Innovation, Sciences et
Développement économique Canada
Office de la Propriété Intellectuelle du Canada

Innovation, Science and
Economic Development Canada
Canadian Intellectual Property Office

CA 3033859 C 2022/07/19

(11)(21) **3 033 859**

(12) **BREVET CANADIEN
CANADIAN PATENT**

(13) **C**

(86) **Date de dépôt PCT/PCT Filing Date:** 2017/07/12

(87) **Date publication PCT/PCT Publication Date:** 2018/01/18

(45) **Date de délivrance/Issue Date:** 2022/07/19

(85) **Entrée phase nationale/National Entry:** 2019/02/13

(86) **N° demande PCT/PCT Application No.:** US 2017/041727

(87) **N° publication PCT/PCT Publication No.:** 2018/013698

(30) **Priorités/Priorities:** 2016/07/15 (US62/362,688); 2016/10/13 (US15/292,510); 2016/10/14 (US15/293,553); 2017/04/14 (US15/488,052)

(51) **Cl.Int./Int.Cl.** *G06Q 10/10* (2012.01), *G06F 40/174* (2020.01), *G06F 40/284* (2020.01), *G06N 20/00* (2019.01), *G06Q 40/00* (2012.01)

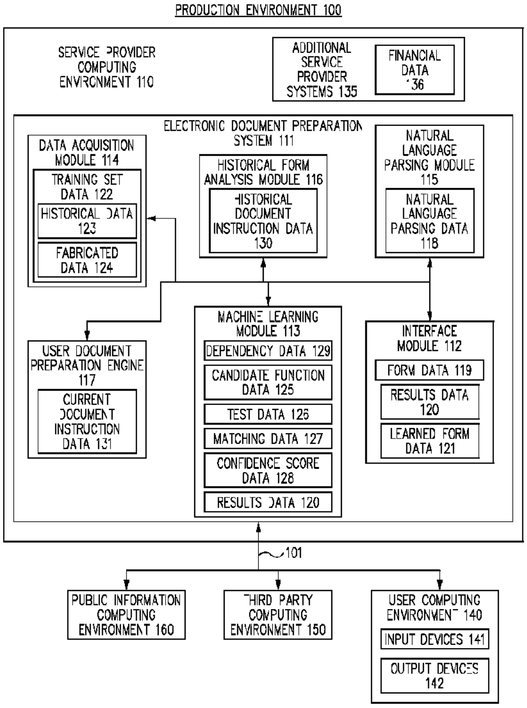
(72) **Inventeurs/Inventors:**
MUKHERJEE, SAIKAT, US;
YAGHOOBZADEH, YADOLLAH, US

(73) **Propriétaire/Owner:**
INTUIT INC., US

(74) **Agent:** OSLER, HOSKIN & HARCOURT LLP

(54) Titre : PROCEDE ET SYSTEME D'EXTRACTION AUTOMATIQUE DE TERMES FISCAUX PERTINENTS DES FORMULAIRES ET INSTRUCTIONS

(54) Title: METHOD AND SYSTEM FOR AUTOMATICALLY EXTRACTING RELEVANT TAX TERMS FROM FORMS AND INSTRUCTIONS


PRODUCTION ENVIRONMENT 100
SERVICE PROVIDER COMPUTING ENVIRONMENT 110
ADDITIONAL SERVICE PROVIDER SYSTEMS 135
FINANCIAL DATA 136
ELECTRONIC DOCUMENT PREPARATION SYSTEM 111
DATA ACQUISITION MODULE 114
TRAINING SET DATA 122
HISTORICAL DATA 123
FABRICATED DATA 124
HISTORICAL FORM ANALYSIS MODULE 116
HISTORICAL DOCUMENT INSTRUCTION DATA 130
NATURAL LANGUAGE PARSING MODULE 115
NATURAL LANGUAGE PARSING DATA 118
USER DOCUMENT PREPARATION ENGINE 117
CURRENT DOCUMENT INSTRUCTION DATA 131
MACHINE LEARNING MODULE 113
DEPENDENCY DATA 129
CANDIDATE FUNCTION DATA 125
TEST DATA 126
MATCHING DATA 127
CONFIDENCE SCORE DATA 128
RESULTS DATA 120
INTERFACE MODULE 112
FORM DATA 119
RESULTS DATA 120
LEARNED FORM DATA 121
101
PUBLIC INFORMATION COMPUTING ENVIRONMENT 160
THIRD PARTY COMPUTING ENVIRONMENT 150
USER COMPUTING ENVIRONMENT 140
INPUT DEVICES 141
OUTPUT DEVICES 142

(57) **Abrégé/Abstract:**

A method and system parses natural language in a unique way, grouping words commonly used together in a text corpus relating to one or more forms associated with document preparation, and eliminating less important words determined by frequency of usage and other techniques. Remaining word groups are then refined using several unique tests and recombinations, resulting in a final word group set that may be used to determine functions associated with form fields on a tax form, for example.

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
18 January 2018 (18.01.2018)

WIPO | PCT

(10) International Publication Number
WO 2018/013698 A1

(51) International Patent Classification:
G06Q 10/10 (2012.01) *G06F 17/27* (2006.01)
G06Q 40/00 (2006.01)

(21) International Application Number: PCT/US2017/041727

(22) International Filing Date: 12 July 2017 (12.07.2017)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:

62/362,688	15 July 2016 (15.07.2016)	US
15/292,510	13 October 2016 (13.10.2016)	US
15/293,553	14 October 2016 (14.10.2016)	US
15/488,052	14 April 2017 (14.04.2017)	US

(71) Applicant: INTUIT INC. [US/US]; 2700 Coast Avenue, Mountain View, California 94043 (US).

(72) Inventors: MUKHERJEE, Saikat; c/o Intuit Inc., 2700 Coast Avenue, Mountain View, California 94043 (US). **YAGHOOBZADEH, Yadollah**; c/o Intuit Inc., 2700 Coast Avenue, Mountain View, California 94043 (US).

(74) Agent: MCKAY, Philip; Hawley Troxell, P.O. Box 1617, Boise, Idaho 83701-1617 (US).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

(54) Title: METHOD AND SYSTEM FOR AUTOMATICALLY EXTRACTING RELEVANT TAX TERMS FROM FORMS AND INSTRUCTIONS

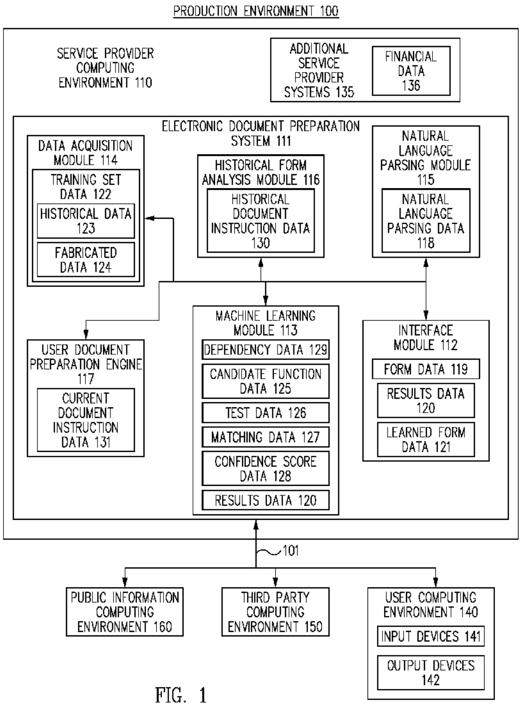

PRODUCTION ENVIRONMENT 100
SERVICE PROVIDER COMPUTING ENVIRONMENT 110
ADDITIONAL SERVICE PROVIDER SYSTEMS 135
FINANCIAL DATA 136
ELECTRONIC DOCUMENT PREPARATION SYSTEM 111
DATA ACQUISITION MODULE 114
TRAINING SET DATA 122
HISTORICAL DATA 123
FABRICATED DATA 124
HISTORICAL FORM ANALYSIS MODULE 116
HISTORICAL DOCUMENT INSTRUCTION DATA 130
NATURAL LANGUAGE PARSING MODULE 115
NATURAL LANGUAGE PARSING DATA 118
USER DOCUMENT PREPARATION ENGINE 117
CURRENT DOCUMENT INSTRUCTION DATA 131
MACHINE LEARNING MODULE 113
DEPENDENCY DATA 129
CANDIDATE FUNCTION DATA 125
TEST DATA 126
MATCHING DATA 127
CONFIDENCE SCORE DATA 128
RESULTS DATA 120
INTERFACE MODULE 112
FORM DATA 119
RESULTS DATA 120
LEARNED FORM DATA 121
101
PUBLIC INFORMATION COMPUTING ENVIRONMENT 160
THIRD PARTY COMPUTING ENVIRONMENT 150
USER COMPUTING ENVIRONMENT 140
INPUT DEVICES 141
OUTPUT DEVICES 142

FIG. 1

(57) Abstract: A method and system parses natural language in a unique way, grouping words commonly used together in a text corpus relating to one or more forms associated with document preparation, and eliminating less important words determined by frequency of usage and other techniques. Remaining word groups are then refined using several unique tests and recombinations, resulting in a final word group set that may be used to determine functions associated with form fields on a tax form, for example.

[Continued on next page]

Declarations under Rule 4.17:

- *as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))*
- *as to the applicant's entitlement to claim the priority of the earlier application (Rule 4.17(iii))*

Published:

- *with international search report (Art. 21(3))*

METHOD AND SYSTEM FOR AUTOMATICALLY EXTRACTING RELEVANT TAX TERMS FROM FORMS AND INSTRUCTIONS

Saikat Mukherjee
Yadollah Yaghoobzadeh

BACKGROUND

[0002] Many people use electronic document preparation systems to help prepare important documents electronically. For example, each year millions of people use electronic document preparation systems customized for tax, i.e. electronic tax return preparation systems, to help prepare and file their tax returns. Typically, electronic tax return preparation systems receive tax related information from a user and then automatically populate the various fields in electronic versions of government tax forms. Electronic tax return preparation systems represent a potentially flexible, highly accessible, and affordable source of tax return preparation assistance for customers. However, processes that enable the electronic tax return preparation systems to determine underlying relations between the various fields and automatically determine and populate various data fields of the tax forms often utilize large amounts of computing system resources and human resources.

[0003] For instance, due to changes in tax laws, or due to updates in government tax rules, tax forms can change from year to year, or even multiple times in a same year. If a physical or electronic tax form required by a governmental entity is updated, or a new tax form is introduced, it is typically very difficult to efficiently update electronic tax return preparation systems to correctly determine tax data appropriate for and populate the various fields of the new or changed tax forms with required values. Tax forms are written by humans for human review, interpretation and understanding. A particular line of an updated tax form may have text describing a requirement of an input according to one or more functions that use line item values from other lines of the updated tax form and/or line item values from other tax related forms or worksheets. These functions range from very simple to very complex, and are often baffling to the humans the text of the various lines was written for, and are thus even much more burdensome when a computing system is introduced in the form of a tax preparation system that is configured to prepare and/or file electronic versions of the tax forms.

[0004] Updating an electronic tax return preparation system often includes utilizing a combination of tax experts to interpret the tax forms consistent with the intent of the humans who prepared the text of the tax forms, software and system engineers who consult with the tax experts to understand and develop the human expert view of individual tax forms, and large amounts of computing resources, to develop, code, and incorporate the new functions and forms into the electronic tax return preparation system.

[0005] Interaction that is required between the tax experts, software and system engineers can lead to significant software release delays and incur great expense in releasing an updated version of the electronic tax return preparation system. These delays and expenses are then passed on to customers of the electronic tax return preparation system who have deadlines to file tax returns associated with the new or updated forms. Furthermore, because humans are inherently error prone, already-existing processes for updating electronic tax returns can introduce significant inaccuracies into the functions and processes of the electronic tax return preparation system.

[0006] These expenses, delays, and inaccuracies can have an adverse impact on the implementation and use of traditional electronic tax return preparation systems. Customers may lose confidence in the electronic tax return preparation systems. Furthermore, customers may simply decide to utilize less expensive options for preparing their taxes. Further, vast amounts of computing resources are consumed determining inaccurate tax return data which is then provided to and processed by other entities, such as government entities, i.e. the Internal Revenue Service.

[0007] These issues and drawbacks are not limited to electronic tax return preparation systems. Any electronic document preparation system that assists users to electronically fill out forms or prepare documents suffer from these same inaccuracies and drawbacks when the physical forms relating to the electronic forms are created or updated. This a longstanding technical problem existing in many computing fields.

SUMMARY

[0008] Embodiments of the present disclosure provide a technical solution to the longstanding problems discussed herein, and thus solve some of the shortcomings associated with traditional electronic document preparation systems by providing methods and systems for employing natural language processing to convert physical text from a text corpus relating to a physical form having one or more form fields to electronic textual data and analyze the electronic textual data to develop and incorporate electronic representations of functions derived from the electronic textual data. The embodiments utilize machine learning to interpret the electronic textual data derived from the physical text-based form and other tax form data to develop electronic representations of functions which will fulfill the requirements of the physical form text with a high degree of accuracy not found in traditional prior art systems. In particular, embodiments of the present disclosure receive text-based form data of or related to a new or updated text-based form that includes text relating to tax form data fields. Embodiments of the present disclosure utilize machine learning to quickly and accurately develop and determine an electronic function that is equivalent to form field text provided on a physical text-based form such as a tax form, an invoice form, or otherwise. In one embodiment, one or more line-items of a new or updated text-based form includes one or more requirements or dependencies to use a result from either of one or more different line items from the new or updated text-based form or one or more line items of a different text-based form that has been or will later be associated with electronic functions as well.

[0009] The machine learning process for learning and incorporating the new and/or updated form includes, in various embodiments, converting at least a portion of the form and at least a portion of textual data of documentation relating to the physical form, such as an instruction booklet or other documentation, to electronic textual data, if needed, extracting terms

from the electronic textual data combining all of the extracted terms, analyzing the extracted form terms to determine word groups that are presented as single terms and/or multiple term groups, and ranking the single terms and multiple term groups according to frequency of usage and other criteria, excluding all single terms and multiple term groups that include one or more words from an exclusion list.

[0010] The machine learning process further includes determining usage frequency data regarding the word groups and eliminating word groups not meeting a predetermined usage frequency criteria. Different frequency ratios are obtained using multiple different criteria and a combined word data set is obtained that meets criteria based on the various frequency ratios. The combined word data set is then refined according to various rules, such as eliminating, as one example, shorter word groups that are always found within a longer word group and further combining two shorter word groups sharing a common word into a single longer word group and eliminating the shorter word groups, resulting in final word group data representing a final word group. The final word group data is then formed as nodes and leaves in a hierarchy for different form fields and incorporated into an electronic word processing system.

[0011] In one embodiment, dependencies for a given data field of the new and/or updated form include references to data values from one or more other data fields of the new and/or updated form. In one embodiment, the dependencies for a given data field of the new and/or updated form includes references to data values from other data fields of one or more other old, new, or updated forms, worksheets, or data values from other locations internal or external to the electronic document management system. In one embodiment, the dependencies include one or more constants.

[0012] In addition to possibly including one or more dependencies, in one embodiment, a final function for a given data field of the new and/or updated form includes one or more operators that operate on one or more of the dependencies in a particular manner. The operators include, in various embodiments, arithmetic operators such as addition, subtraction, multiplication, division or other mathematical operators such as exponential functions and logical operators such as if-then and/or if-then-else operators, and/or Boolean operators such as true/false. The operators can include also existence condition operators that depend on the existence of a data value in another data field of new and/or updated form, in a form other than the new and/or updated form, or in some other location or data set. The operators can include string comparisons and/or rounding or truncating operations.

[0013] Embodiments of the present disclosure address some of the shortcomings associated with traditional electronic document preparation systems that do not adequately and efficiently incorporate functions associated with new forms or with changes associated with updated forms. An electronic document preparation system in accordance with one or more embodiments provides efficient and reliable incorporation of new and/or updated forms by utilizing machine learning in conjunction with training set data in order to quickly and accurately incorporate and learn functions associated with those new and/or updated forms. The various embodiments of the disclosure can be implemented to improve the technical fields of data processing, resource management, data collection, and user experience. Therefore, the various described embodiments of the disclosure and their associated benefits amount to significantly more than an abstract idea. In particular, by utilizing machine learning to learn and incorporate new and/or updated forms in an electronic document preparation system, users can save money and time and can better manage their finances.

[0014] Using the disclosed embodiments of a method and system for learning and incorporating new and/or updated forms in an electronic document preparation system, a method and system for learning and incorporating new and/or updated forms in an electronic document preparation system significantly greater accurately is provided over traditional prior art systems. Therefore, the disclosed embodiments provide a technical solution to the long standing technical problem of efficiently learning and incorporating new and/or updated forms in an electronic document preparation system.

[0015] In addition, the disclosed embodiments of a method and system for learning and incorporating new and/or updated forms in an electronic document preparation system are also capable of dynamically adapting to constantly changing fields such as tax return preparation and other kinds of document preparation. Consequently, the disclosed embodiments of a method and system for learning and incorporating new and/or updated forms in an electronic document preparation system also provide a technical solution to the long standing technical problem of static and inflexible electronic document preparation systems.

[0016] The result is a much more accurate, adaptable, and robust method and system for learning and incorporating new and/or updated forms in an electronic document preparation system, but thereby serves to bolster confidence in electronic document preparation systems. This, in turn, results in: less human and processor resources being dedicated to analyzing new and/or updated forms because more accurate and efficient analysis methods can be implemented, i.e., fewer processing and memory storage assets; less memory and storage bandwidth being dedicated to buffering and storing data; less communication bandwidth being utilized to transmit data for analysis.

[0017] The disclosed method and system for learning and incorporating new and/or updated forms in an electronic document preparation system does not encompass, embody, or preclude other forms of innovation in the area of electronic document preparation systems. In addition, the disclosed method and system for learning and incorporating new and/or updated forms in an electronic document preparation system is not related to any fundamental economic practice, fundamental data processing practice, mental steps, or pen and paper based solutions, and is, in fact, directed to providing solutions to new and existing problems associated with electronic document preparation systems. Consequently, the disclosed method and system for learning and incorporating new and/or updated forms in an electronic document preparation system, does not encompass, and is not merely, an abstract idea or concept.

BRIEF DESCRIPTION OF THE DRAWINGS

[0018] FIG. 1 is a block diagram of software architecture for learning and incorporating new and/or updated forms in an electronic document preparation system, in accordance with one embodiment.

[0019] FIG. 2 is a block diagram of a process for learning and incorporating new and/or updated forms in an electronic document preparation system, in accordance with one embodiment.

[0020] FIG. 3 is a flow diagram of a process for learning and incorporating new and/or updated forms in an electronic document preparation system, in accordance with one embodiment.

[0021] FIG. 4 is a flow diagram of a process for learning and incorporating new and/or updated forms in an electronic document preparation system, in accordance with one embodiment.

[0022] FIG. 5 is a flow diagram of a process for learning and incorporating new and/or updated forms in an electronic document preparation system, in accordance with one embodiment.

[0023] Common reference numerals are used throughout the figures and the detailed description to indicate like elements. One skilled in the art will readily recognize that the above figures are examples and that other architectures, modes of operation, orders of operation, and elements/functions can be provided and implemented without departing from the characteristics and features of the invention, as set forth in the claims.

DETAILED DESCRIPTION

[0024] Embodiments will now be discussed with reference to the accompanying figures, which depict one or more exemplary embodiments. Embodiments may be implemented in many different forms and should not be construed as limited to the embodiments set forth herein, shown in the figures, and/or described below. Rather, these exemplary embodiments are provided to allow a complete disclosure that conveys the principles of the invention, as set forth in the claims, to those of skill in the art.

[0025] Herein, the term "production environment" includes the various components, or assets, used to deploy, implement, access, and use, a given application as that application is intended to be used. In various embodiments, production environments include multiple assets that are combined, communicatively coupled, virtually and/or physically connected, and/or associated with one another, to provide the production environment implementing the application.

[0026] As specific illustrative examples, the assets making up a given production environment can include, but are not limited to, one or more computing environments used to implement the application in the production environment such as a data center, a cloud computing environment, a dedicated hosting environment, and/or one or more other computing environments in which one or more assets used by the application in the production environment are implemented; one or more computing systems or computing entities used to implement the application in the production environment; one or more virtual assets used to implement the application in the production environment; one or more supervisory or control systems, such as hypervisors, or other monitoring and management systems, used to monitor and control assets and/or components of the production environment; one or more communications channels for sending and receiving data used to implement the application in the production environment; one or more access control systems for limiting access to various components of the production environment, such as firewalls and gateways; one or more traffic and/or routing systems used to direct, control, and/or buffer, data traffic to components of the production environment, such as routers and switches; one or more communications endpoint proxy systems used to buffer, process, and/or direct data traffic, such as load balancers or buffers; one or more secure communication protocols and/or endpoints used to encrypt/decrypt data, such as Secure Sockets

Layer (SSL) protocols, used to implement the application in the production environment; one or more databases used to store data in the production environment; one or more internal or external services used to implement the application in the production environment; one or more backend systems, such as backend servers or other hardware used to process data and implement the application in the production environment; one or more software systems used to implement the application in the production environment; and/or any other assets/components making up an actual production environment in which an application is deployed, implemented, accessed, and run, e.g., operated, as discussed herein, and/or as known in the art at the time of filing, and/or as developed after the time of filing.

[0027] As used herein, the terms "computing system", "computing device", and "computing entity", include, but are not limited to, a virtual asset; a server computing system; a workstation; a desktop computing system; a mobile computing system, including, but not limited to, smart phones, portable devices, and/or devices worn or carried by a user; a database system or storage cluster; a switching system; a router; any hardware system; any communications system; any form of proxy system; a gateway system; a firewall system; a load balancing system; or any device, subsystem, or mechanism that includes components that can execute all, or part, of any one of the processes and/or operations as described herein.

