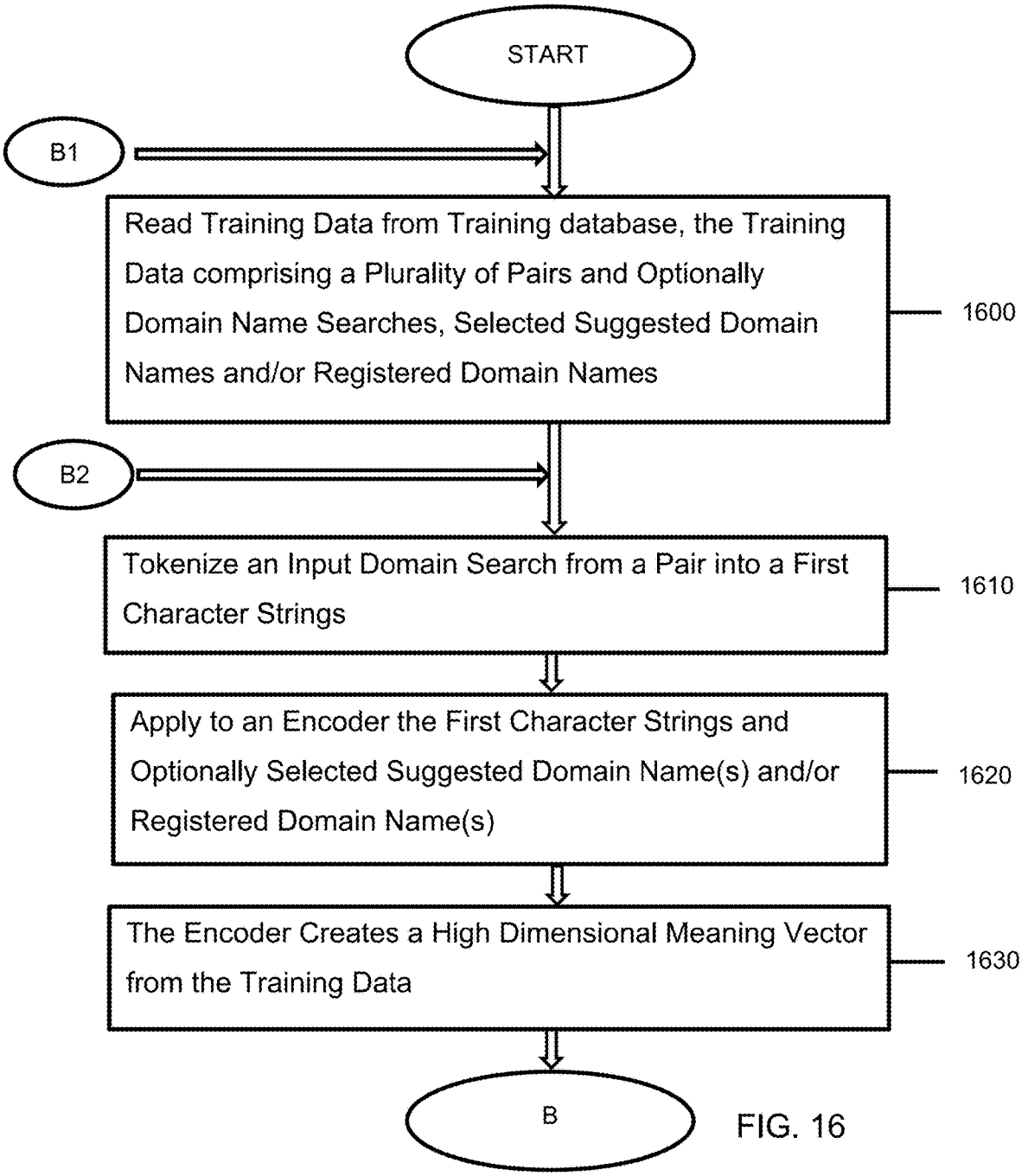


FIG. 16

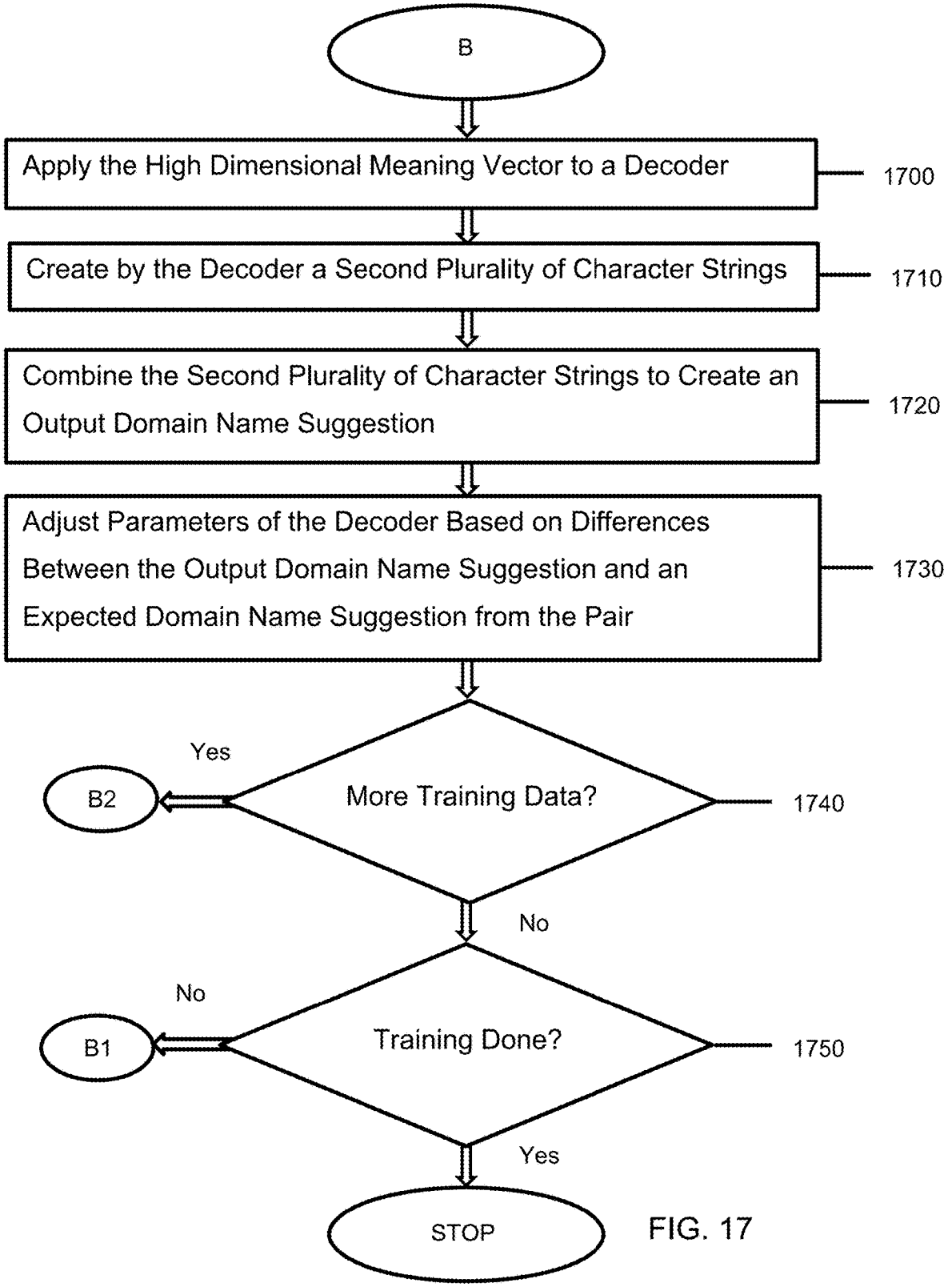

B
Apply the High Dimensional Meaning Vector to a Decoder
1700
Create by the Decoder a Second Plurality of Character Strings
1710
Combine the Second Plurality of Character Strings to Create an Output Domain Name Suggestion
1720
Adjust Parameters of the Decoder Based on Differences Between the Output Domain Name Suggestion and an Expected Domain Name Suggestion from the Pair
1730
More Training Data?
1740
Yes
B2
No
Training Done?
1750
No
B1
Yes
STOP