[0028] In addition, as used herein, the terms computing system and computing entity, can denote, but are not limited to, systems made up of multiple: virtual assets; server computing systems; workstations; desktop computing systems; mobile computing systems; database systems or storage clusters; switching systems; routers; hardware systems; communications systems; proxy systems; gateway systems; firewall systems; load balancing systems; or any devices that can be used to perform the processes and/or operations as described herein.

[0029] As used herein, the term "computing environment" includes, but is not limited to, a logical or physical grouping of connected or networked computing systems and/or virtual assets using the same infrastructure and systems such as, but not limited to, hardware systems, software systems, and networking/communications systems. Typically, computing environments are either known environments, e.g., "trusted" environments, or unknown, e.g., "untrusted" environments. Typically, trusted computing environments are those where the assets, infrastructure, communication and networking systems, and security systems associated with the computing systems and/or virtual assets making up the trusted computing environment, are either under the control of, or known to, a party.

[0030] In various embodiments, each computing environment includes allocated assets and virtual assets associated with, and controlled or used to create, and/or deploy, and/or operate an application.

[0031] In various embodiments, one or more cloud computing environments are used to create, and/or deploy, and/or operate an application that can be any form of cloud computing environment, such as, but not limited to, a public cloud; a private cloud; a virtual private network (VPN); a subnet; a Virtual Private Cloud (VPC); a sub-net or any security/communications grouping; or any other cloud-based infrastructure, sub-structure, or architecture, as discussed herein, and/or as known in the art at the time of filing, and/or as developed after the time of filing.

[0032] In many cases, a given application or service may utilize, and interface with, multiple cloud computing environments, such as multiple VPCs, in the course of being created, and/or deployed, and/or operated.

[0033] As used herein, the term “virtual asset” includes any virtualized entity or resource, and/or virtualized part of an actual, or “bare metal” entity. In various embodiments, the virtual assets can be, but are not limited to, virtual machines, virtual servers, and instances implemented in a cloud computing environment; databases associated with a cloud computing environment, and/or implemented in a cloud computing environment; services associated with, and/or delivered through, a cloud computing environment; communications systems used with, part of, or provided through, a cloud computing environment; and/or any other virtualized assets and/or sub-systems of “bare metal” physical devices such as mobile devices, remote sensors, laptops, desktops, point-of-sale devices, etc., located within a data center, within a cloud computing environment, and/or any other physical or logical location, as discussed herein, and/or as known/available in the art at the time of filing, and/or as developed/made available after the time of filing.

[0034] In various embodiments, any, or all, of the assets making up a given production environment discussed herein, and/or as known in the art at the time of filing, and/or as developed after the time of filing, can be implemented as one or more virtual assets.

[0035] In one embodiment, two or more assets, such as computing systems and/or virtual assets, and/or two or more computing environments, are connected by one or more communications channels including but not limited to, Secure Sockets Layer communications channels and various other secure communications channels, and/or distributed computing system networks, such as, but not limited to: a public cloud; a private cloud; a virtual private network (VPN); a subnet; any general network, communications network, or general network/communications network system; a combination of different network types; a public network; a private network; a satellite network; a cable network; or any other network capable of allowing communication between two or more assets, computing systems, and/or virtual assets, as discussed herein, and/or available or known at the time of filing, and/or as developed after the time of filing.

[0036] As used herein, the term "network" includes, but is not limited to, any network or network system such as, but not limited to, a peer-to-peer network, a hybrid peer-to-peer network, a Local Area Network (LAN), a Wide Area Network (WAN), a public network, such as the Internet, a private network, a cellular network, any general network, communications network, or general network/communications network system; a wireless network; a wired network; a wireless and wired combination network; a satellite network; a cable network; any combination of different network types; or any other system capable of allowing communication between two or more assets, virtual assets, and/or computing systems, whether available or known at the time of filing or as later developed.

[0037] As used herein, the term "user" includes, but is not limited to, any party, parties, entity, and/or entities using, or otherwise interacting with any of the methods or systems discussed herein. For instance, in various embodiments, a user can be, but is not limited to, a person, a commercial entity, an application, a service, and/or a computing system. In one or more embodiments, there may be different parties noted that perform different levels of tasks, such as a user filling in a form supplied through an electronic document system managed, operated or otherwise controlled by a third party, such as a business entity.

[0038] As used herein, the term "relationship(s)" includes, but is not limited to, a logical, mathematical, statistical, or other association between one set or group of information, data, and/or users and another set or group of information, data, and/or users, according to one embodiment. The logical, mathematical, statistical, or other association (i.e., relationship) between the sets or groups can have various ratios or correlation, such as, but not limited to, one-to-one, multiple-to-one, one-to-multiple, multiple-to-multiple, and the like, according to one embodiment. As a non-limiting example, if the disclosed electronic document preparation system determines a relationship between a first group of data and a second group of data, then a characteristic or subset of a first group of data can be related to, associated with, and/or correspond to one or more characteristics or subsets of the second group of data, or vice-versa, according to one embodiment. Therefore, relationships may represent one or more subsets of the second group of data that are associated with one or more subsets of the first group of data, according to one embodiment. In one embodiment, the relationship between two sets or groups of data includes, but is not limited to similarities, differences, and correlations between the sets or groups of data.

HARDWARE ARCHITECTURE

[0039] FIG. 1 illustrates a block diagram of a production environment 100 for learning and incorporating new and/or updated forms in an electronic document preparation system, according to one embodiment. Embodiments of the present disclosure provide methods and systems for learning and incorporating new and/or updated forms in an electronic document preparation system.

[0040] In particular, embodiments of the present disclosure receive form data related to a new and/or updated form having data fields to be completed according to instructions set forth in the new and/or updated form and utilize machine learning to parse natural language and correctly determine and learn one or more functions equivalent to or otherwise represented by instructions for each data field. Those learned functions are then incorporated into the electronic document preparation system.

[0041] Embodiments discussed herein gather training set data including previously filled forms related to the new and/or updated form, and/or including fabricated data as discussed herein. One or more embodiments of the present disclosure generate, for one or more data fields needing a new learned function, dependency data that indicates one or more dependencies likely to be included in an acceptable function for the data field.

[0042] Embodiments of the present disclosure utilize machine learning systems and processes to generate candidate functions for data fields to be learned. The candidate functions may be based on the one or more dependencies and can include one or more operators selected from a set of operators. The operators can operate on one or more of the possible dependencies and training set data. Embodiments of the present disclosure generate test data, i.e. output data, for each candidate function by applying the candidate function to one or more dependencies and/or the training set data.

[0043] Embodiments of the present disclosure compare the test data to the data values in the corresponding fields of the previously filled forms of the training set data or of the fabricated data. Embodiments of the present disclosure generate matching data indicating how closely the test data matches the data values of the previously filled forms of the training set data and/or how closely the test data matches the fabricated data.

[0044] In one embodiment, in a system wherein many candidate functions are generated and tested, components of a predetermined number of candidate functions that match the training set data better than other candidate functions may be used to generate new candidate functions which are then tested. In one embodiment, a component of a new candidate function includes one or more operators of the candidate function. In one embodiment, a component of a new candidate function includes one or more constants of the candidate function. In one embodiment, a component of a new candidate function includes one or more dependencies used to generate the candidate function.

[0045] In one embodiment, one or more of the predetermined number of candidate functions that match the training set data better than other candidate functions are split into two or more components each, and the split components recombined into new candidate functions that are then tested to determine how well test data generated from those new candidate functions match the training set data. One or more of those new candidate functions that are determined to generate test data that match the training set data better than the original candidate functions are then again split, if desired, and recombined into a second set of new candidate functions, and so on, until the resulting candidate functions produce test data that are deemed to match the training set data within a predetermined margin of error, as discussed herein. Thus, machine learning module 113 learns the components of the best functions and uses those components to quickly iterate towards an optimum solution.

[0046] In one embodiment, the machine learning processes continues generating candidate functions and test data until either one or more determined candidate functions are found that provide test data that matches the completed fields of the training set data within a predefined margin of error or until the process is terminated.

[0047] Embodiments of the present disclosure generate results data that indicates the best determined candidate functions for each data field of the new and/or updated form, based on how well test data from the best functions match the training set data. Embodiments of the present disclosure can output the results data for review by users who can review and approve the determined functions.

[0048] Additionally, or alternatively, embodiments of the present disclosure can determine when one or more acceptable candidate functions have been found and/or when the new and/or updated form has been entirely learned and can incorporate the new and/or updated form into a user document preparation engine so that users or customers of the electronic document preparation system can utilize the electronic document preparation system to electronically prepare documents involving the learned functions. By utilizing machine learning to learn and incorporate new and/or updated forms, efficiency of the electronic document preparation system is increased.

[0049] In addition, the disclosed method and system for learning and incorporating new and/or updated forms in an electronic document preparation system provides for significant improvements to the technical fields of electronic financial document preparation, data processing, data management, and user experience.

[0050] In addition, as discussed above, the disclosed method and system for learning and incorporating new and/or updated forms in an electronic document preparation system provide for the processing and storing of smaller amounts of data, i.e., more efficiently acquire and analyze forms and data, thereby eliminating unnecessary data analysis and storage. Consequently, using the disclosed method and system for learning and incorporating new and/or updated forms in an electronic document preparation system results in more efficient use of human and non-human resources, fewer processor cycles being utilized, reduced memory utilization, and less communications bandwidth being utilized to relay data to, and from, backend systems and client systems, and various investigative systems and parties. As a result, computing systems are transformed into faster, more efficient, and more effective computing systems by implementing the method and system for learning and incorporating new and/or updated forms in an electronic document preparation system.

[0051] In one embodiment, production environment 100 includes service provider computing environment 110, user computing environment 140, third party computing environment 150, and public information computing environments 160, for learning and incorporating new and/or updated forms in an electronic document preparation system, according to one embodiment. Computing environments 110, 140, 150, and 160 are communicatively coupled to each other with one or more communication channels 101, according to one embodiment.

[0052] Service provider computing environment 110 represents one or more computing systems such as a server or distribution center that is configured to receive, execute, and host one or more electronic document preparation systems (e.g., applications) for access by one or more users, for learning and incorporating new and/or updated forms in an electronic document preparation system, according to one embodiment. Service provider computing environment

110 represents a traditional data center computing environment, a virtual asset computing environment (e.g., a cloud computing environment), or a hybrid between a traditional data center computing environment and a virtual asset computing environment, according to one embodiment.

[0053] Service provider computing environment 110 includes electronic document preparation system 111 configured to provide electronic document preparation services to a user.

[0054] According to various embodiments, electronic document preparation system 111 is a system that assists in preparing financial documents related to one or more of tax return preparation, invoicing, payroll management, billing, banking, investments, loans, credit cards, real estate investments, retirement planning, bill pay, and budgeting. Electronic document preparation system 111 can be a tax return preparation system or other type of electronic document preparation system. Electronic document preparation system 111 can be a standalone system that provides financial document preparation services to users. Alternatively, electronic document preparation system 111 can be integrated into other software or service products provided by a service provider.

[0055] In one embodiment, electronic document preparation system 111 assists users in preparing documents related to one or more forms that include data fields to be completed by the user. The data fields may require data entries in accordance with specified instructions which can be represented by functions. Once the electronic document preparation system has learned functions that produce the required data entries for the data fields, the electronic document preparation system can assist individual users in electronically completing the form.

[0056] In many situations, such as in tax return preparation situations, state and federal governments or other financial institutions issue new or updated versions of standardized forms each year or even several times within a single year. Each time a new and/or updated form is released, electronic document preparation system 111 needs to learn the specific functions that provide the required data entries for one or more data fields in the new and/or updated form.

[0057] If these data fields are not correctly completed, there can be serious financial consequences for users. Furthermore, if electronic document preparation system 111 does not quickly learn and incorporate new and/or updated forms into electronic document preparation system 111, users of the electronic document preparation system 111 may turn to other forms of financial document preparation services. In traditional electronic document preparation systems, new and/or updated forms are learned and incorporated by financial professionals and/or experts manually reviewing the new and/or updated forms and manually revising software instructions to incorporate the new and/or updated forms. In some cases, this can be a slow, expensive, and unreliable system. Manually revising software instructions can take many man hours over many days or weeks, depending on the extent of the changes. Electronic document preparation system 111 of the present disclosure advantageously utilizes machine learning in addition to training set data in order to quickly and efficiently learn functions related to data fields of a form and incorporate those functions into electronic document preparation system 111.

[0058] According to one embodiment, electronic document preparation system 111 receives form data related to a new form or updated version of a previously known form. Electronic document preparation system 111 analyzes the form data and identifies data fields of the form. Electronic document preparation system 111 acquires training set data that is related to the new or updated version of the form. The training set data can include historical data of or related to previously prepared documents including copies of the form, or a related form, with one or more completed data fields. The previously prepared documents can include previously prepared documents that have already been filed and approved with government or other institutions, or that were otherwise validated or approved.

[0059] Additionally, or alternatively, the training set data can include fabricated data that includes previously prepared documents using fictitious data or real data that has been scrubbed of personal identifiers or otherwise altered. Electronic document preparation system 111 utilizes machine learning in combination with the training set data to learn the functions that provide data entries for the data fields of the new and/or updated form.

[0060] In one embodiment, electronic document preparation system 111 identifies one or more dependencies for each data field to be learned. These dependencies can include one or more data values from other data fields of the new and/or updated form, one or more data values from another related form or worksheet, one or more constants, or many other kinds of dependencies that can be included in an acceptable function for a particular data field.

[0061] Electronic document preparation system 111 can identify the one or more possible dependencies based on natural language parsing of the descriptive text included in the new and/or updated form and related to the data field needing a new function to be learned. Electronic document preparation system can identify one or more possible dependencies by analyzing software from previous electronic document preparation systems that processed forms related to the new and/or updated form. Electronic document preparation system 111 can identify possible dependencies by receiving data from an expert, from a third party, or from another source.

[0062] In one embodiment, electronic document preparation system 111 generates, for each data field to be learned, one or more candidate functions based on the one or more dependencies and including one or more operators from a set of operators. Operators may represent any boolean, logical and/or mathematical operation, or any combination thereof.

[0063] In one embodiment, once one or more candidate functions are generated, electronic document preparation system 111 generates test data by applying the candidate functions to the training set data.

[0064] Electronic document preparation system 111 then generates matching data that indicates how closely the test data matches the training set data. When electronic document preparation system 111 finds a candidate function that results in test data that matches or closely matches the training set data within a predetermined margin of error, electronic document preparation system 111 can determine that the candidate function is an acceptable function for the particular data field of the new and/or updated form. In one embodiment, a fitness function is used to determine that one or more candidate functions are acceptable. In one embodiment, the fitness function includes an error function, such as a root mean square error function, reflecting errors that may be present in test data associated with one or more data sets of the training set data, as discussed herein. Other error functions currently known to those of ordinary skill or later developed may be used without departing from the scope of this disclosure. Other components of a fitness function include, according to various embodiments, one or more of how many operators are present in the candidate function, how many operators depend on results of other operators completing prior operations, whether there are missing arguments in the candidate function, and whether an argument is repeated in the candidate function. The tax return preparation system then generates results data indicating whether the candidate function is acceptable and/or a fitness score, determined using a fitness function or an error function, or both, which may be used in a determination of a level of fitness, or a determination of a level of acceptability, for example

[0065] In one embodiment, electronic document preparation system 111 can generate and output results data for review. The results data can include one or more of the candidate functions that are determined to be acceptable functions, according to the matching data, for respective data fields of the new and/or updated form.

[0066] Electronic document preparation system 111 can request input from the expert to approve at least one of the acceptable candidate functions. Additionally, or alternatively, the electronic document preparation system 111 can automatically determine that the candidate function is acceptable, based on the matching data, and update electronic document preparation system 111 without review or approval. In this way, the electronic document preparation system can automatically learn and incorporate new or revised data fields and forms into electronic document preparation system 111.

[0067] Electronic document preparation system 111 includes interface module 112, machine learning module 113, data acquisition module 114, natural language parsing module 115, historical form analysis module 116, and user document preparation engine 117, according to one embodiment.

[0068] Interface module 112 is configured to receive form data 119 related to a new and/or updated form. Interface module 112 can receive the form data 119 from an expert, from a government agency, from a financial institution, or in other ways now known or later developed.

[0069] According to one embodiment, when a new and/or updated form is made available, an expert, other personnel, or other human or nonhuman resources of electronic document preparation system 111 can upload, scan or otherwise provide an electronic version of the form to interface module 112. Interface module 112 can also receive the form data in an automated manner such as by receiving automatic updates or in another way. The electronic version of the form is represented by form data 119. Form data 119 can include one or more PDF documents, one or more HTML documents, one or more text documents, or other types of electronic document formats. The form data can include data related to data fields of the received form, limiting values, tables, or other data related to the new and/or updated form and its data fields that are used in the machine learning process.

[0070] Interface module 112 can also output results data 120 indicating the results of a machine learning process for particular candidate functions. The interface module 112 can also output learned form data 121 related to finalized learned functions, i.e. those functions that have been determined by processes discussed herein and which have been determined to be acceptable within a predetermined margin of error.

[0071] An expert can obtain and review the results data 120 and the learned form data 121 from the interface module 112. Results data 120 or other test data can also be utilized by an expert and/or an automated system to use for other purposes. For example: results data 120 or other test data can be used by electronic document preparation systems to test software instructions of the electronic document preparation system before making functionality associated with the software instructions available to the public.

[0072] The machine learning module 113 analyzes the form data 119 in order to learn functions for the data fields of the new and/or updated form and incorporate them into the electronic document preparation system 111. The machine learning module 113 generates the results data 120 and the learned form data 121.

[0073] In one embodiment, the machine learning module 113 is able to generate and test thousands of candidate functions very rapidly in successive iterations. The machine learning module 113 can utilize one or more algorithms to generate candidate functions based on many factors.

[0074] For example, machine learning module 113 can generate new candidate functions based on previously tested candidate functions.

[0075] In one embodiment, in a system where many candidate functions are generated and tested, components of a predetermined number of candidate functions that match the training data better than other candidate functions are used to generate new candidate functions which are then tested. In one embodiment, a component of a new candidate function includes one or more operators of the candidate function. In one embodiment, a component of a new candidate function includes one or more constants of the candidate function. In one embodiment, a component of a new candidate function includes one or more dependencies used to generate the candidate function.

[0076] In one embodiment, one or more of the predetermined number of candidate functions that match the training data better than other candidate functions are split into two or more components each, and the split components recombined into new candidate functions that are then tested to determine how well test data generated from those new candidate functions match the training set data. One or more of those new candidate functions that are determined to generate test data that match the training set data better than the original candidate functions may then again be split, if desired, and recombined into a second set of new candidate functions, and so on, until the resulting candidate functions produce test data that are deemed to match the training set data within a predetermined margin of error, as discussed herein. Thus, machine learning module 113 learns the components of the best functions and uses those components to quickly iterate towards an optimum solution. The machine learning module 113 can utilize analysis of the form data and/or other data to learn the best components of the candidate functions for a particular data field and can generate candidate functions based on these best components.

[0077] In one embodiment, the electronic document preparation system 111 uses data acquisition module 114 to acquire training set data 122. Training set data 122 includes, in various embodiments, previously prepared documents for one or more previous users of the electronic document preparation system 111 and/or fictitious users of the electronic document preparation system 111. The training set data 122 can be used by the machine learning module 113 in order to learn and incorporate the new and/or updated form into the electronic document preparation system 111.

[0078] In one embodiment, training set data 122 includes historical data 123 related to previously prepared documents or previously filed forms of one or more users. The historical data 123 can include, for each of a number of previous users of the electronic document preparation system 111, a respective completed or partially completed copy of the new and/or updated form or a completed or partially completed copy of a form related to the new and/or updated form. The copies of the form include data values in at least the data fields for which one or more functions are to be determined.

[0079] In one embodiment, the training set data 122 includes fabricated data 124. The fabricated data 124 can include copies of the new and/or updated form that were previously filled using fabricated data. The fabricated data can include real data from previous users or other people but that has been scrubbed of personal identifiers or otherwise altered. Further, the fabricated data can include data that matches the requirements of each data field, but which may not have been used in a filing of a formal document with the authorities, such as with the Internal Revenue Service.

[0080] In one embodiment, the historical data 123 and/or the fabricated data 124 also includes related data used to complete the forms and to prepare the historical document, such as one or more worksheets or other subcomponents that are used to determine data values of one or more data fields of the training set data. The historical data 123 can include previously prepared documents that include or use completed form data which were filed with and/or approved by a government or other institution. In this way, a large portion of historical data 123 is likely highly accurate and properly prepared, though some of the previously prepared documents will inevitably include errors. Typically, the functions for computing or obtaining the proper data entry for a data field of a form can include data values from other forms related to each other and sometimes complex ways. Thus, the historical data 123 can include, for each historical user in the training set data, a final version of a previously prepared document, the form that is related to the new and/or updated form to be learned, other forms used to calculate the values for the related form, and other sources of data for completing the related form.

[0081] In one embodiment, the electronic document preparation system 111 is a financial document preparation system. In this case, the historical data 123 includes historical financial data. The historical financial data can include, for one or more historical users of the electronic document preparation system 111, data representing one or more items associated with various users, i.e. the subjects of the electronic forms, such as, but not limited to, one or more of a name of the user, a name of the user's employer, an employer identification number (EID), a job title, annual income, salary and wages, bonuses, a Social Security number, a government identification, a driver's license number, a date of birth, an address, a zip code, home ownership status, marital status, W-2 income, an employer's address, spousal information, children's information, asset information, medical history, occupation, information regarding dependents, salary and wages, interest income, dividend income, business income, farm income, capital gain income, pension income, IRA distributions, education expenses, health savings account deductions, moving expenses, IRA deductions, student loan interest, tuition and fees, medical and dental expenses, state and local taxes, real estate taxes, personal property tax, mortgage interest, charitable contributions, casualty and theft losses, unreimbursed employee expenses, alternative minimum tax, foreign tax credit, education tax credits, retirement savings contribution, child tax credits, residential energy credits, item name and description, item purchase cost, date of purchase, and any other information that is currently used, that can be used, or that are used in the future, in a financial document preparation system or in the preparation of financial documents such as a user's tax return, according to various embodiments.