FIG. 17

Using a Learning Algorithm to Suggest a Domain Name

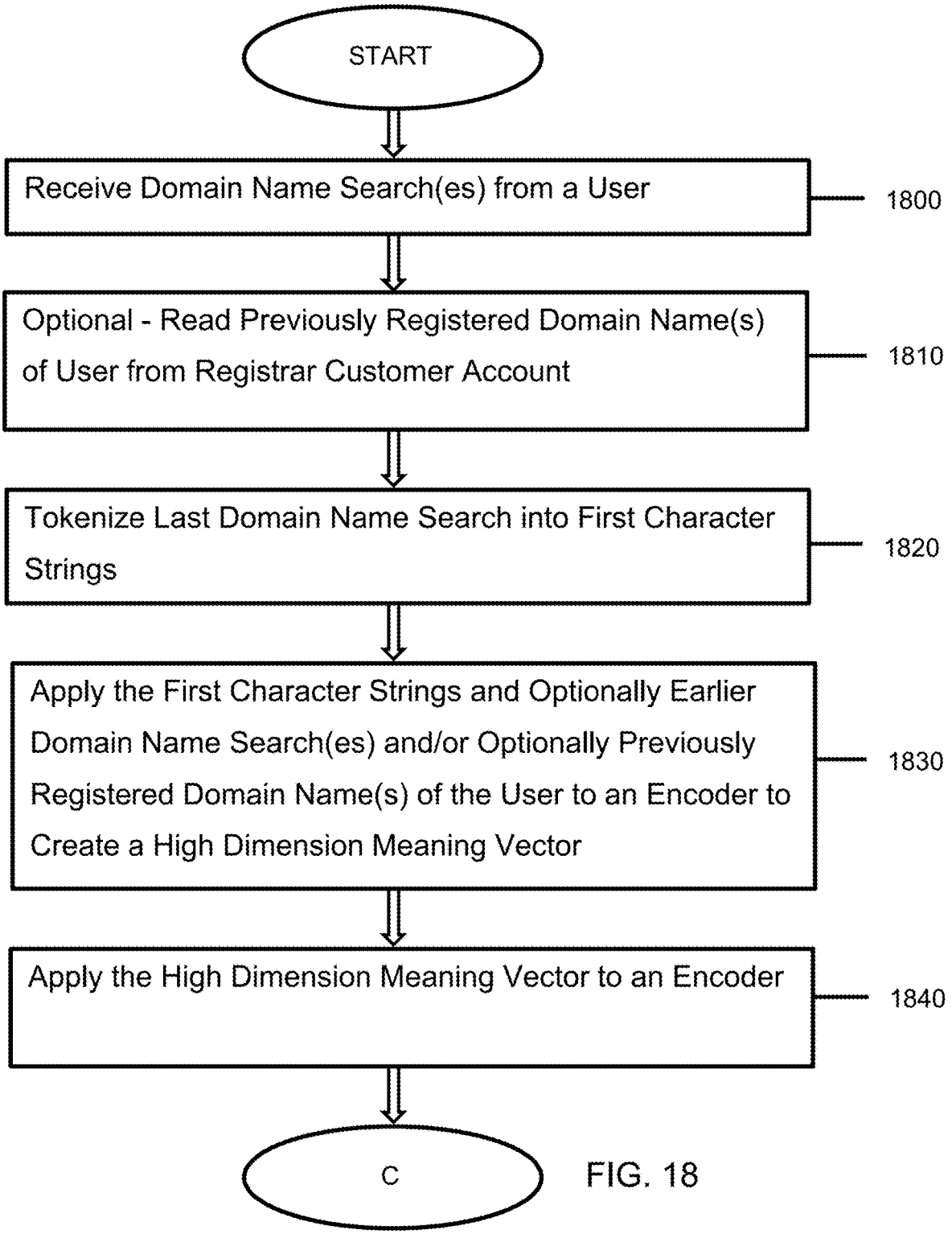


FIG. 18

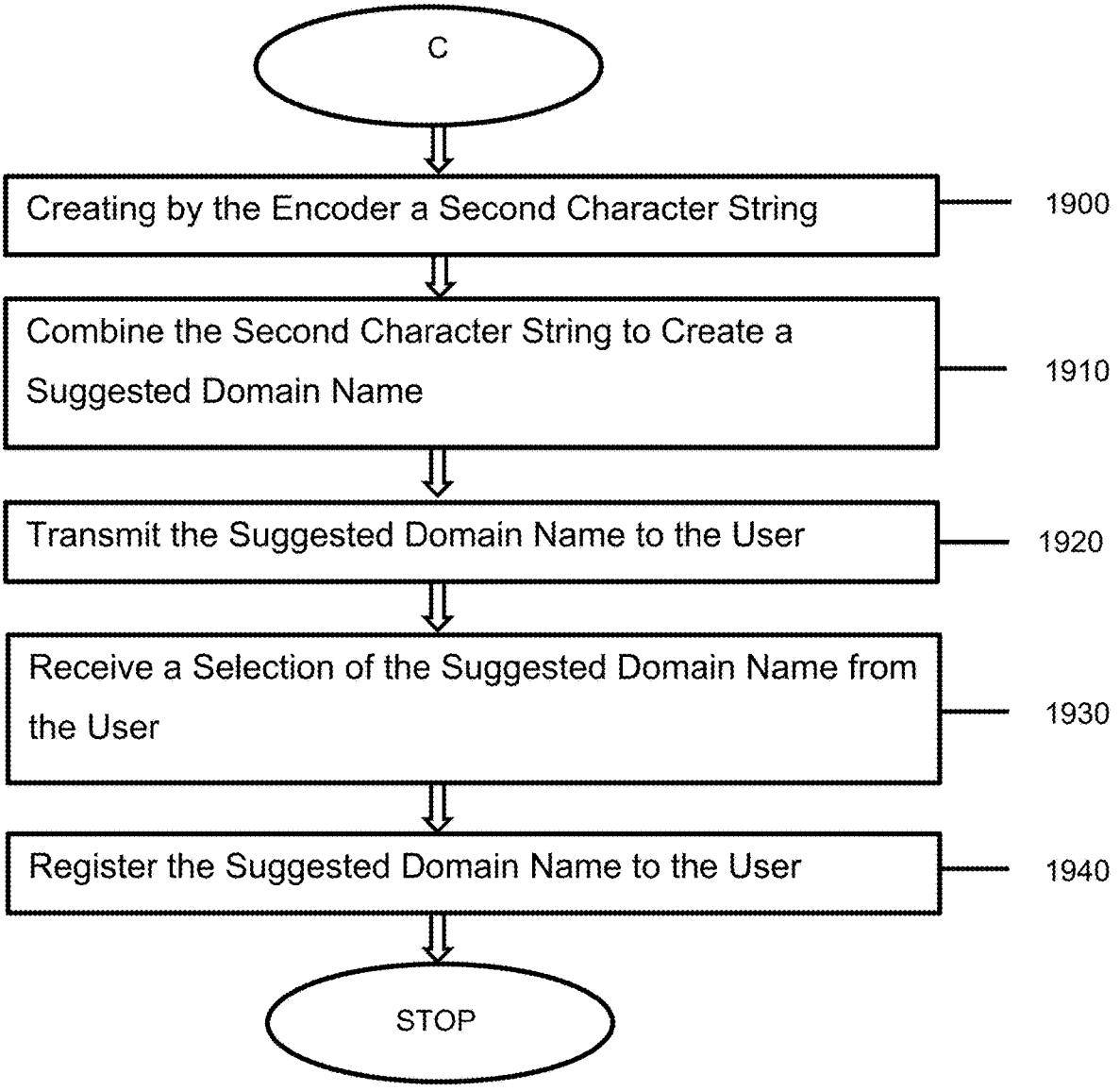

C
Creating by the Encoder a Second Character String
1900
Combine the Second Character String to Create a Suggested Domain Name
1910
Transmit the Suggested Domain Name to the User
1920
Receive a Selection of the Suggested Domain Name from the User
1930
Register the Suggested Domain Name to the User
1940
STOP

FIG. 19

USING A LEARNING ALGORITHM TO SUGGEST DOMAIN NAMES

FIELD OF THE INVENTION

The present invention generally relates to the field of creating training data, using domain name searches, selected suggested domain names and/or registered domain names, to train a learning algorithm to suggest domain names to users.