[0082] In one embodiment, the data acquisition module 114 is configured to obtain or retrieve historical data 123 from one or more sources, including a large number of sources, e.g. 100 or more. The data acquisition module 114 can retrieve, from databases of the electronic document preparation system 111, historical data 123 that has been previously obtained by the electronic document preparation system 111 from third-party institutions. Additionally, or alternatively, the data acquisition module 114 can retrieve the historical data 123 afresh from the third-party institutions.

[0083] In one embodiment, data acquisition module 114 can also supply or supplement historical data 123 by gathering pertinent data from other sources including third party computing environment 150, public information computing environment 160, additional service provider systems 135, data provided from historical users, data collected from user devices or accounts of electronic document preparation system 111, social media accounts, and /or various other sources to merge with or supplement historical data 123, according to various embodiments.

[0084] Data acquisition module 114 can gather additional data including historical financial data and third party data. For example, data acquisition module 114 is configured to communicate with additional service provider systems 135, e.g., a tax return preparation system, a payroll management system, or other electronic document preparation system, to access financial data 136, according to one embodiment. Data acquisition module 114 imports relevant portions of the financial data 136 into the electronic document preparation system 111 and, for example, saves local copies into one or more databases, according to one embodiment.

[0085] In one embodiment, the additional service provider systems 135 include a personal electronic document preparation system, and the data acquisition module 114 is configured to acquire financial data 136 for use by the electronic document preparation system 111 in learning and incorporating the new or updated form into the electronic document preparation system 111. Because the service provider provides both the electronic document preparation system 111 and, for example, the additional service provider systems 135, the service provider computing environment 110 can be configured to share financial information between the various systems. By interfacing with the additional service provider systems 135, the data acquisition module 114 can automatically and periodically supply or supplement the historical data 123 from the financial data 136. The financial data 136 can include income data, investment data, property ownership data, retirement account data, age data, data regarding additional sources of income, marital status, number and ages of children or other dependents, geographic location, and other data that indicates personal and financial characteristics of users of other financial systems, according to one embodiment.

[0086] The data acquisition module 114 is configured to acquire additional information from various sources to merge with or supplement training set data 122, according to one embodiment. For example, the data acquisition module 114 is configured to gather historical data 123 from various sources. For example, the data acquisition module 114 is configured to communicate with additional service provider systems 135, e.g., a tax return preparation system, a payroll management system, or other financial management system, to access financial data 136, according to one embodiment. The data acquisition module 114 imports relevant portions of the financial data 136 into the training set data 122 and, for example, saves local copies into one or more databases, according to one embodiment.

[0087] The data acquisition module 114 is configured to acquire additional financial data from the public information computing environment 160, according to one embodiment. The training set data can be gathered from public record searches of tax records, public information databases, property ownership records, and other public sources of information. The data acquisition module 114 can also acquire data from sources such as social media websites, such as Twitter, Facebook, LinkedIn, and the like.

[0088] The data acquisition module 114 is configured to acquire data from third parties, according to one embodiment. For example, the data acquisition module 114 requests and receives third party data from the third party computing environment 150 to supply or supplement the training set data 122, according to one embodiment. In one embodiment, the third party computing environment 140 is configured to automatically transmit financial data to the electronic document preparation system 111 (e.g., to the data acquisition module 114), to be merged into training set data 122. The third party computing environment 140 can include, but is not limited to, financial service providers, state institutions, federal institutions, private employers, financial institutions, social media, and any other business, organization, or association that has maintained financial data, that currently maintains financial data, or which may in the future maintain financial data, according to one embodiment.

[0089] In one embodiment, the electronic document preparation system 111 utilizes the machine learning module 113 to learn the data fields of the new and/or updated form in conjunction with training set data 122. The machine learning module 113 generates candidate functions for one or more data fields of the new and/or updated form to be learned and applies the candidate functions to the training set data 122 in order to find an acceptable candidate function that produces data values that match or closely match data values of the corresponding data fields of the training set data 122.

[0090] In one embodiment, in a system wherein many candidate functions are generated and tested, components of a predetermined number of candidate functions that match the training data better than other candidate functions are used to generate new candidate functions which are then tested. In one embodiment, a component of a new candidate function includes one or more operators of the candidate function. In one embodiment, a component of a new candidate function includes one or more constants of the candidate function. In one embodiment, a component of a new candidate function includes one or more dependencies used to generate the candidate function.

[0091] In one embodiment, one or more of the predetermined number of candidate functions that match the training data better than other candidate functions are split into two or more components each, and the split components recombined into new candidate functions that are then tested to determine how well test data generated from those new candidate functions match the training set data. One or more of those new candidate functions that are determined to generate test data that match the training set data better than the original candidate functions may then again be split, if desired, and recombined into a second set of new candidate functions, and so on, until the resulting candidate functions produce test data that are deemed to match the training set data within a predetermined margin of error, as discussed herein. Thus, machine learning module 113 learns the components of the best functions and uses those components to quickly iterate towards an optimum solution.

[0092] In one embodiment, the electronic document preparation system 111 identifies dependency data 129 including one or more possible dependencies for one or more data fields to be learned. These possible dependencies can include one or more data values from other data fields of the new and/or updated form, one or more data values from another related form or worksheet, one or more constants, or many other kinds of possible dependencies that can be included in an acceptable function for a particular data field.

[0093] In one embodiment, the machine learning module 113 generates candidate functions based on the dependency data 129 and one or more operators selected from a set of operators. The operators can include arithmetic operators such as addition, subtraction, multiplication, or division operators; logical operators such as if-then operators; existence condition operators that depend on the existence of a data value in another data field of new and/or updated form, in a form other than the new and/or updated form, or in some other location or data set; and string comparisons including greater than, less than and equal to, among others. Each candidate function can include one or more of the operators operating on one or more of the possible dependencies.

[0094] In one embodiment, the machine learning module 113 learns acceptable functions for various data fields of a given form one at a time. In other words, if the form data 119 indicates that a form has ten data fields to be learned, the machine learning module 113 will begin by learning an acceptable function for a first data field of the new and/or updated form before learning acceptable functions for other data fields of the same form. In particular, the machine learning module 113 will generate candidate function data 125 corresponding to one or more candidate functions for the first data field of the new and/or updated form as represented by the form data 119.

[0095] The machine learning module 113 also receives training set data 122 from the data acquisition module 114. The training set data 122 includes data related to previously completed copies of an older version of the form to be learned or previously completed copies of a form closely related to the new and/or updated form to be learned. In particular, the training set data 122 includes copies of the form that have a data entry in the data field that corresponds to the data field of the new and/or updated form currently being analyzed and learned by the machine learning module 113. The training set data 122 also includes data that was used to calculate the data values in the data field for each copy of the form or for each copy of the related form, e.g. W-2 data, income data, data related to other forms such as tax forms, payroll data, personal information, or any other kind of information that was used to complete the copies of the form or the copies of the related form in the training set data 122. The machine learning module 113 generates test data 126 by applying each of the candidate functions to the training set data for the particular data field currently being learned. In particular, for each copy of the form or related form in the training set data 122, the machine learning module 113 applies the candidate function to at least a portion of the training set data related to the data field being learned in order to generate a test data value for the data field. Thus, if the training set data 122 includes 1000 completed copies of the new and/or updated form or a related form, then machine learning module 113 will generate test data 126 that includes one test data value for the particular data field being analyzed for at least a portion of the thousand completed copies.

[0096] In one embodiment, the machine learning module 113 then generates matching data 127 by comparing the test data value for each copy of the form to the actual data value from the completed data field of that copy of the form. The matching data 127 indicates how many of the test data values match their corresponding completed data value from the training set data 122 within a predetermined margin of error.

[0097] In one embodiment, a fitness function is used to determine that one or more candidate functions are acceptable. In one embodiment, the fitness function includes an error function, such as a root mean square error function, reflecting errors that may be present in test data associated with one or more data sets of the training set data, as discussed herein. Other error functions currently known to those of ordinary skill or later developed may be used without departing from the scope of this disclosure. Other components of a fitness function include, according to various embodiments, one or more of how many operators are present in the candidate function, how many operators depend on results of other operators completing prior operations, whether there are missing arguments in the candidate function, and whether an argument is repeated in the candidate function. The tax return preparation system then generates results data indicating whether the candidate function is acceptable and/or a fitness score, determined using a fitness function or an error function, or both, which may be used in a determination of a level of fitness, or a determination of a level of acceptability, for example.

[0098] As explained above, in a system wherein many candidate functions are generated and tested, components of a predetermined number of candidate functions that match the training data better than other candidate functions are used to generate new candidate functions which are then tested. In one embodiment, a component of a new candidate function includes one or more operators of the candidate function. In one embodiment, a component of a new candidate function includes one or more constants of the candidate function. In one embodiment, a component of a new candidate function includes one or more dependencies used to generate the candidate function.

[0099] In one embodiment, one or more of the predetermined number of candidate functions that match the training data better than other candidate functions are split into two or more components each, and the split components recombined into new candidate functions that are then tested to determine how well test data generated from those new candidate functions match the training set data. One or more of those new candidate functions that are determined to generate test data that match the training set data better than the original candidate functions may then again be split, if desired, and recombined into a second set of new candidate functions, and so on, until the resulting candidate functions produce test data that are deemed to match the training set data within a predetermined margin of error, as discussed herein. Thus, machine learning module 113 learns the components of the best functions and uses those components to quickly iterate towards an optimum solution.

[0100] It is expected that the training set data 122 may include some errors in the completed data values for the data field under test. Thus, an acceptable function operating on the test data may result in test data 126 that does not perfectly match the completed data fields in the training set data 122. Thus, an acceptable candidate function will at least result in test data that matches the training set data within a predefined margin of error.

[0101] In one embodiment, a fitness function is used to determine that one or more candidate functions are acceptable. In one embodiment, the fitness function includes an error function, such as a root mean square error function, reflecting errors that may be present in test data associated with one or more data sets of the training set data, as discussed herein. Other error functions currently known to those of ordinary skill or later developed may be used without departing from the scope of this disclosure. Other components of a fitness function include, according to various embodiments, one or more of how many operators are present in the candidate function, how many operators depend on results of other operators completing prior operations, whether there are missing arguments in the candidate function, and whether an argument is repeated in the candidate function. The tax return preparation system then generates results data indicating whether the candidate function is acceptable and/or a fitness score, determined using a fitness function or an error function, or both, which may be used in a determination of a level of fitness, or a determination of a level of acceptability, for example.

[0102] In one embodiment, as discussed herein, the machine learning module 113 will continue to generate and test candidate functions until a candidate function has been found that results in test data that matches the training set data 122 within the predefined margin of error. When at least one acceptable function has been found for the first data field, the machine learning module 113 can repeat this process for a second data field, and so on, for each data field of the new and/or updated form to be learned.

[0103] In one embodiment, the machine learning module 113 generates and tests candidate functions one at a time. Each time the matching data 127 for a candidate function does indicates an error that exceeds the predefined margin of error, i.e. that the candidate function is not acceptable, the machine learning module 113 may generate a new candidate function and tests the new candidate function.

[0104] In one embodiment, to form one or more new candidate functions, components of a predetermined number of previously formed candidate functions that match the training data better than other candidate functions, but perhaps not enough to be determined acceptable functions, are used to generate new candidate functions which are then tested. In one embodiment, a component of a new candidate function includes one or more operators of the previously formed candidate function. In one embodiment, a component of a new candidate function includes one or more constants of the previously formed candidate function. In one embodiment, a component of a new candidate function includes one or more dependencies used to generate the previously formed candidate function.

[0105] In one embodiment, one or more of the predetermined number of candidate functions that match the training data better than other candidate functions are split into two or more components each, and the split components recombined into new candidate functions that are then tested to determine how well test data generated from those new candidate functions match the training set data. One or more of those new candidate functions that are determined to generate test data that match the training set data better than the original candidate functions may then again be split, if desired, and recombined into a second set of new candidate functions, and so on, until one or more resulting candidate functions produce test data that are deemed to match the training set data within a predetermined margin of error, as discussed herein. Thus, machine learning module 113 learns the components of the best functions and uses those components to quickly iterate towards an optimum solution.

[0106] The machine learning module 113 can continue this process until an acceptable candidate function has been found. In this way, the machine learning module 113 generates one or more acceptable candidate functions sequentially for each data field under test.

[0107] In one embodiment, the machine learning module 113 can first generate candidate functions and then test each of the generated candidate functions. If the matching data 127 indicates that none of the generated candidate functions is an acceptable candidate function, then the machine learning module 113 can generate additional candidate functions and apply them to the training set data 122. The machine learning module 113 can continue generating candidate functions and applying them to the training set data until an acceptable function has been found.

[0108] In one embodiment, the machine learning module generates candidate functions in successive iterations based on one or more algorithms. The successive iterations can be based on whether the matching data indicates that the candidate functions are becoming more accurate, such as in the successive iteration algorithm discussed herein where previously tested candidate functions are split into two or more components and recombined into new candidate functions. The machine learning module can continue to make adjustments to the candidate functions in directions that make the matching data more accurate until at least one acceptable function has been found.

[0109] In one embodiment, the machine learning module 113 generates confidence score data 128 based on the matching data 127. The confidence score data 128 can be based on the matching data 127 and data regarding the candidate function itself. For example, the confidence score is adjusted downward, indicating that a less desirable candidate function has been found, if the candidate function uses an operator twice. The confidence score may further be adjusted downward, indicating that a less desirable candidate function has been found, for longer candidate functions, i.e. those functions having more operators. The confidence score may further be adjusted downward or upward based on how quickly a candidate function performs in its entirety. Other such adjustments may be used without departing from the teachings presented herein.

[0110] In one embodiment, the machine learning module 113 generates results data 120. The results data 120 can include matching data 127 and/or confidence score data 128 for each candidate function that has been tested for one or more particular data fields of the new and/or updated form to be learned. Alternatively, the results data 120 can include data indicating that one or more of the candidate functions is possibly acceptable based on the matching data 127 and/or the confidence score 128. Alternatively, the results data 120 can indicate that at least one acceptable function has been found. The results data 120 can also indicate what the acceptable function is. Results data 120 can be provided to the interface module 112. The interface module 112 can output the results data 120 to a user, an expert, or other personnel for review and/or approval.

[0111] In one embodiment, the machine learning module 113 outputs results data 120 indicating that a candidate function has been determined that is likely acceptable. The results data 120 can indicate what the determined candidate function is, the matching data 127 or confidence score data 128 related to the determined candidate function, or any other information that will be useful for review by an expert. The machine learning module 113 can cause the interface module 112 to prompt expert user or other individual to review the results data 120 and to approve the determined candidate function as acceptable or to indicate that the determined candidate function is not acceptable and that the machine learning module 113 should continue generating candidate functions for the data field currently under consideration. The machine learning module 113 awaits input from the expert or other personnel approving the candidate function. If the candidate function is approved by the expert or other personnel, the machine learning module 113 determines that the acceptable candidate function has been found and moves on to finding an acceptable candidate function for a next data field of the new and/or updated form.

[0112] In one embodiment, the machine learning module 113 does not wait for the approval of an expert before determining that an acceptable candidate function was found. Instead, when the machine learning module 113 determines that an acceptable candidate function has been found based on the matching data, the confidence score data 128, and/or other criteria, the machine learning module 113 incorporates the acceptable candidate function and moves onto another data field of the new and/or updated form.

[0113] In one embodiment, when the machine learning module 113 has learned an acceptable candidate function for data fields of the new and/or updated form that needed to be learned, then the machine learning module 113 generates learned form data 121. The learned form data 121 indicates that the new and/or updated form has been learned. The learned form data 121 can also indicate what the acceptable candidate functions are for one or more of the data fields of the new and/or updated form. The interface module 112 can output the learned form data 121 for review and/or approval by a user or expert. In one embodiment, once the user, expert or other personnel has approved the learned form data 121, the machine learning module 113 ceases analysis of the new and/or updated form and awaits form data 119 related to another form to be learned.

[0114] In one embodiment, the electronic document preparation system 111 includes a user document preparation engine 117. The user document preparation engine 117 assists users of the electronic document preparation system 111 to prepare a financial document based on or including the newly learned form as well as other forms. The user document preparation engine 117 includes current document instructions data 131. The current document instructions data 131 includes software instructions, modules, engines, or other data or processes used to assist users of the electronic document preparation system 111 in electronically preparing a document.

[0115] In one embodiment, once the machine learning module 113 has fully learned one or more acceptable candidate functions for the data fields of a new and/or updated form, the machine learning module 113 incorporates the newly learned form into the electronic document preparation system 111 by updating the current document instructions data 131. When the current document instructions data 131 has been updated to include and recognize the new and/or updated form, then users of the electronic document preparation system can electronically complete the new and/or updated form using electronic document preparation system 111. In this way, the electronic document preparation system 111 quickly provides functionality that electronically complete the data fields of the new and/or updated form as part of preparing a financial document.

[0116] In one embodiment, the user computing environment 140 is a computing environment related to a user of the electronic document preparation system 111. The user computing environment 140 includes input devices 141 and output devices 142 for communicating with the user, according one embodiment. The input devices 141 include, but are not limited to, keyboards, mice, microphones, touchpads, touchscreens, digital pens, and the like. The output devices 142 include, but are not limited to, speakers, monitors, touchscreens, and the like. The output devices 142 can display data related to the preparation of the financial document.

[0117] In one embodiment, the machine learning module 113 can also generate interview content to assist in a financial document preparation interview. As a user utilizes the electronic document preparation system 111 to prepare a financial document, the user document preparation engine 117 may guide the user through a financial document preparation interview in order to assist the user in preparing the financial document. The interview content can include graphics, prompts, text, sound, or other electronic, visual, or audio content that assists the user to prepare the financial document. The interview content can prompt the user to provide data, to select relevant forms to be completed as part of the financial document preparation process, to explore financial topics, or otherwise assist the user in preparing the financial document. When the machine learning module 113 learns acceptable functions for one or more data fields of a form, the machine learning module 113 can also generate text or other types of audio or video prompts that describe the function and that can prompt the user to provide information that the user document preparation engine 117 will use to complete the form. Thus, the machine learning module 113 can generate interview content to assist in a financial document preparation interview.

[0118] In one embodiment, the machine learning module 113 updates the current document instruction data 131 once a new and/or updated form has been entirely learned without input or approval of an expert or other personnel. In one embodiment, the machine learning module 113 updates the current document instructions data 131 only after an expert has given approval that the new and/or updated form has properly learned.

[0119] In one embodiment, the machine learning module 113 only learns acceptable functions for selected fields of a new and/or updated form. For example, the machine learning module 113 is configured to perform machine learning processes to learn acceptable functions for certain types of data fields. Some types of data fields may not be as conducive to machine learning processes or for other reasons the machine learning module 113 is configured to learn acceptable functions for only particular data fields of a new and/or updated form. In these cases, the machine learning module 113 will only learn acceptable functions for certain selected data fields of the new and/or updated form. In some cases, the machine learning module 113 may determine that it is unable to learn an acceptable function for one or more data fields after generating and testing many candidate functions for the one or more data fields. The results data 120 can therefore include data indicating that an acceptable function for a particular data field of the new and/or updated form cannot be learned by the machine learning module 113.

[0120] In one embodiment, once the form data 119 has been provided to the electronic document preparation system 111, a user, expert or other personnel can input an indication of which data fields of the new and/or updated form should be learned by the machine learning module 113. The machine learning module 113 will then only learn acceptable functions for those fields of the new and/or updated form that have been indicated by the user, expert or other personnel. In one embodiment, the form data 119 can indicate which data fields the machine learning module 113 should consider. In this way, the machine learning module 113 only attempts to learn acceptable functions for the indicated data fields of a new and/or updated form.

[0121] In one embodiment, an acceptable function for a data field is simple or complex. A complex function may require that multiple data values be gathered from multiple places within other forms, the same form, from a user, or from other locations or databases. A complex function may also include mathematical relationships that will be applied to the multiple data values in complex ways in order to generate the proper data value for the data field. A function may include finding the minimum data value among two or more data values, finding the maximum data value among two or more data values, addition, subtraction, multiplication, division, exponential functions, logic functions, existence conditions, string comparisons, etc. The machine learning module 113 can generate and test complex candidate functions until an acceptable function has been found for a particular data field.

[0122] In one embodiment, new and/or updated forms may include data fields that expect data values that are alphabetical such as a first name, a last name, a middle name, a middle initial, a company name, a name of a spouse, a name of a child, a name of a dependent, a home address, a business address, a state of residence, the country of citizenship, or other types of data values that are generally alphabetic. In these cases, An acceptable function may include a person, a last name, a middle name, a middle initial, a company name, a name of a spouse, a name of a child, a name of a defendant, a home address, a business address, a state residence, the country citizenship, or other types of alphabetic data values. An acceptable function can also include a location from which these alphabetic data values are retrieved in other forms, worksheets, or financial related data otherwise provided by users or gathered from various sources.

[0123] The forms may also include data fields that expect data values that are numeric by nature. These expected data values may include incomes, tax withholdings, Social Security numbers, identification numbers, ages, loan payments, interest payments, charitable contributions, mortgage payments, dates, or other types of data values that are typically numeric in nature.

[0124] In one embodiment, the machine learning module 113 can generate candidate functions for a particular data field based on dependency data that can provide an indication of the types of data that are likely to be included in an acceptable function and their likely location in other forms or data. For example, the machine learning module 113 can utilize historical document instructions data 130, natural language parsing data 118, current document instruction data 121, and other types of contextual clues or hints in order to find a likely starting place for generating candidate functions. For this reason, the electronic document preparation system 111 can include a natural language parsing module 115 and the historical form analysis module 116.