SUMMARY OF THE INVENTION

The present invention provides methods for 1) creating training data for a learning algorithm to suggest a domain name to a user using training data gathered from the activities of a plurality of users searching for domain names; 2) training the learning algorithm on the created training data to suggest a domain name; and 3) suggesting a domain name to a user using the learning algorithm trained on the training data.

Creating Training Data

A domain name registrar may collect training data from activities of a plurality of different users (who may be new visitors and/or returning customers). The users, particularly if they are returning customers, may have a registrar customer account with the domain name registrar. The domain name registrar may log the activities of the users that visit a registrar website operated by the domain name registrar. The domain name registrar may store specific activities of the users/customers in a training database. In general, earlier activities of the user are stored in the training database as inputs to a learning algorithm, while later activities are stored in the training database as expected outputs of the learning algorithm. This trains the learning algorithm to predict a desired future action (possibly suggesting a desired domain name) based on a given a current action (a user entered current domain name search).

As a specific example, a user may visit a registrar website and enter a first domain name search into a field on a webpage of the registrar website. The domain name registrar may, using any desired technique, spin the first domain name search into a first plurality of suggested domain names. The user might or might not select one or more of the suggested domain names for entry into an electronic shopping cart. The user might or might not purchase and register one or more of the selected suggested domain names in the electronic shopping cart. These activities may be monitored and stored for future analysis.

The user may repeat this process one or more times over the course of a single continuous login session. In other words, the user may repeat the process of entering additional domain name searches, receiving additional suggested domain names, selecting (or not selecting) one or more of the suggested domain names and registering (or not registering) one or more of the selected suggested domain names.

In a preferred embodiment, the domain name searches, selected suggested domain names and/or registered domain names are all evaluated to verify they are related to a common search to prevent mixing data, for example, from the user's professional or commercial activities with the user's personal or family activities. This may be performed, as an example, by tokenizing the domain name search(es), selected suggested domain name(s) and/or registered domain name(s) and only counted as related (and thus combined together and stored in the training base) if they share at least one token or a synonym of the one token. This filters out unrelated data and prevents the training data from combining an input related to a first topic with an expected output from a different second topic.

Data entered by the user representing domain name search(es), selected suggested domain name(s) and/or registered domain name(s) entered in a single continuous login session of the user are preferably stored in the training database in predefined allowable formats. As a non-limiting example of an allowable format, a first domain name search may be an input domain search and a second domain name search may be an expected domain name suggestion. Please note that the first domain name search was entered before the second domain name search.

As another non-limiting example of an allowable format, the second domain name search may be an input domain name search and a third domain name search may be an expected domain name suggestion. Again, the input domain name search, i.e., the second domain name search, was entered before the expected domain name suggestion, i.e., the third domain name search.

As another non-limiting example of an allowable format, the first domain name search may be an input domain search and a selected suggested domain name may be an expected domain name suggestion. As before, the first domain name search was entered before a suggested domain name was selected by the user. When the user activities are occurring, the domain name registrar may tag every activity with a time stamp to help insure the training data is kept in the general format of earlier activities being the input to the learning algorithm, while later activities are the expected output of the learning algorithm.

As another non-limiting example, the first domain name search may be an input domain search and a registered domain name may be an expected domain name suggestion. These formats are desirable as the input domain search (entered earlier in time by the user) leads to the expected domain name suggestion (produced later in time). Thus, the learning algorithm will take as input the input domain search and be trained to produce the expected domain name suggestion. The training data is preferably stored in the training database in one or more of these allowable formats.

Data entered by the user representing domain name search(es), selected suggested domain name(s) and/or registered domain name(s) should not be stored in the training database or used to train the learning algorithm in predefined unallowable formats. As a non-limiting example of an unallowable format, a second domain name search should not be stored as an input domain search with a first domain name search being stored as an expected domain name suggestion.

As another non-limiting example of an unallowable format, a third domain name search should not be stored as an input domain name search with the second domain name search being stored as an expected domain name suggestion. Again, this would incorrectly store a later activity as an input with an earlier activity as an expected output.

As another non-limiting example of an unallowable format, a selected suggested domain name should not be stored as an input domain search with a domain name search stored as an expected domain name suggestion.

As another non-limiting example of an unallowable format, a registered domain name should not be stored as an input domain search with a domain name search being stored as an expected domain name suggestion.

These formats are not desirable as the input domain search (produced later in time by the user) leads to the

expected domain name suggestion (entered earlier in time). This is in reverse order from the desired direction of the learning algorithm.

The above process of collecting data may be repeated for a plurality of other users and/or the same user during a different continuous login session.

Training a Learning Algorithm

Learning algorithms are trained using training data. A domain name registrar may read the training data from the training database. The training data preferably comprises an input and an expected output. The input to the learning algorithm may be an input domain name search (preferably tokenized to words) and optionally other domain name search(es), selected suggested domain name(s) and/or registered domain name(s) of the user. If a piece of user activity data is missing, a default value may be used. The expected output may be an expected domain name suggestion.

The training involves applying the input to an encoder of the learning algorithm. The encoder creates a high dimension meaning vector which is applied to a decoder of the learning algorithm. An output of the decoder of the learning algorithm is compared with the expected output as stored in the training data. Using any desired technique, such as, as non-limiting examples, gradient descent and/or back propagation, the parameters of the encoder of the learning algorithm are adjusted so that the input is more likely to produce the expected output in future runs.

The training process of the learning algorithm may be repeated using all (or some given portion) of the training data. A full-cycle of using all (or some desired portion) of the training data may be referred to as an epoch. Additional epochs, i.e., full-cycles of using all (or some given portion) of the training data, may be performed until the learning algorithm has been trained to a desired level. If parts of the training data are not used for training, these unused parts of the training data may be used to check the progress of the training of the learning algorithm to determine if the learning algorithm has been trained to a desired level.

Using a Learning Algorithm to Produce a Suggested Domain Name

A domain name registrar may collect activity data regarding the activities of a user on the registrar website, such as one or more domain name searches entered by the user, one or more suggested domain names selected and entered into an electronic database and/or one or more registered domain names in a registrar customer account of the user.

The registrar may apply the most recent domain name search, preferably tokenized to words, to an encoder of the learning algorithm (now trained using the created training database). In other embodiments, one or more other domain name searches, one or more selected suggested domain names in the electronic database and/or one or more domain names registered to the user are also applied to the learning algorithm as inputs to the encoder of the learning algorithm. If a piece of user activity data is missing, a default value may be used. It should be appreciated that the types of input data used to train the learning algorithm should match and be the same as when actually using the learning algorithm to produce a suggested domain name.

The encoder may create a high dimensional meaning vector from the input and apply the high dimensional meaning vector to a decoder of the learning algorithm. The decoder may convert the high dimensional meaning vector to one or more character strings which may be used to create a suggested domain name. The suggested domain name may be transmitted to the user and, if selected by the user for registration, registering the domain name to the user. In other embodiments, the suggested domain name may be spun into additional suggested domain names which may also be transmitted to the user for selection and possible registration.

The above features and advantages of the present invention will be better understood from the following detailed description taken in conjunction with the accompanying drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. **1** is a block diagram illustrating that a user/customer may use a client device connected, preferably through the Internet, to a domain name registrar. The domain name registrar comprises a registrar website, a learning algorithm, registrar customer accounts and a training database.

FIG. **2** is an illustration of a simple example registrar website. In this simple example website there is a field for entering a domain name search and a place where a plurality of suggested domain names may be listed. The user may select one or more of the suggested domain names, thereby entering the selected suggested domain names into an electronic shopping cart. The user may complete the purchase/registration of one or more of the selected suggested domain names in the electronic shopping cart so that the selected suggested domain names are registered to the user.

FIG. **3** is an illustration of example training data that may be stored in a training database. The training data comprises a pair, which comprises an input domain search and an expected domain name suggestion. The training data may also store one or more searches, related to the associated pair, and/or one or more domain names, also related to the associated pair.

FIG. **4** is an illustration of the Architecture showing the overall process of collecting user activity data, creating and storing training data, training a learning algorithm to suggest a domain name and a user using the learning algorithm to receive a suggested domain name.

FIG. **5** is an example illustration of a learning algorithm that comprises an encoder and a decoder.

FIG. **6** is an example illustration of a learning algorithm with a sequence-to-sequence neural network implementation.

FIG. **7** is an example illustration of a learning algorithm with a generative adversarial network (GAN) implementation.

FIG. **8** is an example illustration of a learning algorithm that is history aware in that the encoder of the learning algorithm receives as inputs earlier related domain name searches of the user.

FIG. **9** is an example illustration of a learning algorithm that is portfolio-driven in that the encoder of the learning algorithm receives as inputs related registered domain names of the user

FIG. **10** is a diagram of a possible arrangement of nodes, i.e., artificial neurons, for a learning algorithm that comprises an Artificial Neural Network (ANN), a Recurrent Neural Network (RNN) and/or a Neural Network (NN). The learning algorithm may comprise an input layer having one or more nodes, one or more hidden layers, with each hidden layer comprising one or more nodes and an output layer having one or more nodes. As illustrated, information is allowed to flow forward (left to right in the diagram), but is preferably prevented from flowing backwards through the nodes.

FIG. **11** is a diagram of a learning algorithm that is a multi-layered RNN that illustrates how different character

strings (encoded into a high dimension meaning vector) may be feed into this multi-layered RNN.

FIG. **12** is a diagram of a neuron, one type of node or artificial neuron that may be used to create a learning algorithm.

FIG. **13** is a diagram of a sigmoid neuron, another type of node or artificial neuron, that may be used to create a learning algorithm.

FIGS. **14** and **15** are flowcharts illustrating a process of creating a training data that may be used to train a learning algorithm to suggest domain names.

FIGS. **16** and **17** are flowcharts illustrating a process of training a learning algorithm using training data.

FIGS. **18** and **19** are flowcharts illustrating a process of using a trained learning algorithm to suggest a domain name to a user for registration.

DETAILED DESCRIPTION

The present inventions will now be discussed in detail with regard to the attached drawing figures that were briefly described above. In the following description, numerous specific details are set forth illustrating the Applicant's best mode for practicing the invention and enabling one of ordinary skill in the art to make and use the invention. It will be obvious, however, to one skilled in the art that the present invention may be practiced without many of these specific details. In other instances, well-known machines, structures, and method steps have not been described in particular detail in order to avoid unnecessarily obscuring the present invention. Unless otherwise indicated, like parts and method steps are referred to with like reference numerals.

A computer network is a collection of links and nodes (e.g., multiple computers and/or other client devices connected together) arranged so that information may be passed from one part of the computer network to another over multiple links and through various nodes. Examples of computer networks include the Internet, the public switched telephone network, the global Telex network, computer networks (e.g., an intranet, an extranet, a local-area network, or a wide-area network), wired networks, and wireless networks.

The Internet is a worldwide network of computers and computer networks arranged to allow the easy and robust exchange of information between computer users on client devices. Hundreds of millions of people around the world have access to client devices connected to the Internet via Internet Service Providers (ISPs). Content providers place multimedia information (e.g., text, graphics, audio, video, animation, and other forms of data) at specific locations on the Internet referred to as websites. The combination of all the websites and their corresponding web pages on the Internet is generally known as the World Wide Web (VVWW) or simply the Web.

For Internet users and businesses alike, the Internet continues to be increasingly valuable. More people use the Web for everyday tasks, from social networking, shopping, banking, and paying bills to consuming media and entertainment. E-commerce is growing, with businesses delivering more services and content across the Internet, communicating and collaborating online, and inventing new ways to connect with each other.

Prevalent on the Internet are multimedia websites, some of which may offer and sell goods and services to individuals and organizations. Websites may consist of a single webpage, but typically consist of multiple interconnected and related webpages. Websites, unless very large and complex or have unusual traffic demands, typically reside on a single server and are prepared and maintained by a single individual or entity (although websites residing on multiple servers is certainly possible). Menus, links, tabs, etc. may be used to move between different web pages within the website or to move to a different website.

Websites may be created using HyperText Markup Language (HTML) to generate a standard set of tags that define how the webpages for the website are to be displayed. Websites may comprise titles, tags and text. Users of the Internet may access content providers' websites using software known as an Internet browser, such as MICROSOFT INTERNET EXPLORER or MOZILLA FIREFOX. After the browser has located the desired webpage, it requests and receives information from the webpage, typically in the form of an HTML document, and then displays the webpage content for the user on the client device. The user may then view other webpages at the same website or move to an entirely different website using the browser.

Some Internet users, typically those that are larger and more sophisticated, may provide their own hardware, software, and connections to the Internet. Many Internet users either do not have the resources available or do not want to create and maintain the infrastructure necessary to host their own websites. To assist such individuals (or entities), hosting companies exist that offer website hosting services. These hosting providers typically provide the hardware, software, and electronic communication means necessary to connect multiple websites to the Internet. A single hosting provider may literally host thousands of websites on one or more hosting servers.

Browsers are able to locate specific websites because each website, resource, and computer on the Internet has a unique Internet Protocol (IP) address. Presently, there are two standards for IP addresses. The older IP address standard, often called IP Version 4 (IPv4), is a 32-bit binary number, which is typically shown in dotted decimal notation, where four 8-bit bytes are separated by a dot from each other (e.g., 64.202.167.32). The notation is used to improve human readability. The newer IP address standard, often called IP Version 6 (IPv6), is a 128-bit binary number. The standard human readable notation for IPv6 addresses presents the address as eight 16-bit hexadecimal words, each separated by a colon (e.g., 2EDC:BA98:0332:0000:CF8A:000C:2154:7313).

IP addresses, however, even in human readable notation, are difficult for people to remember and use. A Uniform Resource Locator (URL) is much easier to remember and may be used to point to any computer, directory, or file on the Internet. A browser is able to access a website on the Internet through the use of a URL. The URL may include a Hypertext Transfer Protocol (HTTP) request combined with the website's Internet address, also known as the website's domain name. An example of a URL with a HTTP request and domain name is: http://www.companyname.com. In this example, the "http" identifies the URL as a HTTP request and the "companyname.com" is the domain name.

Domain names are much easier to remember and use than their corresponding IP addresses. The Internet Corporation for Assigned Names and Numbers (ICANN) approves some Generic Top-Level Domains (gTLD) and delegates the responsibility to a particular organization (a "registry") for maintaining an authoritative source for the registered domain names within a TLD and their corresponding IP addresses. For certain TLDs (e.g., .biz, .info, .name, and .org) the registry is also the authoritative source for contact information related to the domain name and is referred to as

a "thick" registry. For other TLDs (e.g., .com and .net) only the domain name, registrar identification, and name server files are stored within the registry, and a registrar is the authoritative source for the contact information related to the domain name. Such registries are referred to as "thin" registries. Most domain names having a gTLDs are organized through a Shared Registration System (SRS) based on their TLD.

Referring to FIG. **1**, a user **100** is hereby defined to be a person or entity that is in the process of registering a domain name or has already registered a domain name. A user **100** may also be considered a domain name registrant and/or a customer of the domain name registrar **120**. The user **100** may use a client device **110**, such as, as non-limiting examples, a cell phone, PDA, tablet, laptop computer, or desktop computer to access a website (such as a registrar website **130** of a domain name registrar **120**) via a computer network, such as the Internet.