[0125] In one embodiment, the natural language parsing module 115 analyzes the form data 119 with a natural language parsing process. In particular, the natural language parsing module analyzes the text description associated with data fields of the new and/or updated form to be learned. For example, the form data 119 may include text descriptions and/or form text for various data fields of the new and/or updated form. The text descriptions and form text originate from one or more different sources, such as, in the case of the new and/or updated for being a U.S. text form, from the IRS. The text descriptions and form text include, in one embodiment, text of one or more actual tax forms issued by the IRS and required to be filled out by taxpayers for which the new and/or updated form applies. The text descriptions and form text further include, in various embodiments, text of one or more instruction sets and publications issued by the IRS to assist the tax payer or tax preparer properly complete the form. The natural language parsing module 115 analyzes these text descriptions through process described herein and generates natural language parsing data 118 indicating the type of data value expected in each data field as well as function data indicating a hierarchical function representation formed as nodes and leaves of a tree. In various embodiments, the leaves of the function representation includes one or more form dependencies, such as constants, variables, and form/line dependencies where the function represented by the function representation depends on a results from data value associated with one or more different lines of the same form being analyzed, from a data value determined from a worksheet, or from one or more data values associated with one or more lines of a different tax form. The natural language parsing module 115 provides the natural language parsing data 118 to the machine learning module 113. The machine learning module 113 generates candidate functions for the various data fields based on the natural language parsing data 118. In this way, the machine learning module 113 utilizes the natural language parsing data 118 to assist in the machine learning process.

[0126] In one embodiment, the historical form analysis module 116 analyzes the form data 119 in order to determine if it is likely that previous versions of the electronic document preparation system 111 included software instructions that computed data values for data fields of historical forms that are similar to the new and/or updated form. Accordingly, the historical form analysis module 116 analyzes the historical document instruction data 130 that includes software instructions from previous versions of the electronic document preparation system 111. Because it is possible that the previous versions of the electronic document preparation system utilized software languages or structures that are now obsolete, the historical document instructions data 130 cannot easily or simply be analyzed or imported into the current document instructions data 131. For this reason, the historical form analysis module 116 can analyze the historical document instructions data 130 related to historical forms that are similar to the new and/or updated form. Such historical forms may include previous versions of the new and/or updated form. The historical form analysis module 116 can identify from the outdated software language portions of or complete acceptable functions related to data fields of the historical forms and can generate historical instruction analysis data that indicates portions of or complete acceptable functions for the previous version of the form. The machine learning module 113 can utilize these instructions in order to find a starting point for generating the candidate functions in order to learn functions of data fields of the new and/or updated form.

[0127] In some cases, a new and/or updated form is nearly identical to a previous known version of the form. In these cases, the training set data 122 can include historical data 123 that relates to previously prepared, filed, and/or approved financial documents that included or based on the previous known form. In these cases, the data acquisition module 114 will gather a training set data 122 that includes one or more previously completed copies of the previous version of the form. The machine learning module 113 generates the candidate functions and applies them to the training set data as described previously.

[0128] In some cases, a new and/or updated form may include data fields that are different enough that no analogous previously prepared financial documents are available to assist in the machine learning process. In one embodiment, the data acquisition module 114 gathers training set data 122 that includes fabricated financial data 124. The fabricated financial data 124 can include copies of the new and/or updated form prepared with fabricated financial data by a third-party organization or a processor system associated with the service provider computing environment 110. The fabricated financial data 124 can be used by the machine learning module 113 in the machine learning process for learning acceptable functions associated with the data fields of the new and/or updated form. In such a case, the machine learning module generates candidate functions and applies them to the training set data 122 including the fabricated financial data 124 as described previously.

[0129] In one embodiment, the training set data 122 can include both historical data 123 and fabricated financial data 124. In some cases, the historical data 123 can include previously prepared documents as well as previously fabricated financial documents based on fictitious or real financial data.

[0130] In one embodiment, the data acquisition module 114 gathers new training set data 122 each time a new data field of the new and/or updated form is to be analyzed by the machine learning module 113. The data acquisition module 114 can gather a large training set data 122 including many thousands or millions of previously prepared or previously fabricated financial documents. When a new data field of a new and/or updated form is to be learned by the machine learning module 113, the data acquisition module 114 will gather training set data 122, or subset of the training set data 122, that includes a number of previously prepared financial documents that each have a data value in a data field of a form that corresponds to the data field of the new and/or updated form that is currently being learned by the machine learning module 113. In some cases, the training set data 122 can include millions of previously prepared financial documents, only a few hundred or thousands of the previously prepared documents are typically needed for analysis by the machine learning module 113. Thus, the data acquisition module 114 can gather training set data that is appropriate and efficient for the machine learning module 113 to use the learning the current data field of the new and/or updated form.

[0131] In one embodiment, the electronic document preparation system 111 is a tax return preparation system. Preparing a single tax return can require many government tax forms, internal worksheets used by the tax return preparation system in preparing a tax return, W-2 forms, and many other types of forms or financial data pertinent to the preparation of a tax return preparation system. For each tax return that is prepared for a user, the tax return preparation system maintains copies of various tax forms, internal worksheets, data provided by the user and any other relevant financial data used to prepare the tax return. Thus, the tax return preparation system typically maintains historical tax return data related to a large number of previously prepared tax returns. The tax return preparation system can utilize the historical tax return data to gather or generate relevant training set data 122 that can be used by the machine learning module 113.

[0132] In one embodiment, a state or federal agency releases a new tax form that is simply a new version of a previous tax form during tax return preparation season. The form data 119 corresponds to an electronic version of the new version of the tax form. One or more of the data fields of the new tax form is similar to those of the previous tax form. The machine learning module 113 begins to learn the new tax form starting with a first selected data field of the new tax form. The first selected data field corresponds to a first selected line of the new tax form, not necessarily line 1 of the new tax form. The machine learning module 113 causes the data acquisition module 114 to gather training set data 122 that includes a number of previously prepared tax returns and tax related data associated with the previously prepared tax returns. In particular, training set data 122 will include previously prepared tax returns that use the previous version of the new and/or updated form. The machine learning module 113 generates a plurality of candidate functions for the first selected data field and applies them to the training set data 122. For each candidate function, the machine learning module generates matching data 127 and/or confidence score data 128 indicating how well the test data 126 matches the training set data 122. The machine learning module 113 generates results data 120 indicating the matching data 127 and/or the confidence score data 128 of one or more of the candidate functions. The results data 120 can also indicate whether a candidate function is deemed to be an acceptable function for the first selected data field. If candidate functions have been tested and have not been deemed acceptable, additional new candidate functions are formed, with one or more of those new candidate functions being formed from components of one or more of the previous candidate functions.

[0133] In one embodiment, to form one or more new candidate functions, components of a predetermined number of previously formed candidate functions that match the training data better than other candidate functions, but perhaps not enough to be determined acceptable functions, are used to generate new candidate functions which are then tested. In one embodiment, a component of a new candidate function includes one or more operators of the previously formed candidate function. In one embodiment, a component of a new candidate function includes one or more constants of the previously formed candidate function. In one embodiment, a component of a new candidate function includes one or more dependencies used to generate the previously formed candidate function.

[0134] In one embodiment, one or more of the predetermined number of candidate functions that match the training data better than other candidate functions are split into two or more components each, and the split components recombined into new candidate functions that are then tested to determine how well test data generated from those new candidate functions match the training set data. One or more of those new candidate functions that are determined to generate test data that match the training set data better than the original candidate functions may then again be split, if desired, and recombined into a second set of new candidate functions, and so on, until one or more resulting candidate functions produce test data that are deemed to match the training set data within a predetermined margin of error, as discussed herein. Thus, machine learning module 113 learns the components of the best functions and uses those components to quickly iterate towards an optimum solution.

[0135] The machine learning module 113 moves onto a second selected data field after an acceptable function has been found for the first selected data field. In one embodiment, the data fields correspond to selected lines of the new tax form. The machine learning module 113 continues in this manner until functions relating to all selected data fields of the new tax form have been learned. Machine learning module 113 then generates learned form data 121 indicating that all selected fields of the new and/or updated form have been learned. The interface module 112 can present results data 120 or learned form data 121 for review and/or approval by an expert or other personnel. Alternatively, the machine learning module 113 can move from one data field to the next data field without approval or review by an expert, as explained herein.

[0136] In one embodiment, the tax return preparation system receives form data 119 corresponding to a new and/or updated form for which an adequate previously known form cannot be found. In this case, data acquisition module 114 gathers training set data that can include fabricated financial data 124. The fabricated financial data 124 can include fictitious previously prepared tax returns and fabricated financial data that was used to prepare them. The data acquisition module 114 can obtain the fabricated financial data 124 from one or more third parties, one or more associated tax return preparation systems, or in any other way. For example, the tax return preparation system can generate fabricated financial data and provide it to one or more third parties to prepare a fabricated tax return using the new tax form. The fabricated financial data can include data related to real users of the tax return preparation system, a script of actual identifiers such as real names, real Social Security numbers, etc. The third parties can then prepare tax returns from the fabricated financial data using the new and/or updated form.

The third parties can then provide the fabricated tax returns to the tax return preparation system. The tax return preparation system can then utilize the fabricated financial data 124 in conjunction with the machine learning module 113 to learn the functions for the data fields of the new and/or updated form.

[0137] In one specific illustrative example, the tax return preparation system receives form data 119 related to a new tax form. The data acquisition module 114 gathers training set data 122 that at least includes historical tax return data related to previously prepared tax returns and or fabricated historical tax return data related to fabricated tax returns using the new form. In this example, machine learning module 113 undertakes to learn an acceptable function for generating the data value required by line 3 of the new tax form. The machine learning module 113 uses at least a portion of the dependency data that indicates that an acceptable function for line 3 is likely based on the values of line 31, line 2c, and the constants 3000 and 6000.

[0138] The training set data 122 includes previously completed copies of the new form or a related form having data values for line 3 that are believed to be correct. The training set data 122 also includes, in one embodiment, tax related data that were used to prepare the previously completed copies.

[0139] The machine learning module 113 generates at least one candidate function for line 3 of the new form and applies the candidate function(s) to the training set data 122. In particular, the machine learning module 113 generates test values of test data 126 by at least substituting at least a portion of the training set data for one or more of lines 31, 2c and the two constants, 3000 and 6000 in the candidate function for each subset of training set data for one or more of the previously completed copies, resulting in test values for line 3 of previously completed copies of the new or related form. The machine learning module 113 generates matching data by comparing the resulting test values to the actual completed data values for line 3 from the training set data 122. The matching data 127 indicates how well the various test values match the actual values in line 3 of the previously completed forms. Thus, the comparison may include determining a margin of error relating to how well the test values match the actual values, or may include a straight comparison, such as subtracting one value from the other, or may include a more complex comparison, as desired by an implementer of the process operations discussed herein.

[0140] In one embodiment, a fitness function is used to determine that one or more candidate functions are acceptable. In one embodiment, the fitness function includes an error function, such as a root mean square error function, reflecting errors that may be present in test data associated with one or more data sets of the training set data, as discussed herein. Other error functions currently known to those of ordinary skill or later developed may be used without departing from the scope of this disclosure. Other components of a fitness function include, according to various embodiments, one or more of how many operators are present in the candidate function, how many operators depend on results of other operators completing prior operations, whether there are missing arguments in the candidate function, and whether an argument is repeated in the candidate function. The tax return preparation system then generates results data indicating whether the candidate function is acceptable and/or a fitness score, determined using a fitness function or an error function, or both, which may be used in a determination of a level of fitness, or a determination of a level of acceptability, for example.

[0141] If the matching data 127 indicates that at least portions of test data 126 matches the training set data 122 within a predefined margin of error, then the machine learning module 113 determines that the candidate function is acceptable. In the example, after one or more iterations of generating and testing candidate functions, the machine learning module may conclude that an acceptable function for line 3 is that if line 31 exists, then line 3 will be equal to line 31. Alternatively, if line 31 does not exist, then line 3 is the minimum of 6000 or 3000 multiplied by the value from line 2c.

[0142] In one embodiment, machine learning module 113 can also generate confidence score data 128 indicating a level of confidence that the candidate function is acceptable. Machine learning module 113 generates results data 120 that indicate that the candidate function is likely an acceptable function. Interface module 112 outputs results data 120 for review and/or approval by expert, other personnel, or other human and/or nonhuman resources. The expert or other personnel can approve the candidate function, causing machine learning module 113 to move to the next selected line of the new tax form. Alternatively, machine learning module 113 can decide that the candidate function is acceptable without approval from an expert or other personnel and can move onto the next selected line of the new tax form.

[0143] If the matching data 127 indicates that the candidate function does not match the training set data well enough, then the machine learning module 113 generates one or more other candidate functions and generates test data 126 by applying the one or more candidate functions to the training set data 122 as described above.

[0144] In one embodiment, to form one or more new candidate functions, components of previously formed candidate functions that match the training data better than other candidate functions, but perhaps not enough to be determined acceptable functions, are used to generate new candidate functions which are then tested. In one embodiment, a component of a new candidate function includes one or more operators of the previously formed candidate function. In one embodiment, a component of a new candidate function includes one or more constants of the previously formed candidate function. In one embodiment, a component of a new candidate function includes one or more dependencies used to generate the previously formed candidate function.

[0145] In one embodiment, one or more of the predetermined number of candidate functions that match the training data better than other candidate functions are split into two or more components each, and the split components recombined into new candidate functions that are then tested to determine how well test data generated from those new candidate functions match the training set data. One or more of those new candidate functions that are determined to generate test data that match the training set data better than the original candidate functions may then again be split, if desired, and recombined into a second set of new candidate functions, and so on, until one or more resulting candidate functions produce test data that are deemed to match the training set data within a predetermined margin of error, thus determining that the one or more candidate functions are acceptable, as discussed herein. Thus, machine learning module 113 learns the components of the best functions and uses those components to quickly iterate towards an optimum solution.

[0146] The machine learning module 113 can continue to generate candidate functions in successive iterations until an acceptable candidate function has been found. The machine learning module 113 can continue from one line of the new tax form to the next until all selected lines of the tax form have been correctly learned by the machine learning module 113.

[0147] In one embodiment, when all selected lines of the new tax form have been learned, the machine learning module 113 generates learned form data 121 that indicates that the new tax form has been learned. The learned form data 121 can also include acceptable functions for each selected line of the new tax form. The interface module 112 can output the learned form data 121 for review by an expert or other personnel.

[0148] In one embodiment, when the tax form has been learned by the machine learning module 113, the machine learning module 113 updates the current document instructions data 131 to include software instructions for completing the new tax form as part of the tax return preparation process.

[0149] Embodiments of the present disclosure provide a technical solution to longstanding problems associated with traditional electronic document preparation systems that do not adequately learn and incorporate new and/or updated forms into the electronic document preparation system. An electronic document preparation system in accordance with one or more embodiments provides more reliable financial management services by utilizing machine learning and training set data to learn and incorporate new and/or updated forms into the electronic document preparation system. The various embodiments of the disclosure can be implemented to improve the technical fields of data processing, data collection, resource management, and user experience. Therefore, the various described embodiments of the disclosure and their associated benefits amount to significantly more than an abstract idea. In particular, by utilizing machine learning to learn and incorporate new and/or updated forms in the electronic document preparation system, electronic document preparation system can more efficiently learn and incorporate new and/or updated forms into the electronic document preparation system.

PROCESS

[0150] FIG. 2 illustrates a functional flow diagram of a process 200 for learning and incorporating new and/or updated forms in an electronic document preparation system, in accordance with one embodiment.

[0151] At block 202 the user interface module 112 receives form data related to a new and/or updated form having a plurality of data fields that expect data values in accordance with specific functions, according to one embodiment. From block 202 the process proceeds to block 204.

[0152] At block 204 the data acquisition module 114 gathers training set data related to previously filled forms having completed data fields that each correspond to a respective data field of the new and/or updated form, according to one embodiment. From block 204 the process proceeds to block 206.

[0153] At block 206 the machine learning module 113 generates candidate function data including, for one or more data fields of the new and/or updated form, at least one candidate function, according to one embodiment. From block 206 the process proceeds to block 208.

[0154] At block 208 the machine learning module 113 generates test data by applying the candidate functions to the training set data, according to one embodiment. From block 208 the process proceeds to block 210.

[0155] At block 210 the machine learning module 113 generates matching data indicating how closely each candidate function matches the test data, according to one embodiment.

[0156] In one embodiment, a fitness function is used to determine that one or more candidate functions are acceptable. In one embodiment, the fitness function includes an error function, such as a root mean square error function, reflecting errors that may be present in test data associated with one or more data sets of the training set data, as discussed herein. Other error functions currently known to those of ordinary skill or later developed may be used without departing from the scope of this disclosure. Other components of a fitness function include, according to various embodiments, one or more of how many operators are present in the candidate function, how many operators depend on results of other operators completing prior operations, whether there are missing arguments in the candidate function, and whether an argument is repeated in the candidate function. The tax return preparation system then generates results data indicating whether the candidate function is acceptable and/or a fitness score, determined using a fitness function or an error function, or both, which may be used in a determination of a level of fitness, or a determination of a level of acceptability, for example.

[0157] In one embodiment, to form one or more new candidate functions, components of previously formed candidate functions that match the training data better than other candidate functions, but perhaps not enough to be determined acceptable functions, are used to generate new candidate functions which are then tested. In one embodiment, a component of a new candidate function includes one or more operators of the previously formed candidate function. In one embodiment, a component of a new candidate function includes one or more constants of the previously formed candidate function. In one embodiment, a component of a new candidate function includes one or more dependencies used to generate the previously formed candidate function.

[0158] In one embodiment, one or more of the predetermined number of candidate functions that match the training data better than other candidate functions are split into two or more components each, and the split components recombined into new candidate functions that are then tested to determine how well test data generated from those new candidate functions match the training set data. One or more of those new candidate functions that are determined to generate test data that match the training set data better than the original candidate functions may then again be split, if desired, and recombined into a second set of new candidate functions, and so on, until one or more resulting candidate functions produce test data that are deemed to match the training set data within a predetermined margin of error, thus determining that the one or more candidate functions are acceptable, as discussed herein. Thus, machine learning module 113 learns the components of the best functions and uses those components to quickly iterate towards an optimum solution. As discussed herein, determination of acceptability of a given candidate function or the determination of the fitness of a given candidate function includes, in one embodiment, an error function such as a root mean square, for each data set of the training set data, as discussed below. Other considerations include, according to various embodiments, include one or more of how many operators are present in the candidate function, how many operators depend on results of other operators completing prior operations, whether there are missing arguments in the candidate function, and whether an argument is repeated in the candidate function.

[0159] From block 210 the process proceeds to block 212.

[0160] At block 212, the machine learning module 113 identifies a respective acceptable function for each data field of the new and/or updated form based on the matching data. From block 212 the process proceeds to block 214.

[0161] At block 214 the machine learning module 113 generates results data indicating an acceptable function for each data field of the new and/or updated form, according to one embodiment. From block 214 the process proceeds to block 216. At block 216, the interface module 112 optionally outputs the results data for review by an expert or other personnel, according to one embodiment.

[0162] Although a particular sequence is described herein for the execution of the process 200, other sequences can also be implemented. For example, the data acquisition module can gather training set data each time a new data field of the new and/or updated form is to be learned. The machine learning module can generate a single candidate function at a time and can generate test data and matching data for that candidate function and determine if the candidate function is acceptable based on the matching data. If the candidate function is not acceptable, the machine learning module 113 returns to step 206 and generates a new candidate function, as discussed herein, and repeats the process until an acceptable function has been found for the data field currently being learned. When an acceptable function is found for a particular data field, the data acquisition module can again gather training set data for the next data field and the machine learning module 113 can generate, test, and analyze candidate functions until an acceptable function has been found. The machine learning module can generate candidate functions based on dependency data that indicates one or more possible dependencies for an acceptable function for a given data field. The machine learning module can generate candidate functions by selecting one or more operators from a library of operators. Other sequences can also be implemented.

[0163] In one embodiment, following the determination of two or more candidate functions producing test data matching the training set data, a selection of a 'most' acceptable function may be desirable. In one embodiment, candidate functions producing test data matching the training set data are simplified, and candidate functions that contain the same operators, but which may have those operators in a different order, are combined into a single candidate function, and a desirability value is assigned to the resulting candidate function reflecting that the same candidate function was found more than once. The more times a same candidate function appears in results, the greater the desirability value. Further desirability values may be assigned or adjusted based on one or more other factors, in various embodiments, such as whether one operator or another is preferred for a given data field, whether a set of operators is preferred for a given data field, whether a particular type of operator is preferred for a given data field, and the like. Other factors known to those of ordinary skill may also be used in a desirability value determination, including factors that are later developed.

[0164] FIG. 3 illustrates a flow diagram of a process 300 for learning and incorporating new and/or updated forms in an electronic document preparation system, according to various embodiments.

[0165] In one embodiment, process 300 for learning and incorporating new and/or updated forms in an electronic document preparation system begins at BEGIN 302 and process flow proceeds to RECEIVE FORM DATA RELATED TO A NEW AND/OR UPDATED FORM HAVING ONE OR MORE DATA FIELDS TO BE LEARNED 304.

[0166] In one embodiment, at RECEIVE FORM DATA RELATED TO A NEW AND/OR UPDATED FORM HAVING ONE OR MORE DATA FIELDS TO BE LEARNED 304 process 300 for learning and incorporating new and/or updated forms in an electronic document preparation system receives form data related to a new and/or updated form having one or more data fields to be learned.

[0167] In one embodiment, once process 300 for learning and incorporating new and/or updated forms in an electronic document preparation system receives form data related to a new and/or updated form having a plurality of data fields at RECEIVE FORM DATA RELATED TO A NEW AND/OR UPDATED FORM HAVING ONE OR MORE DATA FIELDS TO BE LEARNED 304 process flow proceeds to GATHER TRAINING SET DATA RELATED TO

PREVIOUSLY FILLED FORMS, EACH PREVIOUSLY FILLED FORM HAVING COMPLETED DATA FIELDS THAT CORRESPOND TO A RESPECTIVE DATA FIELD OF THE NEW AND/OR UPDATED FORM TO BE LEARNED 306.