The process for registering a domain name with .com, .net, .org, and some other TLDs allows a user **100** to use an ICANN-accredited domain name registrar **120** to register their domain name. The domain name registrar **120** includes a domain name registration function **134** to register one or more domain names to a plurality of users **100**. If a user **100**, for example John Doe, wishes to register the domain name "mycompany.com," John Doe may initially determine whether the desired domain name is available by contacting a domain name registrar **120**.

A domain name registrar **120** may operate a registrar website **130** having a plurality of webpages accessible over the Internet by users/customers **100** operating client devices **110**. The website is hosted or operated from one or more servers. The server may be, as a non-limiting example, one or more Dell PowerEdge(s)© rack server(s) although other types of servers or combinations of one or more servers may be used.

Referring to FIG. **2**, the webpages may have one or more data entry fields as well as one or more display fields. The data entry fields allow a user **100** to enter data into the registrar website **130** from a client device **110**.

The user **100** may contact the domain name registrar **120** using the registrar website **130** of the domain name registrar **120** and type a domain name search, such as a desired domain name, into a field on the registrar website **130** created for this purpose. As an example, the user **100** may type "mycompany.com" into the data entry field.

Upon receiving the request from the user **100**, the domain name registrar **120** may ascertain whether "mycompany.com" has already been registered by, as a non-limiting example, checking with a Registry associated with the TLD for the domain name. The results of the search may be displayed on the website to thereby notify the potential user **100** (user **100**) of the availability of the domain name.

If the domain name is available, the user **100** may proceed with the registration process. If the domain name is not available for registration, the user **100** may keep selecting alternative domain names until an available domain name is found. In preferred embodiments, the domain name registrar **120** may use the entered domain name as a domain name search **200** and use the domain name search **200** to create other available similar suggested domain names the user **100** may wish to register.

A user **100** may create a registrar customer account **170** with the domain name registrar **120**. As the domain name registrar **120** will have a plurality of customers, i.e., users **100**, the domain name registrar **120** may manage a plurality of registrar customer accounts **170**. Each registrar customer account **170** may be protected by, as non-limiting examples, requiring a user name and a password and/or a secondary authentication method (biometric or possession of a cell phone having a known cell phone number) from the user **100**. The registrar customer account **170** will hold the information of all of the domain names registered to the user **100** with the domain name registrar **120** (the user **100** may have domain names with other domain name registrars) and the other products and services the user **100** has with the domain name registrar **120**.

As non-limiting examples, a registrar customer account **170** may list one or more domain names registered to the account holder (user **100**). The domain name registrar **120** has access to the registrar customer account **170** of the user **100** and thus knows what domain names are used by and registered to the user **100**.

The present invention may use a learning algorithm **140** to suggest domain names to a user **100**. Learning algorithms may be inspired and loosely patterned after a biological brain and are useful in solving highly complex problems that have a large number of inputs. Learning algorithms may comprise electrical circuits and/or software running on one or more computer hardware servers.

Learning algorithms may comprise a plurality of layers, with each layer comprising a plurality of nodes, i.e., artificial neurons, that may be of any known or later developed type of artificial neuron. As non-limiting examples, the nodes may be neurons (as illustrated in FIG. **12**), sigmoid neurons (as illustrated in FIG. **13**), rectified linear unit (ReLU) neurons, Long Short Term Memory (LSTM) neurons and/or Gated Recurrent Units (GRU) neurons. Any combination of these types of nodes or artificial neurons, or later developed nodes or artificial neurons, may be used in the learning algorithm **140**. Regardless of the specific types of nodes used in the learning algorithm **140**, all nodes must run on electrical circuits and/or software running on one or more computer hardware servers.

As illustrated in FIG. **10**, learning algorithms **140** may comprise an input layer having one or more inputs, one or more hidden layers with each hidden layer having one or more nodes (also referred to as artificial neurons) and an output layer having one or more outputs. Each input in the input layer may forward a single output or number to one or more nodes in the first hidden layer.

As illustrated in the example of FIG. **11**, each rectangular box inside of the multi-layer recurrent neural network (RNN) may be an RNN unit at a specific timestep. Each unit may contain several neurons and performs a mix of linear and non-linear operations. Each unit may propagate its output to its upper-layer unit at the next timestep. The units at the last timestep may output the high dimension meaning vector that encodes the meaning of the input domain name.

Inputs to a recurrent neural network (RNN) may be applied directly to any of the hidden layers, thereby skipping over earlier hidden layers. Each node may apply a different weight to each of its inputs (initially the weight is a random real number, but the weight will likely be adjusted during the training process). The output of each node may be based on a summation of the weighted inputs and a bias for the node. This calculation may be referred to as the activation function and it converts a node's weighted inputs and bias to an output of the node.

Depending on the type of node, i.e., artificial neuron being used, the outputs are typically between −1 and 1, inclusive, 0 and 1, inclusive or greater than or equal to 0. The output for each node may be an input to one or more nodes in the next or later layer as specifically shown in FIGS. **10** and **11**.

Outputs are generally not allowed to be connected to an input of the same layer or earlier layers as learning algorithms **140** typically work better using a forward feed of information.

Each layer in the hidden layers may have one or more nodes that receive inputs from a previous hidden layer and apply their outputs to a subsequent layer. The last layer in the hidden layers may be connected to the inputs to the one or more nodes of the output layer.

Creating a Training Database

A method for creating a training database **180**, which may be used to train the learning algorithm **140** to suggest domain names, will now be discussed. The learning algorithm **140** may be any type of desired artificial intelligence. As non-limiting examples, the learning algorithm **140** may use machine learning or be an artificial neural network (ANN).

A user **100** may login to a registrar customer account **170** using any desired method. In other embodiments, the user **100** does not have to login to a registrar customer account **170** to collect activity data from the user **100**.

Referring to the flowcharts in FIGS. **14** and **15**, a domain name registrar **120** may receive a domain name search **200** from the user **100**, i.e., a customer or a domain name registrant. (Step **1400**) As an example, the domain name search **200** illustrated in FIG. **2** is "mybike.com." Thus, in the illustrated example, the domain name search **200** was in the format of a domain name, but the domain name search **200** may comprise one or more words, symbols, abbreviations or other character strings.

The domain name registrar **120** may generate one or more suggested domain names **210** based on the entered domain name search **200** using any desired method. (Step **1410**) The domain name registrar **120** may transmit the suggested domain names **210** as part of a webpage to the client device **110** of the user **100**. The user **100** may see the suggested domain names **210** on the webpage of the registrar website **130**. A list of suggested domain names **210** that may be selected by the user **100** are also on the webpage illustrated in FIG. **2**. In the illustrated example, the suggested domain names **210** of "bestbike.com," "best-bike.com," "topbike.com," "top-bike.com" and "mybike.us" are illustrated as suggested domain names **210** for the user **100** to select.

The user **100** may select none, one or a plurality of the suggested domain names **210** using any desired means, such as by the user **100** clicking (selecting) on one or more of the suggested domain names **210** with a mouse. Selected suggested domain names **210** may be added to an electronic shopping cart **132**. (Step **1420**)

The user **100** might not purchase/register any of the selected suggested domain names in the electronic shopping cart **132** or the user **100** might purchase/register one or more of the selected suggested domain names in the electronic shopping cart **132**. (Step **1430**)

The process of the user **100**: 1) entering domain name searches **200** into the registrar website **130**; 2) being presented with suggested domain names **210**; 3) having the option to select one or more of the suggested domain names **210** to add to an electronic shopping cart **132** and 4) having the option to purchase/register one or more of the selected suggested domain names in the electronic shopping cart **132** may be repeated by the user **100** any number of times as desired by the user **100**. The domain name registrar **120** may log and store the domain name search(es) **200**, the selected suggested domain name(s) and the purchased/registered domain name(s) to the user **100**.

In a preferred embodiment, the domain name search(es) **200**, the selected suggested domain name(s) and/or the purchased/registered domain name(s) are all collected from a single continuous login session of the user **100**. This helps to insure that the collected data from the user **100** are related to each other and to the same search. Activities of different users or activities of the same user **100**, but at different times, are preferably not mixed together as this might corrupt the training data **340**, i.e., combine inputs with unrelated expected outputs for the learning algorithm.