[0168] In one embodiment, at GATHER TRAINING SET DATA RELATED TO PREVIOUSLY FILLED FORMS, EACH PREVIOUSLY FILLED FORM HAVING COMPLETED DATA FIELDS THAT CORRESPOND TO A RESPECTIVE DATA FIELD OF THE NEW AND/OR UPDATED FORM TO BE LEARNED 306, process 300 for learning and incorporating new and/or updated forms in an electronic document preparation system gathers training set data related to previously filled forms having one or more completed data fields that correspond to a data field of the new and/or updated form.

[0169] In one embodiment, one or more data values of the training set data representing previously filled forms is missing one or more data values, such as if a user previously filling in a first form didn't prepare a predicate form that relates to the current form being learned. In this case, a missing data value might be zero, or might be something different, but it is often not desirable to guess a data value to be substituted for that missing data value. Rather, in one embodiment, a known placeholder value is substituted for the missing data value, such as either a high positive value or high negative value, such as -99999 being substituted for the missing data value, in a data set of the training set data. In such circumstances, process 400 is configured to understand that a particular high positive value in a data set, or a particular high negative value indicates a missing data value in a given data set of the training set data.

[0170] In one embodiment, where an acceptable candidate function for a given data field of a form is expected to be complicated, one or more missing data values within a data set of the training data are replaced by a two-variable pair formed of a boolean value and a float value where the boolean value is set to 'true' if the data associated with the missing data value exists and the associated float value is set to the filled data value, and the boolean value is set to 'false' if the field associated with the missing data value is missing and the associated float value is set to a predetermined known placeholder value, such as -99999 discussed above.

[0171] In one embodiment, once process 300 for learning and incorporating new and/or updated forms in an electronic document preparation system gathers training set data related to previously filled forms at GATHER TRAINING SET DATA RELATED TO PREVIOUSLY FILLED FORMS, EACH PREVIOUSLY FILLED FORM HAVING COMPLETED DATA FIELDS THAT CORRESPOND TO A RESPECTIVE DATA FIELD OF THE NEW AND/OR UPDATED FORM TO BE LEARNED 306, process flow proceeds to GENERATE, FOR A

FIRST SELECTED DATA FIELD OF THE NEW AND/OR UPDATED FORM, DEPENDENCY DATA INDICATING ONE OR MORE POSSIBLE DEPENDENCIES FOR AN ACCEPTABLE FUNCTION 308.

[0172] In one embodiment, at GENERATE, FOR A FIRST SELECTED DATA FIELD OF THE NEW AND/OR UPDATED FORM, DEPENDENCY DATA INDICATING ONE OR MORE POSSIBLE DEPENDENCIES FOR AN ACCEPTABLE FUNCTION 308, process 300 for learning and incorporating new and/or updated forms in an electronic document preparation system generates, for a first selected data field of the plurality of data fields of the new and/or updated form, dependency data indicating one or more possible dependencies for an acceptable function that provides a proper data value for the first selected data field.

[0173] In one embodiment, once process 300 for learning and incorporating new and/or updated forms in an electronic document preparation system generates, for a first selected data field of the plurality of data fields of the new and/or updated form, dependency data indicating one or more possible dependencies for an acceptable function that provides a proper data value for the first selected data field at GENERATE, FOR A FIRST SELECTED DATA FIELD OF THE NEW AND/OR UPDATED FORM, DEPENDENCY DATA INDICATING ONE OR MORE POSSIBLE DEPENDENCIES FOR AN ACCEPTABLE FUNCTION 308, process flow proceeds to GENERATE, FOR THE FIRST SELECTED DATA FIELD, CANDIDATE FUNCTION DATA INCLUDING ONE OR MORE CANDIDATE FUNCTIONS BASED ON THE DEPENDENCY DATA AND ONE OR MORE OPERATORS 310.

[0174] In one embodiment, at GENERATE, FOR THE FIRST SELECTED DATA FIELD, CANDIDATE FUNCTION DATA INCLUDING ONE OR MORE CANDIDATE FUNCTIONS BASED ON THE DEPENDENCY DATA AND ONE OR MORE OPERATORS 310, process 300 for learning and incorporating new and/or updated forms in an electronic document preparation system generates, for the first selected data field, candidate function data including one or more candidate functions based on the dependency data and one or more operators. The candidate functions include, in various embodiments, one or more operators selected from a set of operators which includes logical and mathematical functionality. The operators include, in various embodiments, arithmetic operators such as addition, subtraction, multiplication, division or other mathematical operators, exponential functions, logical operators such as if-then operators, and/or Boolean operators such as true/false. The operators can include existence condition operators that depend on the existence of a data value in another data field of new and/or updated form, in a form other than the new and/or updated form, or in some other location or data set. The operators can include string comparisons and/or rounding or truncating operations, or operators representing any other functional operation that can operate on dependencies and constants to provide a suitable output data value for the data field being learned.

[0175] In one embodiment, once process 300 for learning and incorporating new and/or updated forms in an electronic document preparation system generates, for the first selected data field, candidate function data including one or more candidate functions based on the dependency data and one or more operators selected from a set of operators at GENERATE, FOR THE FIRST SELECTED DATA FIELD, CANDIDATE FUNCTION DATA INCLUDING ONE OR MORE CANDIDATE FUNCTIONS BASED ON THE DEPENDENCY DATA AND ONE OR MORE OPERATORS 310, process flow proceeds to GENERATE, FOR ONE OR MORE CANDIDATE FUNCTIONS, TEST DATA BY APPLYING THE CANDIDATE FUNCTION TO THE TRAINING SET DATA 312.

[0176] In one embodiment, at GENERATE, FOR ONE OR MORE CANDIDATE FUNCTIONS, TEST DATA BY APPLYING THE CANDIDATE FUNCTION TO THE TRAINING SET DATA 312 the process 300 generates, for each candidate function, test data by applying the candidate function to the training set data. The machine learning module 113 of FIG. 1 generates test values of test data 126, in one embodiment, by substituting at least a portion of the training set data for one or more of lines 31 and 2c in the candidate function and determining a result of performing the candidate function.

[0177] In one embodiment, once process 300 generates, for each candidate function, test data by applying the candidate function to the training set data at GENERATE, FOR ONE OR MORE CANDIDATE FUNCTIONS, TEST DATA BY APPLYING THE CANDIDATE FUNCTION TO THE TRAINING SET DATA 312 of FIG. 3, process flow proceeds to GENERATE, FOR ONE OR MORE CANDIDATE FUNCTIONS, MATCHING DATA INDICATING HOW CLOSELY THE TEST DATA MATCHES CORRESPONDING COMPLETED DATA FIELDS OF THE PREVIOUSLY FILLED FORMS 314.

[0178] In one embodiment, at GENERATE, FOR ONE OR MORE CANDIDATE FUNCTIONS, MATCHING DATA INDICATING HOW CLOSELY THE TEST DATA MATCHES CORRESPONDING COMPLETED DATA FIELDS OF THE PREVIOUSLY FILLED FORMS 314 the process 300 for learning and incorporating new and/or updated forms in an electronic document preparation system generates, for one or more candidate functions being learned, matching data. In one embodiment, the matching data is generated by comparing the test data to training set data corresponding to the first selected data field, the matching data indicating how closely the test data matches the corresponding completed data fields of the previously filled forms.

[0179] In one embodiment, a fitness function is used to determine whether one or more candidate functions are acceptable. In one embodiment, the fitness function includes consideration of an error function such as a square root of the sum of the squares of the differences between the desired output of a candidate function and the actual output of the candidate function, for each data set of the training set data, as discussed below. Other considerations included in a fitness function, according to various embodiments, are one or more of how many operators are present in the candidate function, how many operators depend on results of other operators completing prior operations, whether there are missing arguments in the candidate function, and whether an argument is repeated in the candidate function.

[0180] In one embodiment, once the process 300 for learning and incorporating new and/or updated forms in an electronic document preparation system generates, for each candidate function, matching data by comparing the test data to the completed data fields corresponding to the first selected data field, the matching data indicating how closely the test data matches the corresponding completed data fields of the previously filled forms at GENERATE, FOR ONE OR MORE CANDIDATE FUNCTIONS, MATCHING DATA INDICATING HOW CLOSELY THE TEST DATA MATCHES CORRESPONDING COMPLETED DATA FIELDS OF THE PREVIOUSLY FILLED FORMS 314, process flow proceeds to IDENTIFY, FROM THE CANDIDATE FUNCTIONS, AN ACCEPTABLE CANDIDATE FUNCTION FOR THE FIRST DATA FIELD OF THE NEW AND/OR UPDATED FORM BY DETERMINING, FOR EACH CANDIDATE FUNCTION, WHETHER OR NOT THE CANDIDATE FUNCTION IS AN ACCEPTABLE FUNCTION FOR THE FIRST SELECTED DATA FIELD OF THE NEW AND/OR UPDATED FORM BASED ON THE MATCHING DATA 316.

[0181] In one embodiment, at IDENTIFY, FROM THE CANDIDATE FUNCTIONS, AN ACCEPTABLE CANDIDATE FUNCTION FOR THE FIRST DATA FIELD OF THE NEW AND/OR UPDATED FORM BY DETERMINING, FOR EACH CANDIDATE FUNCTION, WHETHER OR NOT THE CANDIDATE FUNCTION IS AN ACCEPTABLE FUNCTION FOR THE FIRST SELECTED DATA FIELD OF THE NEW AND/OR UPDATED FORM BASED ON THE MATCHING DATA 316 the process 300 for learning and incorporating new and/or updated forms in an electronic document preparation system identifies, from the plurality of functions, an acceptable candidate function for the first data field of the new and/or updated form by determining, for the various candidate functions, whether or not the candidate function is an acceptable function for the first selected data field of the new and/or updated form based on the matching data.

[0182] If, at IDENTIFY, FROM THE CANDIDATE FUNCTIONS, AN ACCEPTABLE CANDIDATE FUNCTION FOR THE FIRST DATA FIELD OF THE NEW AND/OR UPDATED FORM BY DETERMINING, FOR EACH CANDIDATE FUNCTION, WHETHER OR NOT THE CANDIDATE FUNCTION IS AN ACCEPTABLE FUNCTION FOR THE FIRST SELECTED DATA FIELD OF THE NEW AND/OR UPDATED FORM BASED ON THE MATCHING DATA 316, the matching data may indicate that there are no acceptable candidate functions among the candidate functions being considered. If so, new candidate functions are generated and considered.

[0183] In one embodiment, to form one or more new candidate functions, components of previously formed candidate functions, such as previously formed candidate functions that match the training data better than other candidate functions but perhaps not enough to be determined acceptable functions, are used to generate new candidate functions which are then tested. In one embodiment, a component of a new candidate function includes one or more operators of a previously formed candidate function. In one embodiment, a component of a new candidate function includes one or more constants of the previously formed candidate function. In one embodiment, a component of a new candidate function includes one or more dependencies used to generate the previously formed candidate function.

[0184] In one embodiment, one or more of the predetermined number of candidate functions that match the training data better than other candidate functions are split into two or more components each, and the split components recombined into new candidate functions that are then tested to determine how well test data generated from those new candidate functions match the training set data. One or more of those new candidate functions that are determined to generate test data that match the training set data better than the original candidate functions may then again be split, if desired, and recombined into a second set of new candidate functions, and so on, until one or more resulting candidate functions produce test data that are deemed to match the training set data within a predetermined margin of error, as discussed herein. Thus, machine learning module 113 of FIG. 1 learns the components of the best functions and uses those components to quickly iterate towards an optimum solution.

[0185] In one embodiment, once the process 300 for learning and incorporating new and/or updated forms in an electronic document preparation system identifies, from the plurality of functions, an acceptable candidate function for the first data field of the new and/or updated form by determining, for each candidate function, whether or not the candidate function is an acceptable function for the first selected data field of the new and/or updated form based on the matching data at IDENTIFY, FROM THE CANDIDATE FUNCTIONS, AN ACCEPTABLE CANDIDATE FUNCTION FOR THE FIRST DATA FIELD OF THE NEW AND/OR UPDATED FORM BY DETERMINING, FOR EACH CANDIDATE FUNCTION, WHETHER OR NOT THE CANDIDATE FUNCTION IS AN ACCEPTABLE FUNCTION FOR THE FIRST SELECTED DATA FIELD OF THE NEW AND/OR UPDATED FORM BASED ON THE MATCHING DATA 316, process flow proceeds to GENERATE, AFTER IDENTIFYING AN ACCEPTABLE FUNCTION FOR THE FIRST DATA FIELD, RESULTS DATA INDICATING THE ACCEPTABLE FUNCTION FOR THE FIRST SELECTED DATA FIELD OF THE NEW AND/OR UPDATED FORM 318.

[0186] In one embodiment, at GENERATE, AFTER IDENTIFYING AN ACCEPTABLE FUNCTION FOR THE FIRST DATA FIELD, RESULTS DATA INDICATING THE ACCEPTABLE FUNCTION FOR THE FIRST SELECTED DATA FIELD OF THE NEW AND/OR UPDATED FORM 318, the process 300 for learning and incorporating new and/or updated forms in an electronic document preparation system generates, after identifying an acceptable function for the first data field, results data indicating the acceptable function for the first selected data field of the new and/or updated form. If more than one acceptable function has been found, the results data may optionally include more than one of the identified acceptable functions.

[0187] In one embodiment, once the process 300 for learning and incorporating new and/or updated forms in an electronic document preparation system generates, after identifying an acceptable function for the first selected data field, results data indicating the acceptable function for the first data field of the new and/or updated form at GENERATE, AFTER IDENTIFYING AN ACCEPTABLE FUNCTION FOR THE FIRST DATA FIELD, RESULTS DATA INDICATING THE ACCEPTABLE FUNCTION FOR THE FIRST SELECTED DATA FIELD OF THE NEW AND/OR UPDATED FORM 318 proceeds to OUTPUT THE RESULTS DATA 320.

[0188] In one embodiment, at OUTPUT THE RESULTS DATA 320 the process 300 for learning and incorporating new and/or updated forms in an electronic document preparation system outputs the results data.

[0189] In one embodiment, once the process 300 for learning and incorporating new and/or updated forms in an electronic document preparation system outputs the results data at OUTPUT THE RESULTS DATA 320, process flow proceeds to END 322 where the process awaits further input.

[0190] In one embodiment, at END 322 the process for learning and incorporating new and/or updated forms in an electronic document preparation system is exited to await new data and/or instructions.

[0191] In one embodiment, following the determination of two or more candidate functions producing test data matching the training set data, a selection of a 'most' acceptable function may be desirable. In one embodiment, candidate functions producing test data matching the training set data are simplified, and candidate functions that contain the same operators, but which may have those operators in a different order, are combined into a single candidate function, and a desirability value is assigned to the resulting candidate function reflecting that the same candidate function was found more than once. The more times a same candidate function appears in results, the greater the desirability value. Further desirability values may be assigned or adjusted based on one or more other factors, in various embodiments, such as whether one operator or another is preferred for a given data field, whether a set of operators is preferred for a given data field, whether a particular type of operator is preferred for a given data field, and the like. Other factors known to those of ordinary skill may also be used in a desirability value determination, including factors that are later developed.

[0192] In one embodiment, there is a need to identify specific candidate functions that perform better, i.e. have a lower error or otherwise have test results that differ from the training set data less than other candidate functions, and use one or more components of those specific candidate functions to form new candidate functions, in order to arrive at an acceptable solution very quickly.

[0193] FIG. 4 is a flow diagram of a process 400 for learning and incorporating new and/or updated forms in an electronic document preparation system, in accordance with one embodiment.

[0194] In one embodiment, process 400 for learning and incorporating new and/or updated forms in an electronic document preparation system begins at BEGIN 402 and process flow proceeds to RECEIVE TRAINING SET DATA RELATING TO A FORM FIELD TO BE LEARNED 404.

[0195] In one embodiment, at RECEIVE TRAINING SET DATA RELATING TO A FORM FIELD TO BE LEARNED 404, training set data is received as discussed above with respect to GATHER TRAINING SET DATA RELATED TO PREVIOUSLY FILLED FORMS, EACH PREVIOUSLY FILLED FORM HAVING COMPLETED DATA FIELDS THAT CORRESPOND TO A RESPECTIVE DATA FIELD OF THE NEW AND/OR UPDATED FORM TO BE LEARNED 306 of FIG. 3. Here, we are focusing our example on a single data field of a form to be learned, and thus only need training set data of the single data field to be learned, including training set data for any other data fields that are used in the determination of a data value for the single data field being learned. For example, if a data field for line 5 of a given form is being learned, and line 5 depends from line 2b of the same form and line 12 of a different form, the training set data will include many different sets of data, where those sets of data ideally include at least lines 2b and 12, and also data from line 5, the field being learned.

[0196] The received training set data will typically include hundreds, thousands, or possibly even millions of sets of data from previously filed tax returns, or from other data sources, depending on the character of the data field being learned. In some instances, a large number of data sets of the received training set data is duplicative, i.e. uses identical data values in lines 2b and 12, for example, thus resulting in the same training set value for line 5 as well. In one embodiment, the received training set data is processed to eliminate duplicate data sets, retaining only one copy for use in learning a function for line 5. Further, in situations where there is a bound placed on data values allowed of a given data field, and where the training set data includes data values outside of that bound, it may be beneficial to eliminate data sets from the training set data those data sets that have data values exceeding that bound. In one embodiment, where line 2b of the example above is only allowed to be a positive number, any data sets of the training set data that have a negative number for line 2b is eliminated from the received training set data. Other observations may also be made, automatically by a computing system, such as determining that one or more of the data values of one or more data sets are zero, such as if one or more of line 2b or line 12 is zero in those data sets. If the number of data sets having a data value of zero is large, it may be advantageous in some situations to eliminate all but a few such data sets, thus reducing the data sets of the training set data. By reducing the number of data sets being used to learn functions, significant time savings is achieved, in addition to significantly reducing memory requirements and processor cycles needed to accomplish the processes described herein.

[0197] Further details on forming training data sets may be found in the U.S. Patent application filed October 13, 2016 having attorney docket number INTU179969, serial number 15/292,510, and entitled SYSTEM AND METHOD FOR SELECTING DATA SAMPLE GROUPS FOR MACHINE LEARNING OF CONTEXT OF DATA FIELDS FOR VARIOUS DOCUMENT TYPES AND/OR FOR TEST DATA GENERATION FOR QUALITY ASSURANCE SYSTEMS naming inventor Cem Unsal.

[0198] In one embodiment, following the receipt of training set data at RECEIVE TRAINING SET DATA RELATING TO A FORM FIELD TO BE LEARNED 404 of FIG. 4, process flow proceeds to DETERMINE PARAMETERS FOR LEARNING CANDIDATE FUNCTIONS FOR THE FORM FIELD 406.

[0199] In one embodiment, at DETERMINE PARAMETERS FOR LEARNING CANDIDATE FUNCTIONS FOR THE FORM FIELD 406, one or more parameters to be incorporated into the learning process are determined. In some embodiments, limits are placed on the number of functions to be generated and tested in a single cycle of the process. For example, it may be desirable to generate and test no more than 200 functions at a time, and then rank those functions according to how closely test data from those functions match the training set data for the particular line of a form associated with the function. In one or more embodiments, if a given form is likely to have less complex functions that can be used to determine one or more data values associated with various data fields of the form, it may be desirable to limit the number of operators to be used in a given candidate function. In a third example, it may be desirable in some circumstances to limit the number of times particular operators are used in a given candidate function. Thus, according to these examples, parameters that may be used in a given instance of the process may include one or more of a maximum number of functions to be generated and tested in a given cycle of the process, a maximum number of operators to be used in candidate functions generated and tested in a given cycle of the process, a maximum total number of candidate functions to be generated and tested prior to the process pausing and presenting results data to a user or other expert, a maximum number of rounds of generating and testing candidate functions, and a maximum number of times particular operators are used in a given candidate function, or any combination thereof. Other parameters may be developed and used in the processes described herein without departing from the teachings of the present disclosure. In this disclosure, the parameters further include, but are not limited to the dependencies discussed herein.

[0200] In one embodiment, following the determination of one or more parameters to be incorporated into the function learning process at DETERMINE PARAMETERS FOR LEARNING CANDIDATE FUNCTIONS FOR THE FORM FIELD 406, process flow proceeds at GENERATE CANDIDATE FUNCTIONS FOR THE FORM FIELD ACCORDING TO THE DETERMINED PARAMETERS 408.

[0201] In one embodiment, at GENERATE CANDIDATE FUNCTIONS FOR THE FORM FIELD ACCORDING TO THE DETERMINED PARAMETERS 408, one or more candidate functions are generated according to the parameters determined at DETERMINE PARAMETERS FOR LEARNING CANDIDATE FUNCTIONS FOR THE FORM FIELD 406. If, for example, a parameter indicates a maximum number of candidate functions to be tested in a given cycle of the process is one hundred, only one hundred or fewer candidate functions are generated at a time. Further, if there is also a parameter indicating that the maximum number of operators in a given candidate function is twenty, then each generated candidate function will contain twenty or fewer operators. If, as a third example, a parameter indicates a maximum number of times a given operator may appear in a given candidate function is four, then each generated candidate function will not generate any candidate functions having any particular operator appearing more than four times. As discussed above, the parameters may also include dependencies, such as other lines that a data field of the current line needs to be determined correctly. Therefore, in one embodiment, candidate functions generated at GENERATE CANDIDATE FUNCTIONS FOR THE FORM FIELD ACCORDING TO THE DETERMINED PARAMETERS 408 will include consideration of those dependencies. For example, a data field depending on line 2 and having a constant of 3000 will consider, and perhaps include, one or more of those dependencies when generating the candidate functions. It is not necessarily true that each dependency will be overtly present in each candidate function. It has been seen, for example, that a seemingly complex line in a tax return that has complicated accompanying instructions depending on many factors may actually be able to be determined with a single operator function copying a data value from a worksheet or other data field. This is largely due to many different scenarios the current line is designed to cover rarely or never actually take place.