In another embodiment, the domain name search(es) **200**, selected suggested domain name(s) and/or registered domain name(s) are all evaluated to verify they are related to a common search to prevent mixing inputs with unrelated expected outputs, for example, from the user's professional or commercial activities with the user's personal or family activities. This may be performed, as an example, by tokenizing the domain name search(es) **200**, selected suggested domain name(s) and/or registered domain name(s) and only counted as related if they share at least one token or a synonym of the one token. This filters out unrelated data and prevents the training data **340** from being corrupted by mixing or associating unrelated inputs and expected outputs with each other. (Step **1500**)

Referring to FIG. **3**, the training data **340** may be generated and stored in a training database **180** in predefined allowable formats. (Step **1510**) While the training data **340** may be stored in any type of data structure, the training data **340** in FIG. **3** is stored in a table. In preferred embodiments, the table has a column for an input domain search **300** (input) and a column for an expected domain name suggest (expected output).

The table may also optionally have a column for related searches **320** and/or a column for related registered domain names **330**. The columns in the table may be in any order. Each row contains data for one cycle of training the learning algorithm **140**. Thus, in FIG. **3**, four cycles of training data are illustrated. In practice, it would be desirable to have at least data for thousands of cycles and preferably as many training cycles as possible. The rows may be in any order or may be in a random order in the table.

Data entered by the user **100** representing domain name search(es) **200**, selected suggested domain name(s) and/or registered domain name(s) entered in a single continuous login session of the user **100** may be stored in the training database **180** in the predefined allowable formats. (Step **1520**)

As a non-limiting example of an allowable format, a first domain name search may be an input domain search **300** (stored under the "input domain search" column) and a second domain name search may be an expected domain name suggestion **310** (stored under the "expected domain name suggestion" column). As another non-limiting example of an allowable format, the second domain name search may be an input domain search **300** (stored under the "input domain search" column) and a third domain name search may be an expected domain name suggestion **310** (stored under the "expected domain name suggestion" column). As another non-limiting example of an allowable format, the first domain name search **200** may be an input domain search **300** (stored under the "input domain search" column) and a selected suggested domain name may be an expected domain name suggestion **310** (stored under the "expected domain name suggestion" column). As another non-limiting example of an allowable format, the first domain name search **200** may be an input domain search **300** (stored under the "input domain search" column) and a

registered domain name may be an expected domain name suggestion **310** (stored under the "expected domain name suggestion" column).

These formats are desirable as the input domain search **300** (entered earlier in time by the user **100**) leads to the expected domain name suggestion **310** (produced later in time). Thus, the learning algorithm **140** will take as input the input domain search **300** and be trained to produce the expected domain name suggestion **310**. The training data **340** is preferably stored in the training database **180** in one or more of these allowable formats.

Data entered by the user **100** representing domain name search(es) **200**, selected suggested domain name(s) and/or registered domain name(s) should not be stored in the training database **180** and/or used to train the learning algorithm in predefined unallowable formats. As a non-limiting example of an unallowable format, a second domain name search should not be stored as an input domain search **300** with a first domain name search being stored as an expected domain name suggestion **310**.

As another non-limiting example of an unallowable format, a third domain name search should not be stored as an input domain search **300** with the second domain name search being stored as an expected domain name suggestion **310**. Again, this would incorrectly store a later activity as an input with an earlier activity as an expected output.

As another non-limiting example of an unallowable format, a selected suggested domain name should not be stored as an input domain search **300** with a domain name search stored as an expected domain name suggestion **310**.

As another non-limiting example of an unallowable format, a registered domain name should not be stored as an input domain search **300** with a domain name search being stored as an expected domain name suggestion **310**.

These formats are not desirable as the input domain search **300** (produced later in time by the user) leads to the expected domain name suggestion **310** (entered earlier in time). This is in reverse order from the desired direction of the learning algorithm.

Referring to FIG. **4**, the learning algorithm **140** may transform data from different sources into a common format: pairs of input domain searches **300** and expected domain name suggestions **310**. As a non-limiting example for the searches (bestbike.com, best-bike.com, topbike.com, thetopbike.com), the learning algorithm **140** may generate pairs such as: (bestbike.com, best-bike.com), (best-bike.com, topbike.com), (topbike.com, thetopbike.com), (best-bike.com, topbike.com), (bestbike.com, thetopbike.com).

As another non-limiting example, for the search bestbike.com, (excellentbike.com, greatbike.com) may have been added to an electronic shopping cart **132**. Based on these selected generated domain name, the learning algorithm **140** may generate pairs such as: (bestbike.com, excellentbike.com) and (bestbike.com, greatbike.com).

As another non-limiting example, for the search bestbike.com, if greatbike.com was purchased, the learning algorithm **140** may generate the pair (bestbike.com, greatbike.com).

The learning algorithm **140** may also filter out pairs where two domains are not relevant to each other to improve the quality of the training data **340**, such as: (bestbike.com, sunnyvaleparks.com)

The above process of collecting data may be repeated for a plurality of other users **100** and/or the same user **100** during a different continuous login session. (Step **1530**)

Training a Learning Algorithm

Referring to FIG. **5**, the parameters in the learning algorithm **140**, including the encoder **150** and decoder **160**, may be randomly initialized at the start of the training process. For the training pair (bestbike.com, best-bike.com), the encoder **150** encodes the meaning of the input domain (bestbike.com) into a high dimension meaning vector and passes the high dimension meaning vector to the decoder **160**. The decoder **160** may generate a domain name suggestion based on the high dimension meaning vector. Based on the differences between the decoder **160** output and the expected output, the parameters in the encoder **150** and the decoder **160** may be adjusted so that there will be a higher chance to generate the expected output in a future cycle. All of the pairs in the training data **340** may be fed into the learning algorithm **140** in an iterative manner several times before a domain name suggestion model is generated, i.e., the learning algorithm **140** is fully trained.

FIGS. **16** and **17** are flowcharts for a method of training a learning algorithm **140** using the training data **340** stored in the training database **180**. A domain name registrar **120** may read the training data **340** from the training database **180**. (Step **1600**) The training data **340** preferably comprises an input and an expected output. The input to the learning algorithm **140** may be an input domain name search **200** (preferably tokenized to words) and optionally other domain name search(es), selected suggested domain name(s) and/or registered domain name(s) of a user **100**. The expected output may be an expected domain name suggestion **310**.

FIG. **8** is an example illustration of a learning algorithm **140** that is history aware in that the encoder **150** of the learning algorithm **140** receives as inputs earlier related domain name searches of the user **100**. FIG. **9** is an example illustration of a learning algorithm **140** that is portfolio-driven in that the encoder **150** of the learning algorithm **140** receives as inputs related registered domain names **330** of the user **100**. It should also be understood that the learning algorithm **140** may be both history aware and portfolio-driven so that the encoder **150** of the learning algorithm **140** receives earlier related domain name searches **320** and related registered domain names **330** of the user **100**.

The training involves applying the input to an encoder **150** of the learning algorithm **140**. The example training data **340** in FIG. **3** illustrates inputs and expected outputs for four cycles.

As a first example cycle, the input to an encoder **150** may be the input domain search **300** "mybike.com" (preferably tokenized to "my," "bike" and ".com" (Step **1610**)), and optionally the related searches **320** "best-bike.com" and "topbike.com" and/or the related registered domain names **330** "hotbikes.com" and "great-bikes.com." (Step **1620**) The encoder **150** creates a high dimension meaning vector for this input, which is applied to a decoder **160** of the learning algorithm **140**. (Step **1630**) FIG. **6** illustrates how each token in the input domain search **300** may be applied to a different cell in the encoder **150**.

Any desired method of tokenizing the input domain search **300** into one or more tokens may be used. As a nonlimiting example, the learning algorithm **140** may parse each input domain search **300**, from left to right, character by character and compare the parsed characters to one or more electronic dictionaries looking for words. For the purposes of the specification, each token may comprise a word, place, name, abbreviation, phrase, number, dash and/or a string of characters (default if the word does not fit any other category).