[0202] In one embodiment, once candidate functions are generated at GENERATE CANDIDATE FUNCTIONS FOR THE FORM FIELD ACCORDING TO THE

DETERMINED PARAMETERS 408, process flow proceeds at GENERATE MATCHING DATA FOR CANDIDATE FUNCTIONS 410. In one embodiment, this process operation includes one or more operations previously discussed with respect to FIG. 3, including one or more of GENERATE, FOR ONE OR MORE CANDIDATE FUNCTIONS, TEST DATA BY APPLYING THE CANDIDATE FUNCTION TO THE TRAINING SET DATA 312 of FIG. 3 and GENERATE, FOR ONE OR MORE CANDIDATE FUNCTIONS, MATCHING DATA INDICATING HOW CLOSELY THE TEST DATA MATCHES CORRESPONDING COMPLETED DATA FIELDS OF THE PREVIOUSLY FILLED FORMS 314. In one embodiment, once test data is generated by, for example, substituting a portion of training set data associated with one or more dependencies, that test data is compared against an actual, known correct data value of the training set data associated with the current line associated with the function being learned. An error function may be used to provide an indication of how closely the actual, known correct data value of the training set data matches the test data generated by the candidate function. Continuing the example above where line 2b of the same form as the data field and function being learned and line 12 of a different form are dependencies associated with line 5 of a current form, where a function for line 5 is being learned, each data set of the training set data used to learn an acceptable function includes at least three data values, the values for line 2b and line 5 of the current form and line 12 of a different form. Furthering the example, assume that there are twenty-four such data sets within the training set data. When test data is generated, each of the respective data values for line 2b and line 5 are substituted, if needed, into a given candidate function being considered, resulting in a line 5 result in the test data. Thus, if all twenty-four data sets are used, then there will be twenty-four data values representing the line 5 test data results for the various data sets. Each of those twenty-four data values representing the line 5 within the test data are compared with the respective line 5 data values within the training set data. Some of the twenty-four line 5 data values may match their line 5 counterpart data values within the training set data exactly, while others may match closely, but not exactly, while yet others may not even be close matches.

[0203] In one embodiment, at GENERATE MATCHING DATA FOR CANDIDATE FUNCTIONS 410 of FIG. 4, the matching data is in the form of a confidence score which includes consideration of how many data values of the test data match their line counterpart data values within the training set data, with points being assigned to a given candidate function based on a percentage of those values that match. In one embodiment, higher numbers of points are assigned for higher percentages of the values matching, reflecting a preference for higher percentages of matches, where candidate functions having higher numbers of points are preferred over candidate functions having lower numbers of points.

[0204] In one embodiment, a given candidate function is further assigned an additional points value depending on whether the candidate function uses one or more operators more than once. In one embodiment, higher numbers of points are assigned for functions using operators fewer numbers of times with candidate functions having higher numbers of points being preferred over candidate functions having lower numbers of points.

[0205] In one embodiment, a given candidate function is further assigned an additional points value depending on whether the candidate function is shorter than other candidate functions. In one embodiment, higher numbers of points are assigned for shorter functions with candidate functions having higher numbers of points being preferred over candidate functions having lower numbers of points. In one embodiment, a shorter candidate function is a candidate function having a fewer total number of operators present in the candidate function. In one embodiment, a shorter candidate function is a candidate function having a fewer total number of operators and constants present in the candidate function. In one embodiment, a shorter candidate function is a candidate function having a fewer total number of operators and dependencies present in the candidate function.

[0206] In one embodiment, a fitness function is used to determine whether one or more candidate functions are acceptable. In one embodiment, the fitness function includes consideration of an error function such as a square root of the sum of the squares of the differences between the desired output of a candidate function and the actual output of the candidate function, for each data set of the training set data, as discussed below. Other considerations included in a fitness function, according to various embodiments, are one or more of how many operators are present in the candidate function, how many operators depend on results of other operators completing prior operations, whether there are missing arguments in the candidate function, and whether an argument is repeated in the candidate function.

[0207] Many other types of matching data reflecting the degree of preference of one or more candidate functions over other candidate functions may be developed and used similarly, without departing from the scope and teachings of this disclosure.

[0208] It may be desirable, in some situations, to discontinue producing new candidate functions, such as if an error function or a fitness function discussed herein reflects that the fitness, or acceptability, of the entire population is within a predetermined margin, such as if fitness values for each candidate function determined using a fitness function discussed herein are all within 10% of each other, or if a standard deviation of the fitness values is below a certain predetermined value, or using other criteria. Thus, a process operation to test exit conditions is performed at any point during the operation of process 400, using any exit criteria desired by an implementer of process 400. If an exit condition is found to be satisfied, the process exits, . In one embodiment, as the process exits, results data is produced reflecting one or more candidate functions. In one embodiment, the one or more candidate functions of the results data includes at least one candidate function which is a better or more acceptable candidate function than at least one other candidate function. In one embodiment, acceptability or a determination of whether one candidate function is better than another candidate function is based on comparing the results of applying a fitness function to test data associated with the candidate functions.

[0209] Exit criteria may include a wide variety of conditions. Such conditions include, in various embodiment, a minimum value of an error function associated with the population of candidate functions remaining unchanged within a most recent predetermined number of iterations of process 400, and/or a predefined number of iterations of process 400 have already occurred,

[0210] In one embodiment, once matching data has been generated at GENERATE MATCHING DATA FOR CANDIDATE FUNCTIONS 410, process flow proceeds at SELECT ONE OR MORE CANDIDATE FUNCTIONS NOT MEETING ACCEPTABILITY CRITERIA 412.

[0211] In one embodiment, at SELECT ONE OR MORE CANDIDATE FUNCTIONS NOT MEETING ACCEPTABILITY CRITERIA 412 there is acceptability criteria that must be met in order for a given candidate function to be determined to be an acceptable candidate function so that learning may be considered to be complete. In one embodiment, using the example provided above where the matching data include points being assigned to a candidate function based on one or more factors such as the length of the function, how many data sets are matched by the test data, etc., the acceptability criteria includes a threshold number of points a given candidate function must have in order to be considered acceptable.

[0212] In one embodiment, after having been evaluated at GENERATE MATCHING DATA FOR CANDIDATE FUNCTIONS 410, each candidate function has a number of points assigned. In a system, like the examples above, where having a greater number of points is better than having fewer points, a given candidate function is not acceptable if it has fewer than a threshold number of points assigned to it.

[0213] In one embodiment, at SELECT ONE OR MORE CANDIDATE FUNCTIONS NOT MEETING ACCEPTABILITY CRITERIA 412 any candidate functions not meeting acceptability criteria, such as not having enough points assigned to exceed a threshold number of points, are determined. In one embodiment, only a predetermined number of candidate functions are selected from all of the candidate functions generated at GENERATE CANDIDATE FUNCTIONS FOR THE FORM FIELD ACCORDING TO THE DETERMINED PARAMETERS 408. In one embodiment, the predetermined number of candidate functions selected at SELECT ONE OR MORE CANDIDATE FUNCTIONS NOT MEETING ACCEPTABILITY CRITERIA 412 are the best candidate functions, as determined by those candidate functions having the highest number of points, or those candidate functions having the lowest error, or using any other criteria known to those of ordinary skill or developed later. In one example, assume two hundred candidate functions were generated at GENERATE CANDIDATE FUNCTIONS FOR THE FORM FIELD ACCORDING TO THE DETERMINED PARAMETERS 408. Further assume that none of the candidate functions meet acceptability criteria, such as a point threshold discussed above. In one embodiments, at SELECT ONE OR MORE CANDIDATE FUNCTIONS NOT MEETING ACCEPTABILITY CRITERIA 412, a subset of the 200 generated candidate functions are selected for further processing. In one embodiment, the subset includes the best twenty candidate functions selected, based on the matching data of GENERATE MATCHING DATA FOR CANDIDATE FUNCTIONS 410.

[0214] In one embodiment, tested candidate functions may be grouped into random groups of a predetermined size, and the best one or more candidate functions in each group may also/instead be selected at SELECT ONE OR MORE CANDIDATE FUNCTIONS NOT MEETING ACCEPTABILITY CRITERIA 412.

[0215] Many other options for selecting candidate functions to be at least partly used in process operations below are possible, with the variation remaining under the scope of this disclosure.

[0216] Once one or more candidate functions not meeting acceptability criteria are selected at SELECT ONE OR MORE CANDIDATE FUNCTIONS NOT MEETING ACCEPTABILITY CRITERIA 412, process flow proceeds at SPLIT EACH OF THE ONE OR MORE SELECTED CANDIDATE FUNCTIONS INTO COMPONENTS; RECOMBINE THE COMPONENTS INTO NEW CANDIDATE FUNCTIONS 414.

[0217] In one embodiment, at SPLIT EACH OF THE ONE OR MORE SELECTED CANDIDATE FUNCTIONS INTO COMPONENTS; RECOMBINE THE COMPONENTS INTO NEW CANDIDATE FUNCTIONS 414, one or more of the candidate functions selected at SELECT ONE OR MORE CANDIDATE FUNCTIONS NOT MEETING ACCEPTABILITY CRITERIA 412 are split into two or more components. One or more of those components are then recombined with other candidate functions, or other components, resulting in new candidate functions.

[0218] In one embodiment, one or more candidate functions are split at or near a halfway point, leaving equal or relatively equal numbers of operators in each of the resulting components. In one embodiment, in the case of a candidate function having an odd number of operators, the candidate function is split, resulting in two components, where one of the components has one operators more than the component. In one embodiment, one or more candidate functions are split into three or more components. Further, it is not necessary that each candidate function be split into the same number of components. Finally, one or more components from a first split candidate function may be recombined with components from one, two, three or more other split candidate functions.

[0219] If it is desirable in a given implementation to generate additional candidate functions from the original candidate functions, one or more of the original candidate functions are used, in one embodiment, to generate one or more new candidate functions through process 400 randomly replacing one or more portions of the original candidate function. In one embodiment, randomly replacing one or more portions of the original candidate function includes replacing one or more operators and/or constants in the original candidate function with one or more different operators. In one embodiment, the one or more different operators are randomly selected. In one embodiment, the one or more different operators are selected from a group of operators not already present in the original candidate function.

[0220] In one embodiment, one or more of the original candidate functions are grouped with or otherwise used in a future fitness evaluation/test cycle with the new candidate functions. Thus, those original candidate functions that are used in a later evaluation/test cycle will also be referred to as new candidate functions just to ensure that one or more operations described herein as being performed on new candidate functions may also be performed on those original candidate functions.

[0221] In one embodiment, once new candidate functions are generated at SPLIT EACH OF THE ONE OR MORE SELECTED CANDIDATE FUNCTIONS INTO COMPONENTS;

RECOMBINE THE COMPONENTS INTO NEW CANDIDATE FUNCTIONS 414, process flow proceeds at IDENTIFY ONE OR MORE CANDIDATE FUNCTIONS THAT MEET ACCEPTABILITY CRITERIA, OR ALTERNATIVELY SPLIT AND RECOMBINE CANDIDATE FUNCTIONS UNTIL ACCEPTABILITY CRITERIA IS SATISFIED 416.

[0222] In one embodiment, the process flow continues by testing the new candidate functions and identifying, using matching data or otherwise any candidate functions meeting acceptability criteria, any of the new candidate functions that are acceptable. If no candidate functions found to be acceptable, process flow repeats the splitting, recombining, and testing operations until one or more acceptable candidate functions are found. Following one or more acceptable candidate functions being found, process flow proceeds at GENERATE RESULTS DATA INDICATING ONE OR MORE ACCEPTABLE CANDIDATE FUNCTIONS 418.

[0223] In one embodiment, at GENERATE RESULTS DATA INDICATING ONE OR MORE ACCEPTABLE CANDIDATE FUNCTIONS 418, results data is generated indicating one or more acceptable functions. If more than one acceptable function has been found, the results data may optionally include more than one of the acceptable functions.

[0224] In one embodiment, process flow then proceeds to OUTPUT THE RESULTS DATA 420.

[0225] In one embodiment, at OUTPUT THE RESULTS DATA 420 the results data are provided to one or more users of the process as discussed herein after which process flow proceeds to END 422 where the process awaits further input.

[0226] In one embodiment, at END 422 the process for learning and incorporating new and/or updated forms in an electronic document preparation system is exited to await new data and/or instructions.

[0227] In the discussion above, reference was made to the natural language parsing module 115 analyzing the form data 119 with a natural language parsing process. The disclosure below teaches one embodiment of the natural language parsing process.

[0228] FIG. 5 is a flow diagram of a process for learning and incorporating new and/or updated forms in an electronic document preparation system, in accordance with one embodiment.

[0229] In one embodiment, process 500 for learning and incorporating new and/or updated forms in an electronic document preparation system begins at BEGIN OPERATION 502 and proceeds with ACQUIRE EXTERNAL AND LOCAL TEXTUAL DATA RELATING TO A FORM HAVING FORM FIELDS TO BE LEARNED; INCORPORATE AND

CONVERT ELECTRONIC AND PHYSICAL TEXTUAL DATA INTO AN ELECTRONIC CORPUS OPERATION 504.

[0230] In one embodiment, interface module 112 is configured to receive form data 119 related to a new and/or updated form. Interface module 112 can receive the form data 119 from an expert, from a government agency, from a financial institution, or in other ways now known or later developed. In various embodiments, form data 119 originates as one or more physical printed pages or electronic equivalents of actual form data relating to the physical form, such as an instruction booklet or other documentation, to electronic textual data. For example, the form data 119 may include text descriptions and/or form text for various data fields of the new and/or updated form. The text descriptions and form text originate from one or more different sources, such as, in the case of the new and/or updated for being a U.S. text form, from the IRS. The text descriptions and form text include, in one embodiment, text of one or more actual tax forms issued by the IRS and required to be filled out by taxpayers for which the new and/or updated form applies. The text descriptions and form text further include, in various embodiments, text of one or more instruction sets and publications issued by the IRS to assist the tax payer or tax preparer properly complete the form. The natural language parsing module 115 analyzes these text descriptions through process described herein and generates natural language parsing data 118 indicating the type of data value expected in each data field.

[0231] In one embodiment, form data 119 relates to specific subsections of a given new or updated form, such as form text and/or form data of or relating to one or more form fields of the new or updated form, such as changed sections of the form from a prior version. In one embodiment, at ACQUIRE EXTERNAL AND LOCAL TEXTUAL DATA RELATING TO A FORM HAVING FORM FIELDS TO BE LEARNED; INCORPORATE AND CONVERT ELECTRONIC AND PHYSICAL TEXTUAL DATA INTO AN ELECTRONIC CORPUS OPERATION 504, form data 119 originates as one or more portions or components of physical forms such as paper forms which are scanned or otherwise converted through optical character recognition or other known or later developed methods from physical form to electronic textual data of form data 119. In one embodiment, the electronic textual data relating to the new or updated form is collected into an electronic text corpus including all of the acquired and converted text data and stored as at least a portion of form data 119.

[0232] In one embodiment, following completion of ACQUIRE EXTERNAL AND LOCAL TEXTUAL DATA RELATING TO A FORM HAVING FORM FIELDS TO BE LEARNED; INCORPORATE AND CONVERT ELECTRONIC AND PHYSICAL TEXTUAL

DATA INTO AN ELECTRONIC CORPUS OPERATION 504, process flow proceeds with SELECT A FORM FIELD TO BE LEARNED AND PREPROCESS CORPUS TO EXTRACT ELECTRONIC TEXTUAL DATA RELATING TO THE SELECTED FORM FIELD OPERATION 506.

[0233] In one embodiment, at SELECT A FORM FIELD TO BE LEARNED AND PREPROCESS CORPUS TO EXTRACT ELECTRONIC TEXTUAL DATA RELATING TO THE SELECTED FORM FIELD OPERATION 506, a form field to be learned is selected, and the electronic text corpus of form data 199 is analyzed to identify and extract electronic corpus data of or relating to the selected form field.

[0234] As an example, IRS form 2441, a form for determining and/or reporting Child and Dependent Care Expenses includes a line 3 of that form which recites "Add the amounts in column (c) of line 2. Do not enter more than $3,000 for one qualifying person or $6,000 for two or more persons. If you completed Part III, enter the amount from line 31" and has a form field associated with the text. In this example, the selected form field is a data storage location for a data value determined in accordance with the requirements of the text as understood in the context of any other instructions of documentation associated with the form and/or line number associated with the selected form field. As discussed herein, dependencies for this line on form 2441 include but are not limited to one or more of "amounts in column (c) of line 2" and line 31 of part 3, if completed.

[0235] In this example, at SELECT A FORM FIELD TO BE LEARNED AND PREPROCESS CORPUS TO EXTRACT ELECTRONIC TEXTUAL DATA RELATING TO THE SELECTED FORM FIELD OPERATION 506, the electronic text corpus is analyzed to identify and extract electronic corpus data of or relating to IRS form 2441 and/or line 3 of IRS form 2441. As discussed above, the extracted electronic corpus data will include, in various embodiments, one or more of electronic data of or relating to the actual text of line 3 of IRS form 2441, documentation, explanations and/or instructions relating to the determination of data values of or relating to IRS form 2441 and any other electronic data determined to be useful by a designer of a particular implementation of the processes discussed herein.

[0236] In one embodiment, at SELECT A FORM FIELD TO BE LEARNED AND PREPROCESS CORPUS TO EXTRACT ELECTRONIC TEXTUAL DATA RELATING TO THE SELECTED FORM FIELD OPERATION 506, the various extracted electronic corpus data is mapped or otherwise tagged with one or more identifiers that indicate a particular line item, form field, or form to which the extracted electronic corpus data relates. The mapping may take place with tags, a second database or other tracking system, or in any other way known to persons of skill in the art or later developed.

[0237] In one embodiment, a given tag is associated with an entire set of textual data of the extracted electronic corpus data. In one embodiment, a given tag is associated with paragraph of textual data of the extracted electronic corpus data. In one embodiment, a given tag is associated with a sentence of textual data of the extracted electronic corpus data. In one embodiment, a given tag is associated with a multitoken sentence fragment of textual data of the extracted electronic corpus data. In one embodiment, a given tag is associated with a single token sentence fragment of textual data of the extracted electronic corpus data. Various types of tags may be associated with various parts of speech, various lines of a form, or any other association desirable to an implementer of a given embodiment. Further, tags may become associated with portions of the extracted electronic corpus data at any time, and thus need not be assigned at this process operation.

[0238] In one embodiment, following completion of SELECT A FORM FIELD TO BE LEARNED AND PREPROCESS CORPUS TO EXTRACT ELECTRONIC TEXTUAL DATA RELATING TO THE SELECTED FORM FIELD OPERATION 506, process flow proceeds with SEPARATE THE EXTRACTED TEXTUAL DATA INTO WORD GROUPS OF N-GRAMS, OMITTING WORD GROUPS HAVING WORDS FOUND ON AN EXCLUSION LIST OPERATION 508.

[0239] In one embodiment, at SEPARATE THE EXTRACTED TEXTUAL DATA INTO WORD GROUPS OF N-GRAMS, OMITTING WORD GROUPS HAVING WORDS FOUND ON AN EXCLUSION LIST OPERATION 508, the textual data of SELECT A FORM FIELD TO BE LEARNED AND PREPROCESS CORPUS TO EXTRACT ELECTRONIC TEXTUAL DATA RELATING TO THE SELECTED FORM FIELD OPERATION 506 is analyzed and the text data converted to a group of N-grams, where N-grams are commonly known as sequences of words from a given sequence of text. In some circumstances, 1-grams are special single-word cases of N-gram analysis which we will discuss below. In various embodiments, N-grams include only multi-word groups, i.e. no one-word groups, where the number of words is less than, or less than or equal to, a predetermined maximum word group length. In one embodiment, separated extracted textual data only includes N-grams up to a predetermined maximum word group length. In one embodiment, only N-grams equal to or smaller than a word length of five are kept. In one embodiment, only N-grams equal to or smaller than a predetermined maximum word group length of four are kept. Other predetermined maximum word group length are also applicable, such as predetermined word lengths between two and ten, for example. N-grams formed using the first three words of the example text "Do not enter more than $3,000 for one qualifying person or $6,000 for two or more persons" include, for example, "do not," "not enter," and "do not enter."

[0240] Following the separation of the extracted textual data into N-grams that are of an acceptable word length, based on the predetermined maximum word group length, N-grams are eliminated that include any single or multiple word groups that are found on an exclusion list. In one embodiment, N-grams on the exclusion list include one or more single words or N-grams considered to be less important in the subject matter field of the form and related documentation.

[0241] In one embodiment, following the completion of SEPARATE THE EXTRACTED TEXTUAL DATA INTO WORD GROUPS OF N-GRAMS, OMITTING WORD GROUPS HAVING WORDS FOUND ON AN EXCLUSION LIST OPERATION 508, process flow proceeds with DETERMINE A RANKING MEASURE FOR THE WORD GROUPS AND ELIMINATE WORD GROUPS NOT MEETING A RANKING MEASURE CRITERIA, RESULTING IN A FIRST EXTRACTED GROUP OPERATION 510.

[0242] In one embodiment, at DETERMINE A RANKING MEASURE FOR THE WORD GROUPS AND ELIMINATE WORD GROUPS OUTSIDE A RANKING MEASURE CRITERIA, RESULTING IN A FIRST EXTRACTED GROUP OPERATION 510, a ranking measure is determined for each N-gram of SEPARATE THE EXTRACTED TEXTUAL DATA INTO WORD GROUPS OF N-GRAMS, OMITTING WORD GROUPS HAVING WORDS FOUND ON AN EXCLUSION LIST OPERATION 508. In one embodiment, the ranking measure includes a poisson-stirling analysis of the word groups and indicates a degree of importance of a given N-gram. Thus, after ranking all N-grams, a ranking list may be formed from more important to least important and a predetermined ranking criteria may be applied, thus eliminating less important N-grams and leaving only more important N-grams. N-grams not meeting predetermined importance criteria are eliminated, resulting in a first extracted group. In one embodiment, the ranking measure takes into account how many words or word groups a given word of a word group is associated with in the corpus, compared to how many words or word groups other words of a word group is associated with. In an example, the ranking measure will rate "earned income" higher than "the earned", even though the two word groups are both bi-grams. One reason for this is because "the" is typically associated with many other words, and earned if most often associated with the word "earned." In one embodiment, the word groups found in the first extracted group are the highest ranked word groups according to the ranking measure. In one embodiment, only a limited predetermined number of the highest ranked word groups are kept in the first extracted group, eliminating the remaining, lowest ranked word groups.

[0243] In one embodiment, following completion of DETERMINE A RANKING MEASURE FOR THE WORD GROUPS AND ELIMINATE WORD GROUPS OUTSIDE A RANKING MEASURE CRITERIA, RESULTING IN A FIRST EXTRACTED GROUP OPERATION 510, process flow proceeds with SELECT ALL NOUNS IN THE EXTRACTED TEXTUAL DATA, ELIMINATING NOUNS THAT ARE FOUND ON THE EXCLUSION LIST OPERATION 512.

[0244] In one embodiment, at SELECT ALL NOUNS IN THE EXTRACTED TEXTUAL DATA, ELIMINATING NOUNS THAT ARE FOUND ON THE EXCLUSION LIST OPERATION 512, a group is formed of all nouns in the extracted data of SELECT A FORM FIELD TO BE LEARNED AND PREPROCESS CORPUS TO EXTRACT ELECTRONIC TEXTUAL DATA RELATING TO THE SELECTED FORM FIELD OPERATION 506 that are not found on an exclusion list. In one embodiment, determination of whether a given word is being used as a noun may be made based on a dictionary analysis of the given word, or through any other process known to those of ordinary skill or later developed.

[0245] In one embodiment, following completion of SELECT ALL NOUNS IN THE EXTRACTED TEXTUAL DATA, ELIMINATING NOUNS THAT ARE FOUND ON THE EXCLUSION LIST OPERATION 512. process flow proceeds with DETERMINE A FIRST RATIO OF A FREQUENCY EACH NOUN IS FOUND IN THE TEXT CORPUS TO A FREQUENCY THE SAME NOUN IS FOUND IN A GENERIC CORPUS OPERATION 514.

[0246] In one embodiment, at DETERMINE A FIRST RATIO OF A FREQUENCY EACH NOUN IS FOUND IN THE TEXT CORPUS TO A FREQUENCY THE SAME NOUN IS FOUND IN A GENERIC CORPUS OPERATION 514, for each given noun of SELECT ALL NOUNS IN THE EXTRACTED TEXTUAL DATA, ELIMINATING NOUNS THAT ARE FOUND ON THE EXCLUSION LIST OPERATION 512, two frequencies are determined. The first determined frequency is a frequency that the given noun is found in the text corpus formed at ACQUIRE EXTERNAL AND LOCAL TEXTUAL DATA RELATING TO A FORM HAVING FORM FIELDSTO BE LEARNED; INCORPORATE AND CONVERT ELECTRONIC AND PHYSICAL TEXTUAL DATA INTO AN ELECTRONIC CORPUS OPERATION 504.

[0247] The second determined frequency is a frequency that the given noun is found in a generic text corpus. Following determination of the first and second frequencies, they are combined in a first ratio. In one embodiment, the first ratio is formed by dividing the first determined frequency by the second determined frequency. In one embodiment, the first ratio is formed by dividing the second determined frequency by the first determined frequency.

[0248] Following completion of DETERMINE A FIRST RATIO OF A FREQUENCY EACH NOUN IS FOUND IN THE TEXT CORPUS TO A FREQUENCY THE SAME NOUN IS FOUND IN A GENERIC CORPUS OPERATION 514, process flow proceeds with DETERMINE A SECOND RATIO OF A DEGREE OF EACH NOUN TO A FREQUENCY THE SAME NOUN IS FOUND IN THE EXTRACTED WORD GROUPS OPERATION 516.

[0249] In one embodiment, at DETERMINE A SECOND RATIO OF A DEGREE OF EACH NOUN TO A FREQUENCY THE SAME NOUN IS FOUND IN THE EXTRACTED WORD GROUPS OPERATION 516, for each noun in the extracted word groups of SEPARATE THE EXTRACTED TEXTUAL DATA INTO WORD GROUPS OF N-GRAMS, OMITTING WORD GROUPS HAVING WORDS FOUND ON AN EXCLUSION LIST OPERATION 508, a first determination is made of the degree of the noun, and a second determination is made of how often the noun is reflected in the N-grams. A "degree" of a noun is the sum of the lengths of word groups (i.e. number of words in each group) which contain the noun.

[0250] The data values resulting from the first and second determinations are then combined into a second ratio. In one embodiment, the second ratio is formed by dividing the data value associated with the first determination by the data value associated with the second determination.

[0251] Following completion of DETERMINE A SECOND RATIO OF A DEGREE OF EACH NOUN TO A FREQUENCY THE SAME NOUN IS FOUND IN THE EXTRACTED WORD GROUPS OPERATION 516, process flow proceeds with COMBINE THE FIRST AND SECOND RATIOS, RESULTING IN A FINAL RATIO; SELECT WORD GROUPS MEETING FINAL RATIO ACCEPTANCE CRITERIA, ELIMINATING WORD GROUPS OUTSIDE THE CRITERIA, RESULTING IN A SECOND EXTRACTED GROUP OPERATION 518.

[0252] In one embodiment, at COMBINE THE FIRST AND SECOND RATIOS, RESULTING IN A FINAL RATIO; SELECT WORD GROUPS MEETING FINAL RATIO ACCEPTANCE CRITERIA, ELIMINATING WORD GROUPS OUTSIDE THE CRITERIA, RESULTING IN A SECOND EXTRACTED GROUP OPERATION 518, the first ratio of DETERMINE A FIRST RATIO OF A FREQUENCY EACH NOUN IS FOUND IN THE TEXT CORPUS TO A FREQUENCY THE SAME NOUN IS FOUND IN A GENERIC CORPUS OPERATION 514 and the second ratio of DETERMINE A SECOND RATIO OF A DEGREE OF EACH NOUN TO A FREQUENCY THE SAME NOUN IS FOUND IN THE EXTRACTED WORD GROUPS OPERATION 516 are combined in a final ratio. In one embodiment, the first ratio is averaged with the second ratio, giving each ratio equal weight, resulting in a final ratio for each word group. In one embodiment, word groups having final ratios that meet predetermined final ratio acceptance criteria are selected, while all other word groups not meeting final ratio acceptance criteria are eliminated or otherwise ignored, resulting in a second extracted group.

[0253] In one embodiment, following completion of COMBINE THE FIRST AND SECOND RATIOS, RESULTING IN A FINAL RATIO; SELECT WORD GROUPS MEETING FINAL RATIO ACCEPTANCE CRITERIA, ELIMINATING WORD GROUPS OUTSIDE THE CRITERIA, RESULTING IN A SECOND EXTRACTED GROUP OPERATION 518, process flow proceeds with COMBINE THE FIRST AND SECOND EXTRACTED GROUPS INTO A FINAL EXTRACTED GROUP AND REFINE ACCORDING TO REFINEMENT RULES OPERATION 520.

[0254] In one embodiment, at COMBINE THE FIRST AND SECOND EXTRACTED GROUPS INTO A FINAL EXTRACTED GROUP AND REFINE ACCORDING TO REFINEMENT RULES OPERATION 520, the first extracted group of DETERMINE A RANKING MEASURE FOR THE WORD GROUPS AND ELIMINATE WORD GROUPS OUTSIDE A RANKING MEASURE CRITERIA, RESULTING IN A FIRST EXTRACTED GROUP OPERATION 510 and the second extracted group of COMBINE THE FIRST AND SECOND RATIOS, RESULTING IN A FINAL RATIO; SELECT WORD GROUPS MEETING FINAL RATIO ACCEPTANCE CRITERIA, ELIMINATING WORD GROUPS OUTSIDE THE CRITERIA, RESULTING IN A SECOND EXTRACTED GROUP OPERATION 518 are combined into a single final extracted word group and refined according to refinement rules. In one embodiment, the refinement rules include using the final extracted word groups of COMBINE THE FIRST AND SECOND EXTRACTED GROUPS INTO A FINAL EXTRACTED GROUP AND REFINE ACCORDING TO REFINEMENT RULES OPERATION 520 and the original extracted electronic textual data of SELECT A FORM FIELD TO BE LEARNED AND PREPROCESS CORPUS TO EXTRACT ELECTRONIC

TEXTUAL DATA RELATING TO THE SELECTED FORM FIELD OPERATION 506 and performs one or more process operations in accordance with refinement rules. In one embodiment, for each sentence of the original extracted electronic textual data of SELECT A FORM FIELD TO BE LEARNED AND PREPROCESS CORPUS TO EXTRACT ELECTRONIC TEXTUAL DATA RELATING TO THE SELECTED FORM FIELD OPERATION 506, a longest extracted word group of that given sentence is determined, and a determination is made as to how many words are in that longest extracted word group. Using the determination of how many words are in that longest extracted group, any shorter word groups of the final extracted word groups of COMBINE THE FIRST AND SECOND EXTRACTED GROUPS INTO A FINAL EXTRACTED GROUP AND REFINE ACCORDING TO REFINEMENT RULES OPERATION 520 are removed from the final extracted group if those shorter word groups are only used with that longest word group and are thus now used with other unrelated word groups.

[0255] A second refinement operation of COMBINE THE FIRST AND SECOND EXTRACTED GROUPS INTO A FINAL EXTRACTED GROUP AND REFINE ACCORDING TO REFINEMENT RULES OPERATION 520 merges two or more word groups that are found in the same sentence and also share one or more common linking word. For example, if the sentence includes word groups “capital gain tax” and “gain tax worksheet,” those two word groups are combined into a single longer word group “capital gain tax worksheet” and the two or more word groups that are found in the same sentence and also share a common linking word are eliminated from the final extracted group.

[0256] A third refinement operation of COMBINE THE FIRST AND SECOND EXTRACTED GROUPS INTO A FINAL EXTRACTED GROUP AND REFINE ACCORDING TO REFINEMENT RULES OPERATION 520 merges two or more word groups that are found in the final extracted group, in the same sentence, and also share a conjunction that was not originally extracted. For example, if the final extracted group includes word groups “credit for tax” and “lump-sum distribution,” and the sentence includes both word groups with a conjunction such as “on”, those two word groups are combined with the conjunction into a single longer word group “credit for tax on lump-sum distribution” and the two or more original word groups are eliminated from the final extracted group.

[0257] A fourth refinement operation of COMBINE THE FIRST AND SECOND EXTRACTED GROUPS INTO A FINAL EXTRACTED GROUP AND REFINE ACCORDING TO REFINEMENT RULES OPERATION 520 merges word group data representing two or more word groups that are found in the final extracted group, in the same sentence, where the sentence had a possessive case and one of the two or more word groups is a possessive noun. For example, if the final extracted group includes word groups "spouse's" and "earned income," and sentence data includes both word groups with one or those word groups indicating a possessive, word group data representing those two word groups are combined into word group data representing a single longer word group "spouse's earned income" and the word group data representing the two or more original word groups are eliminated from the word group data representing the final extracted group.

[0258] A fifth refinement operation of COMBINE THE FIRST AND SECOND EXTRACTED GROUPS INTO A FINAL EXTRACTED GROUP AND REFINE ACCORDING TO REFINEMENT RULES OPERATION 520 where if a noun is in a "group" with other terms, data representing that noun is added as a single word group on its own, if the final extracted group data representing the final extracted group didn't have it already.

[0259] In one embodiment, following completion of COMBINE THE FIRST AND SECOND EXTRACTED GROUPS INTO A FINAL EXTRACTED GROUP AND REFINE ACCORDING TO REFINEMENT RULES OPERATION 520, process flow proceeds with ORGANIZE THE REFINED FINAL EXTRACTED GROUP IN A HIERARCHY OPERATION 522.

[0260] In one embodiment, at ORGANIZE THE REFINED FINAL EXTRACTED GROUP IN A HIERARCHY OPERATION 522, the final extracted group data representing the final extracted group includes, in various embodiments, one or more single words as single word groups, and one or more multiple-word word groups. And, in one embodiment, the single word groups are also found within the multiple word groups. For example, in one embodiment, the final extracted group data representing the final extracted group includes "interest," "mortgage interest," "home mortgage interest," excess mortgage interest," and "deductible mortgage interest."

[0261] In one embodiment, a word of the word groups having common words is designated as a most important word, and a hierarchy is formed using the most important word as a "parent word" of the groups. Other word groups containing the parent term Thus, in the example above, if the word "interest" is determined to be an important term, a hierarchy is formed using "interest" as the head term. Correspondingly, the other word groups each have "mortgage interest" as common words. Thus, "mortgage interest" may also be used as a parent group, below the head term "interest."

[0262] In one embodiment, the example hierarchy is thus formed as a tree of groups of word group data from the final extracted group data, and looks like

```
interest
    mortgage interest
        home mortgage interest
        excess mortgage interest
        deductible mortgage interest
```

[0263] If additional terms that included one of the parent terms were in the final extracted group data, a longer tree would include those words, such as, in one embodiment,

```
interest
    mortgage interest
        home mortgage interest
        excess mortgage interest
        deductible mortgage interest
    bond interest
        saving bond interest
            excludable savings bond interest
```

[0264] Organizing the word group data of the terms in such a tree makes it easy to know which terms survived the process and thus which terms are the most important for a given form, or for a given genre of document. For example, word group data of a first tree might indicate important word groups in the tax genre, while word group data of a second tree might indicate important word groups in the retail invoice genre.

[0265] In one embodiment, following completion of ORGANIZE THE REFINED FINAL EXTRACTED GROUP IN A HIERARCHY OPERATION 522, process flow proceeds with OUTPUT THE FINAL EXTRACTED GROUP OPERATION 524.

[0266] In one embodiment, at OUTPUT THE FINAL EXTRACTED GROUP OPERATION 524, results of the natural language parsing processes of process 500 for learning and incorporating new and/or updated forms in an electronic document preparation system are provided to one or more of process 300 for learning and incorporating new and/or updated forms in an electronic document preparation system and process 400 for learning and incorporating new and/or updated forms in an electronic document preparation system.

[0267] In one embodiment, following completion of OUTPUT THE FINAL EXTRACTED GROUP OPERATION 524, process flow proceeds with END OPERATION 526 where the process exist awaiting further input.

[0268] As noted above, the specific illustrative examples discussed above are but illustrative examples of implementations of embodiments of the method or process for learning and incorporating new and/or updated forms in an electronic document preparation system. Those of skill in the art will readily recognize that other implementations and embodiments are possible. Therefore, the discussion above should not be construed as a limitation on the claims provided herein.

[0269] In one embodiment, a computing system implements a method for learning and incorporating new and/or updated forms in an electronic document preparation system. The method includes receiving form data related to a new and/or updated form having a plurality of data fields and gathering training set data related to previously filled forms. Each previously filled form has completed data fields that each correspond to a respective data field of the new and/or updated form. The method also includes generating, for a first selected data field from the plurality of data fields of the new and/or updated form, candidate function data including a plurality of candidate input functions for providing a proper data value for the first selected data field, generating, for each candidate function, test data by applying the candidate function to the training set data, and generating, for each candidate function, matching data by comparing the test data to the completed data fields corresponding to the first selected data field. The matching data indicates how closely the test data matches the corresponding completed data fields of the previously filled forms. The method also includes identifying, from the plurality of functions, an acceptable candidate function for the first data field of the new and/or updated form by determining, for each candidate function, whether or not the candidate function is an acceptable function for the first selected data field of the new and/or updated form based on the matching data. The method also includes generating, after identifying an acceptable function for the first data field, results data indicating an acceptable function for the first data field of the new and/or updated form and outputting the results data.

[0270] In one embodiment, a non-transitory computer-readable medium has a plurality of computer-executable instructions which, when executed by a processor, perform a method for learning and incorporating new and/or updated forms in an electronic document preparation system. The instructions include an interface module configured to receive form data representing to a new and/or updated form having a plurality of data fields and a data acquisition module configured to gather training set data related to previously filled forms. Each previously filled form has completed data fields that each correspond to a respective data field of the new and/or updated form. The instructions also include a machine learning module configured to identify a respective acceptable function for each of the data fields of the new and/or updated form by generating candidate function data relating to a plurality of candidate functions, generating test data by applying the candidate functions to the training set data, and finding, for each of the data fields a respective acceptable function from the plurality of candidate functions based on a how closely the test data matches the candidate function data.

[0271] One embodiment is a system for learning and incorporating new and/or updated forms in an electronic document preparation system. The system includes one or more computing processors and at least one memory coupled to the at least one computing processor, the at least one memory having stored therein instructions which, when executed by any set of the one or more processors, perform a process. The process includes receiving, with an interface module of a computing system, form data related to a new and/or updated form having a plurality of data fields and gathering training set data related to previously filled forms. Each previously filled form has completed data fields that each correspond to a respective data field of the new and/or updated form. The process also includes generating, with a data acquisition module of a computing system, for a first selected data field from the plurality of data fields of the new and/or updated form, candidate function data including a plurality of candidate input functions for providing a proper data value for the first selected data field. The process also includes generating, with a machine learning module of a computing system, for each candidate function, test data by applying the candidate function to the training set data and generating, for each candidate function, matching data by comparing the test data to the completed data fields corresponding to the first selected data field. The matching data indicates how closely the test data matches the corresponding completed data fields of the previously filled forms. The process also includes identifying, with the machine learning module, from the plurality of functions, an acceptable candidate function for the first data field of the new and/or updated form by determining, for each candidate function, whether or not the candidate function is an acceptable function for the first selected data field of the new and/or updated form based on the matching data. The process also includes generating, with the machine learning module, after identifying an acceptable function for the first data field, results data indicating an acceptable function for the first data field of the new and/or updated form and outputting, with the interface module, the results data.

[0272] One embodiment is a computing system implemented method for learning and incorporating new and/or updated forms in an electronic document preparation system. The method includes receiving form data related to a new and/or updated form having a plurality of data fields, gathering training set data related to previously filled forms. Each previously filled form has completed data fields that each correspond to a respective data field of the new and/or updated form. The method also includes generating, for a first selected data field of the plurality of data fields of the new and/or updated form, dependency data indicating one or more possible dependencies for an acceptable function that provides a proper data value for the first selected data field. The method further includes generating, for the first selected data field, candidate function data including a plurality of candidate functions based on the dependency data and one or more operators selected from a library of operators, generating, for each candidate function, test data by applying the candidate function to the training set data, and generating, for each candidate function, matching data by comparing the test data to the completed data fields corresponding to the first selected data field, the matching data indicating how closely the test data matches the corresponding completed data fields of the previously filled forms. The method also includes identifying, from the plurality of functions, an acceptable candidate function for the first selected data field of the new and/or updated form by determining, for each candidate function, whether or not the candidate function is an acceptable function for the first selected data field of the new and/or updated form based on the matching data, generating, after identifying an acceptable function for the first data field, results data indicating an acceptable for the first data field of the new and/or updated form, and outputting the results data.

[0273] One embodiment is a non-transitory computer-readable medium having a plurality of computer-executable instructions which, when executed by a processor, perform a method for learning and incorporating new and/or updated forms in an electronic document preparation system. The instructions include an interface module configured to receive form data representing to a new and/or updated form having a plurality of data fields. The instructions include a data acquisition module configured to gather training set data related to previously filled forms. Each previously filled form has completed data fields that each correspond to a respective data field of the new and/or updated form. The instructions also include a machine learning module configured to identify a respective acceptable function for each of the data fields of the new and/or updated form by generating candidate function data relating to a plurality of candidate functions based on dependency data indicating possible dependencies for each data field of the new and/or updated form and including one or more operators from a library of operators, generating test data by applying the candidate functions to the training set data, and finding, for each of the data fields a respective acceptable function from the plurality of candidate functions based on a how closely the test data matches the candidate function data.

[0274] One embodiment is a system for learning and incorporating new and/or updated forms in an electronic document preparation system. The system includes at least one processor at least one memory coupled to the at least one processor. The at least one memory has stored therein instructions which, when executed by any set of the one or more processors, perform a process. The process includes receiving, with an interface module of a computing system, form data related to a new and/or updated form having a plurality of data fields, gathering, with a data acquisition module of a computing system, training set data related to previously filled forms. Each previously filled form has completed data fields that each correspond to a respective data field of the new and/or updated form. The process also includes generating, with a machine learning module of a computing system, for a first selected data field of the plurality of data fields of the new and/or updated form, dependency data indicating one or more possible dependencies for an acceptable function that provides a proper data value for the first selected data field. The process also includes generating, with the machine learning module, for the first selected data field, candidate function data including a plurality of candidate functions based on the dependency data and one or more operators selected from a library of operators, generating, with the machine learning module, for each candidate function, test data by applying the candidate function to the training set data, and generating, with the machine learning module, for each candidate function, matching data by comparing the test data to the completed data fields corresponding to the first selected data field, the matching data indicating how closely the test data matches the corresponding completed data fields of the previously filled forms. The process also includes identifying, with the machine learning module, from the plurality of functions, an acceptable candidate function for the first selected data field of the new and/or updated form by determining, for each candidate function, whether or not the candidate function is an acceptable function for the first selected data field of the new and/or updated form based on the matching data, generating, with the machine learning module and after identifying the correct function for the first data field, results data indicating an acceptable function for the first data field of the new and/or updated form, and outputting, with the interface module, the results data.

[0275] Using the disclosed embodiments of a method and system for learning and incorporating new and/or updated forms in an electronic document preparation system, a method and system for learning and incorporating new and/or updated forms in an electronic document preparation system more accurately is provided. Therefore, the disclosed embodiments provide a technical solution to the long standing technical problem of efficiently learning and incorporating new and/or updated forms in an electronic document preparation system.