In preferred embodiments, many different language dictionaries (preferably at least the languages most commonly

used on the Internet), location dictionaries, name dictionaries, abbreviation dictionaries and/or phrase dictionaries may be used in identifying tokens/words. The parsing process may be repeated until one or more tokens are found for each input domain search **300**. In some cases, an input domain search **300** might comprise a string of characters that do not match any entries in any dictionary. In such a case, that string of unrecognized characters may still be considered a token.

In some cases, an input domain search **300** may be able to be broken up into two or more sets of tokens. As an example, if the input domain search **300** is "carrot.com," the input domain search **300** may be broken into a first set of tokens "car" and "rot" and a second set of the tokens "carrot." In this situation, the learning algorithm **140** may determine how frequently the words "car" and "rot" appear (and perhaps how frequently they appear in close proximity) and how frequently the word "carrot" appears in some text corpus selected for this purpose.

The encoder **150** may use, as a non-limiting example, 40 dimensions to create a high dimension meaning vector for the input. Thus, the input might have a high dimension meaning vector of [0.5, 0.1, 0.9, 0.2, 0.4, 0.4, 0.3, 0.2, 0.7, 0.1, 0.2, 0, 1, 0.6, 0.6, 0.8, 0.1, 0.8, 0.2, 0.1, 0.8, 0.7, 0.4, 0.3, 0.6, 0.9, 0.9, 0.1, 0.2, 0.1, 0.6, 0.5, 0.3, 0.5, 0.4, 0.2, 0.6, 0.9, 0.7, 0.7] as determined by the encoder **150**. This high dimension meaning vector maps the input to a specified point in high dimensional space.

While each dimension may be assigned any number, each of the 40 dimensions are preferably assigned a value between −1 and 1, inclusive or between 0 and 1, inclusive. The range of values may be selected depending on the types of nodes or artificial neurons used by the learning algorithm **140**. This allows each dimension to be easily applied to the inputs of the nodes of the learning algorithm **140** without additional weighting or scaling of the values. While a single decimal digit (tenth's place) is displayed for each value in the above example to keep the example simple, additional digits, for example, 0.234956, may be used to increase the accuracy of each value for each dimension and thus of the learning algorithm **140** overall.

The high dimension meaning vector for the input may be applied to a decoder **160**. (Step **1700**) The high dimension meaning vector may pass through the nodes or artificial neurons of the decoder **160** to produce a second high dimension meaning vector. The second high dimension meaning vector may be decoded to create a plurality of characters strings. (Step **1710**) The plurality of character strings may be combined to create an output domain name suggestion. (Steps **1720**)

The output domain name suggestion from the decoder **160** of the learning algorithm **140** may be compared with the expected domain name suggestion **310** as stored in the training data **340**. In the current example, the expected domain name suggestion **310** from the training data **340** in FIG. **3** is "bestbike.com."

Using any desired technique, such as, as non-limiting examples, gradient descent and/or back propagation, the parameters of the encoder **150** of the learning algorithm **140** may be adjusted so that the input is more likely to produce the expected output in future cycles.

As a second example cycle, the input to the encoder **150** may be the input domain search **300** "new-york-pizza.com" (preferably tokenized to "new," "-," "york," "-," "pizza" and ".com" (Step **1610**)), and optionally the related searches **320** "italian-food.ny" and/or the related registered domain names **330** "best-ny-pizza.us." (Step **1620**) The encoder **150** creates a high dimension meaning vector for this input, which is applied to a decoder **160** of the learning algorithm **140**. (Step **1630**) The expected domain name suggestion **310** for this cycle, as shown in the training data **340**, is "pizza.ny."

As a third example cycle, the input to the encoder **150** may be the input domain search **300** "smith-family.com" (preferably tokenized to "smith," "-," "family," and ".com" (Step **1610**)), and optionally no related searches **320** and/or the related registered domain names **330** "smith-vacation.us." (Step **1620**) The encoder **150** creates a high dimension meaning vector for this input, which is applied to a decoder **160** of the learning algorithm **140**. (Step **1630**) The expected domain name suggestion **310** for this cycle, as shown in the training data **340**, is "smithfamily.com."

As a fourth example cycle, the input to the encoder **150** may be the input domain search **300** "dogbreeders.us" (preferably tokenized to "dog," "breeders," and ".us" (Step **1610**)), and optionally no related searches **320** and/or no related registered domain names **330**. (Step **1620**) The encoder **150** creates a high dimension meaning vector for this input, which is applied to a decoder **160** of the learning algorithm **140**. (Step **1630**) The expected domain name suggestion **310** for this cycle, as shown in the training data **340**, is "canine.com."

Additional training data **340** is desirable as it allows additional cycles to be run on the learning algorithm **140**, thereby improving the training of the learning algorithm **140**. (Step **1740**)

Additional epochs, i.e., full-cycles of using all (or some given portion) of the training data **340**, may be performed until the learning algorithm **140** has been trained to a desired level. (Step **1750**) If parts of the training data **340** are not used for training, these unused parts of the training data **340** may be applied to the learning algorithm **140** to check the progress of the training of the learning algorithm **140** to determine if the learning algorithm **140** has been trained to the desired level.

Using a Learning Algorithm to Produce a Suggested Domain Name

A domain name registrar may collect activity data regarding the activities of a user **100** on a registrar website **130**, such as one or more domain name searches entered by the user **100** (Step **1800**), one or more suggested domain names **210** selected and entered into an electronic database by the user **100** and/or one or more registered domain names in a registrar customer account **170** of the user **100** (Step **1810**).

The registrar may apply the most recent domain name search **200**, preferably tokenized to words or tokens (Step **1820**), to an encoder **150** of the learning algorithm **140**. In other embodiments, one or more other domain name searches, one or more selected suggested domain names and/or one or more domain names registered to the user **100** are also applied as inputs to the encoder **150** of the learning algorithm **140**. (Step **1830**) It should be appreciated that the same types of input data used to train the learning algorithm **140** should also be used when using the learning algorithm **140** to produce a suggested domain name, i.e., the training data and actual use data should match formats.

The encoder **150** may create a high dimensional meaning vector from the input and apply the high dimensional meaning vector to a decoder **160** of the learning algorithm **140**. (Step **1840**) The decoder **160** may convert the high dimensional meaning vector to one or more character strings (Step **1900**) which may be combined to create a suggested domain name (Step **1910**).

Referring to FIG. **7**, the learning algorithm **140** may also comprise a generative adversarial network (GAN). The GAN may be trained to determine good and bad suggested domain names. Thus, the suggested domain names **210** from the learning algorithm **140** may be applied to the GAN. The suggested domain names **210** that are determined to be good may be transmitted to the user **100** for selection and/or registration while the suggested domain names **210** that are determined to be bad may be screened or filtered out and thus not transmitted to the user **100** for selection and/or registration.

The good suggested domain name may be transmitted to the user **100** (Step **1920**) and, if selected by the user **100** for registration (Step **1930**), the selected suggested domain name may be registered to the user **100** (Step **1940**).

In addition, the domain name registrar **120** may spin the generated suggested domain name to create additional suggested domain names **210** to transmit to the user **100** for possible selection and registration. Any desired method of spinning a suggested domain name may be used.

Using the domain name sunnyvalezoepizza.com as an example, one or more tokens may be dropped to thereby spin the domain names of sunnyvalezoe.com, zoepizza.com, sunnyvalepizza.com, zoe.com and pizza.com.

As another non-limiting example, tokens may be swapped in the example domain name sunnyvalezoepizza.com to thereby produce the domain names of zoesunnyvalepizza.com and sunnyvalepizzazoe.com.

As another non-limiting example, the domain name sunnyvalezoepizza.com may be rewritten to thereby spin the domain names of zoepizzasunnyvale.com, zoespizza.com, thezoepizza.com, zoesunnyvale.com, thesunnyvalepizza.com, pizzazoe.com, sunnyvalepizzeria.com, sunnyvalezoerestaurant.com, zoe-pizza.com, zoepizzeria.com, sunnyvale-zoe.com and sunnyvalezoeonline.com.

As another non-limiting example, the domain name sunnyvalezoepizza.com may be made a plural to spin the domain name sunnyvalezoepizzas.com.

As another non-limiting example, a synonym may replace one of the tokens in the domain name sunnyvalezoepizza.com to spin the domain name sunnyvalezoepizzeria.com.

Other embodiments and uses of the above inventions will be apparent to those having ordinary skill in the art upon consideration of the specification and practice of the invention disclosed herein. The specification and examples given should be considered exemplary only, and it is contemplated that the appended claims will cover any other such embodiments or modifications as fall within the true scope of the invention.

The Abstract accompanying this specification is provided to enable the United States Patent and Trademark Office and the public generally to determine quickly from a cursory inspection the nature and gist of the technical disclosure and in no way intended for defining, determining, or limiting the present invention or any of its embodiments.

The invention claimed is:

1. A method for a learning algorithm to suggest a domain name to a user, comprising the steps of:

receiving, by a domain name registrar, in chronological order from oldest to newest, a first domain name search and a second domain name search from the user;

reading, by the domain name registrar, a previously registered domain name from a registrar customer account of the user;

tokenizing, by the domain name registrar, the second domain name search into a first plurality of character strings;

applying, by the domain name registrar, the previously registered domain name, the first domain name search and the first plurality of character strings to a corresponding plurality of cells in an encoder of a trained machine learning algorithm;

wherein the machine learning algorithm has been trained to obtain the trained machine learning algorithm, by:

utilizing training data comprising

an input comprising

a training domain name search, and

an output comprising

a training expected domain name suggestion;

applying the input to the encoder of the machine learning algorithm;

outputting a training actual domain name suggestion by a decoder of the machine learning algorithm;

comparing the training expected domain name suggestion to the training actual suggested domain name; and

iteratively modifying the machine learning algorithm until a second training actual domain name suggestion is more likely to match the training expected domain name suggestion;

creating, by the encoder, a vector for the previously registered domain name, the first domain name search and the plurality of character strings;

applying, by the domain name registrar, the vector to the decoder of the trained machine learning algorithm;

creating, by the decoder, a second plurality of character strings;

combining, by the domain name registrar, the second plurality of character strings to create a suggested domain name;

transmitting, by the domain name registrar, the suggested domain name to the user;

receiving, by the domain name registrar, a selection of the suggested domain name for registration from the user; and

registering, by the domain name registrar, the selected suggested domain name to the user.

2. The method of claim **1**, further comprising the steps of:

spinning, by the domain name registrar, the suggested domain name to create a plurality of suggested domain names;

transmitting, by the domain name registrar, the plurality of suggested domain names to the user;

receiving, by the domain name registrar, a selection of one or more domain names from the plurality of suggested domain names by the user; and

registering, by the domain name registrar, the selection of one or more domain names to the user.

3. The method of claim **1**, wherein the domain name registrar is in communication with the user through a domain name registrar website.

4. The method of claim **1**, wherein the domain name registrar has a plurality of customers and each customer in the plurality of customers has a registrar customer account.

5. The method of claim **1**, wherein the learning algorithm was trained using actions of a plurality of customers of the domain name registrar.

6. The method of claim **1**, wherein the domain name registrar confirms the suggested domain name is available to be registered prior to transmitting the suggested domain name to the user.

7. A method for a learning algorithm to suggest a domain name to a user, comprising the steps of:

receiving, by a domain name registrar, in chronological order from oldest to newest, a first domain name search and a second domain name search from the user;

tokenizing, by the domain name registrar, the second domain name search into a first plurality of character strings;

applying, by the domain name registrar, the first domain name search and the first plurality of character strings to a corresponding plurality of cells in an encoder of a trained machine learning algorithm;

wherein the machine learning algorithm has been trained to obtain the trained machine learning algorithm, by:

utilizing training data comprising

an input comprising

a training domain name search, and

an output comprising

a training expected domain name suggestion;

applying the input to the encoder of the machine learning algorithm;

outputting a training actual domain name suggestion by a decoder of the machine learning algorithm;

comparing the training expected domain name suggestion to the training actual suggested domain name; and

iteratively modifying the machine learning algorithm until a second training actual domain name suggestion is more likely to match the training expected domain name suggestion;

creating, by the encoder, a vector for the first domain name search and the first plurality of character strings;

applying, by the domain name registrar, the vector to the decoder of the trained machine learning algorithm;

creating, by the decoder, a second plurality of character strings;

combining, by the domain name registrar, the second plurality of character strings to create a suggested domain name;

transmitting, by the domain name registrar, the suggested domain name to the user;

receiving, by the domain name registrar, a selection of the suggested domain name for registration from the user; and

registering, by the domain name registrar, the selected suggested domain name to the user.

8. The method of claim **7**, further comprising the steps of:

spinning, by the domain name registrar, the suggested domain name to create a plurality of suggested domain names;

transmitting, by the domain name registrar, the plurality of suggested domain names to the user;

receiving, by the domain name registrar, a selection of one or more domain names from the plurality of suggested domain names by the user; and

registering, by the domain name registrar, the selection of one or more domain names to the user.

9. The method of claim **7**, wherein the domain name registrar is in communication with the user through a domain name registrar website.

10. The method of claim **7**, wherein the domain name registrar has a plurality of customers and each customer in the plurality of customers has a registrar customer account.

11. The method of claim **7**, wherein the learning algorithm was trained using actions of a plurality of customers of the domain name registrar.

12. The method of claim **7**, wherein the domain name registrar confirms the suggested domain name is available to be registered prior to transmitting the suggested domain name to the user.

13. A method for a learning algorithm to suggest a domain name to a user, comprising the steps of:

receiving, by a domain name registrar, a domain name search from the user;

reading, by the domain name registrar, a previously registered domain name from a registrar customer account of the user;

tokenizing, by the domain name registrar, the domain name search into a first plurality of character strings;

applying, by the domain name registrar, the previously registered domain name and the first plurality of character strings to a corresponding plurality of cells in an encoder of a trained machine learning algorithm;

wherein the machine learning algorithm has been trained to obtain the trained machine learning algorithm, by:

utilizing training data comprising

an input comprising

a training domain name search, and

an output comprising

a training expected domain name suggestion;

applying the input to the encoder of the machine learning algorithm;

outputting a training actual domain name suggestion by a decoder of the machine learning algorithm; and

comparing the training expected domain name suggestion to the training actual suggested domain name; and

iteratively modifying the machine learning algorithm until a second training actual domain name suggestion is more likely to match the training expected domain name suggestion;

creating, by the encoder, a vector for past registered domain names and the plurality of character strings;

applying, by the domain name registrar, the vector to the decoder of the trained machine learning algorithm;

creating, by the decoder, a second plurality of character strings;

combining, by the domain name registrar, the second plurality of character strings to create a suggested domain name;

transmitting, by the domain name registrar, the suggested domain name to the user;

receiving, by the domain name registrar, a selection of the suggested domain name for registration from the user; and

registering, by the domain name registrar, the selected suggested domain name to the user.

14. The method of claim **13**, further comprising the steps of:

spinning, by the domain name registrar, the suggested domain name to create a plurality of suggested domain names;

transmitting, by the domain name registrar, the plurality of suggested domain names to the user;

receiving, by the domain name registrar, a selection of one or more domain names from the plurality of suggested domain names by the user; and

registering, by the domain name registrar, the selection of one or more domain names to the user.

15. The method of claim 13, wherein the domain name registrar is in communication with the user through a domain name registrar website.

16. The method of claim 13, wherein the domain name registrar has a plurality of customers and each customer in the plurality of customers has a registrar customer account.

17. The method of claim 13, wherein the learning algorithm was trained using actions of a plurality of customers of the domain name registrar.

18. The method of claim 13, wherein the domain name registrar confirms the suggested domain name is available to be registered prior to transmitting the suggested domain name to the user.

* * * * *