[0276] In the discussion above, certain aspects of one embodiment include process steps and/or operations and/or instructions described herein for illustrative purposes in a particular order and/or grouping. However, the particular order and/or grouping shown and discussed herein are illustrative only and not limiting. Those of skill in the art will recognize that other orders and/or grouping of the process steps and/or operations and/or instructions are possible and, in some embodiments, one or more of the process steps and/or operations and/or instructions discussed above can be combined and/or deleted. In addition, portions of one or more of the process steps and/or operations and/or instructions can be re-grouped as portions of one or more other of the process steps and/or operations and/or instructions discussed herein. Consequently, the particular order and/or grouping of the process steps and/or operations and/or instructions discussed herein do not limit the scope of the invention as claimed below.

[0277] As discussed in more detail above, using the above embodiments, with little or no modification and/or input, there is considerable flexibility, adaptability, and opportunity for customization to meet the specific needs of various parties under numerous circumstances.

[0278] In the discussion above, certain aspects of one embodiment include process steps and/or operations and/or instructions described herein for illustrative purposes in a particular order and/or grouping. However, the particular order and/or grouping shown and discussed herein are illustrative only and not limiting. Those of skill in the art will recognize that other orders and/or grouping of the process steps and/or operations and/or instructions are possible and, in some embodiments, one or more of the process steps and/or operations and/or instructions discussed above can be combined and/or deleted. In addition, portions of one or more of the process steps and/or operations and/or instructions can be re-grouped as portions of one or more other of the process steps and/or operations and/or instructions discussed herein. Consequently, the particular order and/or grouping of the process steps and/or operations and/or instructions discussed herein do not limit the scope of the invention as claimed below.

[0279] The present invention has been described in particular detail with respect to specific possible embodiments. Those of skill in the art will appreciate that the invention may be practiced in other embodiments. For example, the nomenclature used for components, capitalization of component designations and terms, the attributes, data structures, or any other programming or structural aspect is not significant, mandatory, or limiting, and the mechanisms that implement the invention or its features can have various different names, formats, or protocols. Further, the system or functionality of the invention may be implemented via various combinations of software and hardware, as described, or entirely in hardware elements. Also, particular divisions of functionality between the various components described herein are merely exemplary, and not mandatory or significant. Consequently, functions performed by a single component may, in other embodiments, be performed by multiple components, and functions performed by multiple components may, in other embodiments, be performed by a single component.

[0280] Some portions of the above description present the features of the present invention in terms of algorithms and symbolic representations of operations, or algorithm-like representations, of operations on information/data. These algorithmic or algorithm-like descriptions and representations are the means used by those of skill in the art to most effectively and efficiently convey the substance of their work to others of skill in the art. These operations, while described functionally or logically, are understood to be implemented by computer programs or computing systems. Furthermore, it has also proven convenient at times to refer to these arrangements of operations as steps or modules or by functional names, without loss of generality.

[0281] Unless specifically stated otherwise, as would be apparent from the above discussion, it is appreciated that throughout the above description, discussions utilizing terms such as, but not limited to, “activating”, “accessing”, “adding”, “aggregating”, “alerting”, “applying”, “analyzing”, “associating”, “calculating”, “capturing”, “categorizing”, “classifying”, “comparing”, “creating”, “defining”, “detecting”, “determining”, “distributing”, “eliminating”, “encrypting”, “extracting”, “filtering”, “forwarding”, “generating”, “identifying”, “implementing”, “informing”, "monitoring", “obtaining”, “posting”, “processing”, “providing”, “receiving”, "requesting", “saving”, “sending”, “storing”, “substituting”, “transferring”, “transforming”, “transmitting”, “using”, etc., refer to the action and process of a computing system or similar electronic device that manipulates and operates on data represented as physical (electronic) quantities within the computing system memories, resisters, caches or other information storage, transmission or display devices.

[0282] The present invention also relates to an apparatus or system for performing the operations described herein. This apparatus or system may be specifically constructed for the required purposes, or the apparatus or system can comprise a general purpose system selectively activated or configured/reconfigured by a computer program stored on a computer program product as discussed herein that can be accessed by a computing system or other device.

[0283] Those of skill in the art will readily recognize that the algorithms and operations presented herein are not inherently related to any particular computing system, computer architecture, computer or industry standard, or any other specific apparatus. Various general purpose systems may also be used with programs in accordance with the teaching herein, or it may prove more convenient/efficient to construct more specialized apparatuses to perform the required operations described herein. The required structure for a variety of these systems will be apparent to those of skill in the art, along with equivalent variations. In addition, the present invention is not described with reference to any particular programming language and it is appreciated that a variety of programming languages may be used to implement the teachings of the present invention as described herein, and any references to a specific language or languages are provided for illustrative purposes only and for enablement of the contemplated best mode of the invention at the time of filing.

[0284] The present invention is well suited to a wide variety of computer network systems operating over numerous topologies. Within this field, the configuration and management of large networks comprise storage devices and computers that are communicatively coupled to similar or dissimilar computers and storage devices over a private network, a LAN, a WAN, a private network, or a public network, such as the Internet.

[0285] It should also be noted that the language used in the specification has been principally selected for readability, clarity and instructional purposes, and may not have been selected to delineate or circumscribe the inventive subject matter. Accordingly, the disclosure of the present invention is intended to be illustrative, but not limiting, of the scope of the invention, which is set forth in the claims below.

[0286] In addition, the operations shown in the figures, or as discussed herein, are identified using a particular nomenclature for ease of description and understanding, but other nomenclature is often used in the art to identify equivalent operations.

[0287] Therefore, numerous variations, whether explicitly provided for by the specification or implied by the specification or not, may be implemented by one of skill in the art in view of this disclosure.

The embodiments of the present invention for which an exclusive property or privilege is claimed are defined as follows:

1. A computing system implemented method for learning and incorporating forms in an electronic document preparation system, the method comprising:

obtaining form data from one or more portions of a physical document;

converting the obtained form data into electronic textual data relating to a first data field of a form for which a function needs to be determined;

separating the electronic textual data into distinct data sets representing different word groups, omitting distinct data sets representing word groups which include one or more predetermined exclusion words, resulting in separated textual data;

determining usage frequency data representing a usage frequency for word groups of the separated textual data and eliminating separated textual data word groups from the separated textual data that are outside a predetermined usage frequency criteria, resulting in first extracted group data representing a first extracted word group;

determining first ratio data representing first ratios of a frequency each noun appears within the first extracted group data also found in the electronic textual data to a frequency the same noun appears in a generic text corpus;

determining second ratio data representing second ratios of a degree of each noun within the first extracted group to a frequency the same noun is found in the first extracted group data;

operating on the first ratio data and the second ratio data to combine the first and second ratios, resulting in final ratio data representing a final ratio, and selecting word groups from the first extracted group meeting final acceptance data representing final ratio acceptance criteria, resulting in second extracted group data representing a second extracted word group;

combining the first extracted group data and the second extracted group data representing first and second extracted word groups into final extracted group data representing a final extracted word group and refine the resulting combination according to refinement rules, resulting in refined word group data representing a refined word group;

structuring the refined word group as nodes and leaves in a hierarchy according to function rules, resulting in function data representing one or more functions of the first data field; and incorporating at least a portion of the function data into an electronic document preparation system to generate updated forms.

2. The computing system implemented method for learning and incorporating forms in an electronic document preparation system of Claim 1 wherein the refinement rules require a preference for keeping longer word groups that include shorter word groups and eliminating shorter word groups that are always found inside longer word groups.

3. The computing system implemented method for learning and incorporating forms in an electronic document preparation system of Claim 1 further comprising:

for each given word group of the final extracted group data:

determining, by examining the final extracted group data, a word length of the given word group;

determining, by examining the electronic textual data, that the given word group only appears together with word groups of the final extracted word group that are longer than the given word group; and

ensuring that the given word group does not appear in the refined word group data.

4. The computing system implemented method for learning and incorporating forms in an electronic document preparation system of Claim 1 wherein the refinement rules trigger merging, prior to finalizing the refined word group data, multiple smaller word groups related to the same form field into a single larger word group and eliminating the multiple smaller word groups.

5. The computing system implemented method for learning and incorporating forms in an electronic document preparation system of Claim 1 further comprising:

selecting first word data representing a first word group of the final extracted group data, the first word group having a plurality of words;

determining, by examining the final extracted group data, at least second word data representing a second word group of the final extracted group that shares at least one common word of the first word group;

determining that the first word group represented by the first word data contains the common word at the end of the first word group and that the second word data contains the common word at the beginning of the first word group;

combining the first word group data and the second word group data into a third word group represented by third word group data.

6. The computing system implemented method for learning and incorporating forms in an electronic document preparation system of Claim 5 wherein combining the first word group data and the second word group data into a third word group represented by third word group data further comprises:

combining the first word group data and the second word group data into a third word group represented by third word group data resulting in the third word group including a portion of first word group data followed by at least a portion of second word group data.

7. The computing system implemented method for learning and incorporating forms in an electronic document preparation system of Claim 6 wherein combining the first word group data and the second word group data into a third word group represented by third word group data further comprises:

eliminating data representing the common word from one of either the first word data or the second word data, resulting in modified data;

if the common word was eliminated from the first word data, forming the third word data by combining the modified data followed by the second word data; and

if the common word was eliminated from second word data, forming third word data by combining the first word data followed by the modified data.

8. The computing system implemented method for learning and incorporating forms in an electronic document preparation system of Claim 1 wherein the refinement rules trigger determining word groups of the final extracted group that were previously connected by one or more conjunctions in the electronic textual data, combining those determined word groups and the one or more conjunctions, and eliminating the word groups.

9. The computing system implemented method for learning and incorporating forms in an electronic document preparation system of Claim 8 wherein the one or more conjunctions include at least one conjunction from the group of conjunctions consisting of "of", "in", "to", "in", "for" and "on."

10. The computing system implemented method for learning and incorporating forms in an electronic document preparation system of Claim 1 comprising examining the electronic textual data for nouns that are grouped with refinement data word groups, adding those nouns to the refinement data if they are not already present within the refinement data.

11. The computing system implemented method for learning and incorporating forms in an electronic document preparation system of Claim 1 wherein at least a portion of training set data is applied to one or more functions of the function data, resulting in test data, and

analyzing the test data to determine a degree of accuracy of the one or more functions of the function data.

12. The computing system implemented method for learning and incorporating forms in an electronic document preparation system of Claim 11 wherein applying at least a portion of the training set data to one or more functions of the function data includes substituting one or more data values for at least one field-related dependency.

13. The computing system implemented method for learning and incorporating forms in an electronic document preparation system of Claim 1, further comprising

generating, for the first data field, dependency data indicating one or more dependencies, wherein the dependencies include one or more of:
a second data field from a form associated with the first data field;
multiple data fields from the form associated with the first data field;
a data field from a form other than the form associated with the first data field;
multiple data fields from multiple different forms; and
a constant.

14. The computing system implemented method for learning and incorporating forms in an electronic document preparation system of Claim 1, wherein the first data field is a field of one of a new or updated tax form.

15. The computing system implemented method for learning and incorporating forms in an electronic document preparation system of Claim 11, wherein the training set data includes previously prepared tax returns.

16. The computing system implemented method for learning and incorporating forms in an electronic document preparation system of Claim 11, wherein the training set data includes fabricated tax returns.

17. A computing system implemented system for learning and incorporating forms in an electronic document preparation system comprising:

one or more computing processors;

one or more memories coupled to the one or more computing processors, the one or more memories having stored therein which when executed by the one or more computing processors perform a process for learning and incorporating forms in an electronic document preparation system comprising:

obtaining form data from one or more portions of a physical document;

converting the form data into electronic textual data relating to a first data field of a form for which a function needs to be determined;

separating the electronic textual data into distinct data sets representing different word groups, omitting distinct data sets representing word groups which include one or more predetermined exclusion words, resulting in separated textual data;

determining usage frequency data representing a usage frequency for word groups of the separated textual data and eliminating separated textual data word groups from the separated textual data that are outside a predetermined usage frequency criteria, resulting in first extracted group data representing a first extracted word group;

determining first ratio data representing first ratios of a frequency each noun appears within the first extracted group data also found in the electronic textual data to a frequency the same noun appears in a generic text corpus;

determining second ratio data representing second ratios of a degree of each noun within the first extracted group to a frequency the same noun is found in the first extracted group data;

operating on the first ratio data and the second ratio data to combine the first and second ratios, resulting in final ratio data representing a final ratio, and selecting word groups from the first extracted group meeting final acceptance data representing final ratio acceptance criteria, resulting in second extracted group data representing a second extracted word group;

combining the first extracted group data and the second extracted group data representing first and second extracted word groups into final extracted group data representing a final extracted word group and refine the resulting combination according to refinement rules, resulting in refined word group data representing a refined word group;

structuring the refined word group as nodes and leaves in a hierarchy according to function rules, resulting in function data representing one or more functions of the first data field; and

incorporating at least a portion of the function data into an electronic document preparation system to generate updated forms.

18. The computing system implemented system for learning and incorporating forms in an electronic document preparation system of Claim 17 wherein the refinement rules require a preference for keeping longer word groups that include shorter word groups and eliminating shorter word groups that are always found inside longer word groups.

19. The computing system implemented system for learning and incorporating forms in an electronic document preparation system of Claim 17 further comprising:

for each given word group of the final extracted group data:

determining, by examining the final extracted group data, a word length of the given word group;

determining, by examining the electronic textual data, that the given word group only appears together with word groups of the final extracted word group that are longer than the given word group; and

ensuring that the given word group does not appear in the refined word group data.

20. The computing system implemented system for learning and incorporating forms in an electronic document preparation system of Claim 17 wherein the refinement rules trigger merging, prior to finalizing the refined word group data, multiple smaller word groups related to the same form field into a single larger word group and eliminating the multiple smaller word groups.

21. The computing system implemented system for learning and incorporating forms in an electronic document preparation system of Claim 17 further comprising:

selecting first word data representing a first word group of the final extracted group data, the first word group having a plurality of words;

determining, by examining the final extracted group data, at least second word data representing a second word group of the final extracted group that shares at least one common word of the first word group;

determining that the first word group represented by the first word data contains the common word at the end of the first word group and that the second word data contains the common word at the beginning of the first word group;

combining the first word group data and the second word group data into a third word group represented by third word group data.

22. The computing system implemented system for learning and incorporating forms in an electronic document preparation system of Claim 21 wherein combining the first word group data and the second word group data into a third word group represented by third word group data further comprises:

combining the first word group data and the second word group data into a third word group represented by third word group data resulting in the third word group including a portion of first word group data followed by at least a portion of second word group data.

23. The computing system implemented system for learning and incorporating forms in an electronic document preparation system of Claim 22 wherein combining the first word group data and the second word group data into a third word group represented by third word group data further comprises:

eliminating data representing the common word from one of either the first word data or the second word data, resulting in modified data;

if the common word was eliminated from the first word data, forming the third word data by combining the modified data followed by the second word data; and

if the common word was eliminated from second word data, forming third word data by combining the first word data followed by the modified data.

24. The computing system implemented system for learning and incorporating forms in an electronic document preparation system of Claim 17 wherein the refinement rules trigger determining word groups of the final extracted group that were previously connected by one or more conjunctions in the electronic textual data, combining those determined word groups and the one or more conjunctions, and eliminating the word groups.

25. The computing system implemented system for learning and incorporating forms in an electronic document preparation system of Claim 24 wherein the one or more conjunctions include at least one conjunction from the group of conjunctions consisting of "of", "in", "to", "in", "for" and "on."

26. The computing system implemented system for learning and incorporating forms in an electronic document preparation system of Claim 17 comprising examining the electronic textual data for nouns that are grouped with refinement data word groups, adding those nouns to the refinement data if they are not already present within the refinement data.

27. The computing system implemented system for learning and incorporating forms in an electronic document preparation system of Claim 17 wherein at least a portion of training set data is applied to one or more functions of the function data, resulting in test data, and

analyzing the test data to determine a degree of accuracy of the one or more functions of the function data.

28. The computing system implemented system for learning and incorporating forms in an electronic document preparation system of Claim 27 wherein applying at least a portion of the training set data to one or more functions of the function data includes substituting one or more data values for at least one field-related dependency.

29. The computing system implemented system for learning and incorporating forms in an electronic document preparation system of Claim 17, further comprising

generating, for the first data field, dependency data indicating one or more dependencies, wherein the dependencies include one or more of:
a second data field from a form associated with the first data field;
multiple data fields from the form associated with the first data field;
a data field from a form other than the form associated with the first data field;
multiple data fields from multiple different forms; and
a constant.

30. The computing system implemented system for learning and incorporating forms in an electronic document preparation system of Claim 17, wherein the first data field is a field of one of a new or updated tax form.

31. The computing system implemented system for learning and incorporating forms in an electronic document preparation system of Claim 27, wherein the training set data includes previously prepared tax returns.

32. The computing system implemented system for learning and incorporating forms in an electronic document preparation system of Claim 27, wherein the training set data includes fabricated tax returns.

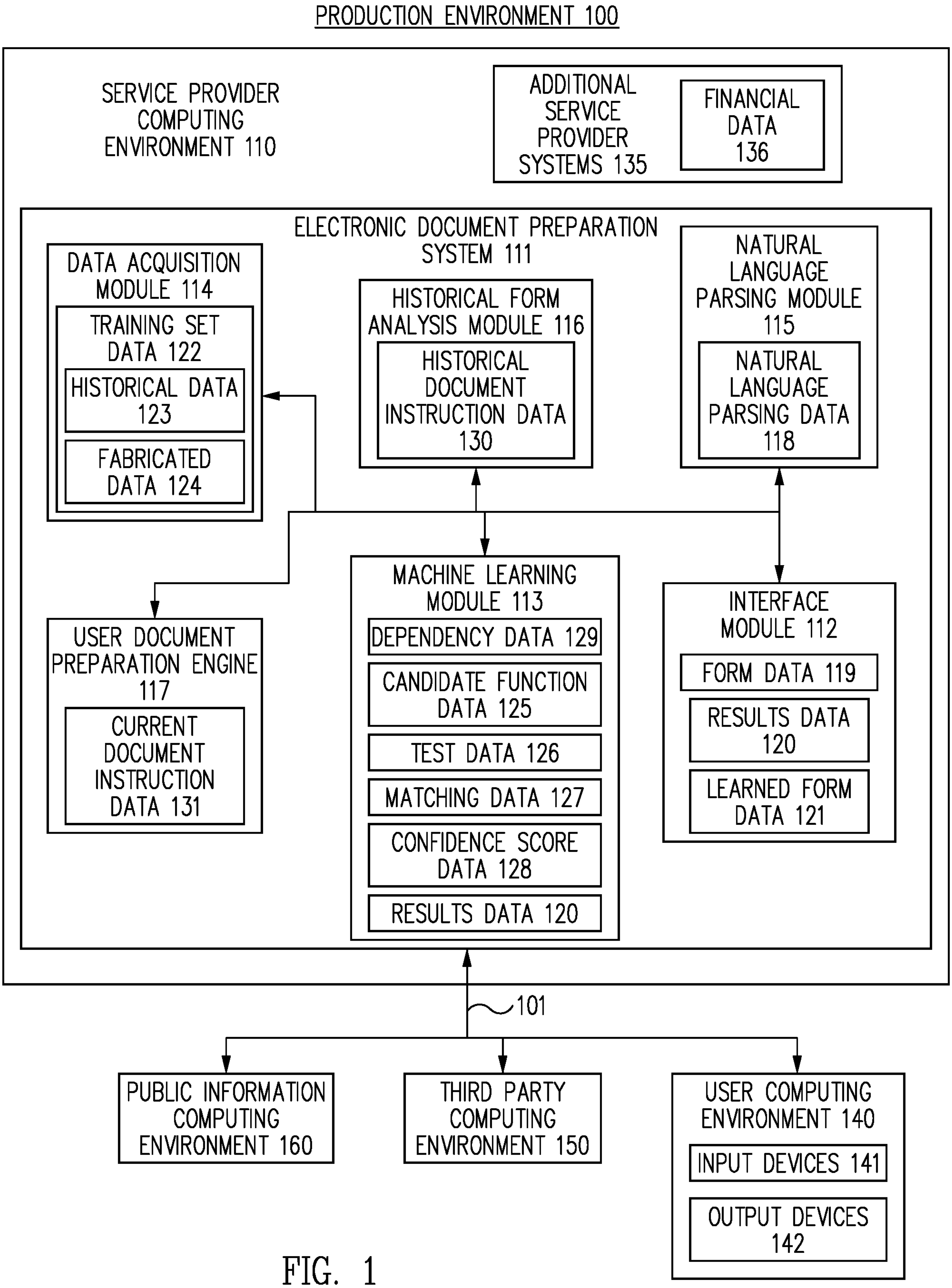

PRODUCTION ENVIRONMENT 100
SERVICE PROVIDER COMPUTING ENVIRONMENT 110
ADDITIONAL SERVICE PROVIDER SYSTEMS 135
FINANCIAL DATA 136
ELECTRONIC DOCUMENT PREPARATION SYSTEM 111
DATA ACQUISITION MODULE 114
TRAINING SET DATA 122
HISTORICAL DATA 123
FABRICATED DATA 124
HISTORICAL FORM ANALYSIS MODULE 116
HISTORICAL DOCUMENT INSTRUCTION DATA 130
NATURAL LANGUAGE PARSING MODULE 115
NATURAL LANGUAGE PARSING DATA 118
USER DOCUMENT PREPARATION ENGINE 117
CURRENT DOCUMENT INSTRUCTION DATA 131
MACHINE LEARNING MODULE 113
DEPENDENCY DATA 129
CANDIDATE FUNCTION DATA 125
TEST DATA 126
MATCHING DATA 127
CONFIDENCE SCORE DATA 128
RESULTS DATA 120
INTERFACE MODULE 112
FORM DATA 119
RESULTS DATA 120
LEARNED FORM DATA 121
101
PUBLIC INFORMATION COMPUTING ENVIRONMENT 160
THIRD PARTY COMPUTING ENVIRONMENT 150
USER COMPUTING ENVIRONMENT 140
INPUT DEVICES 141
OUTPUT DEVICES 142

FIG. 1

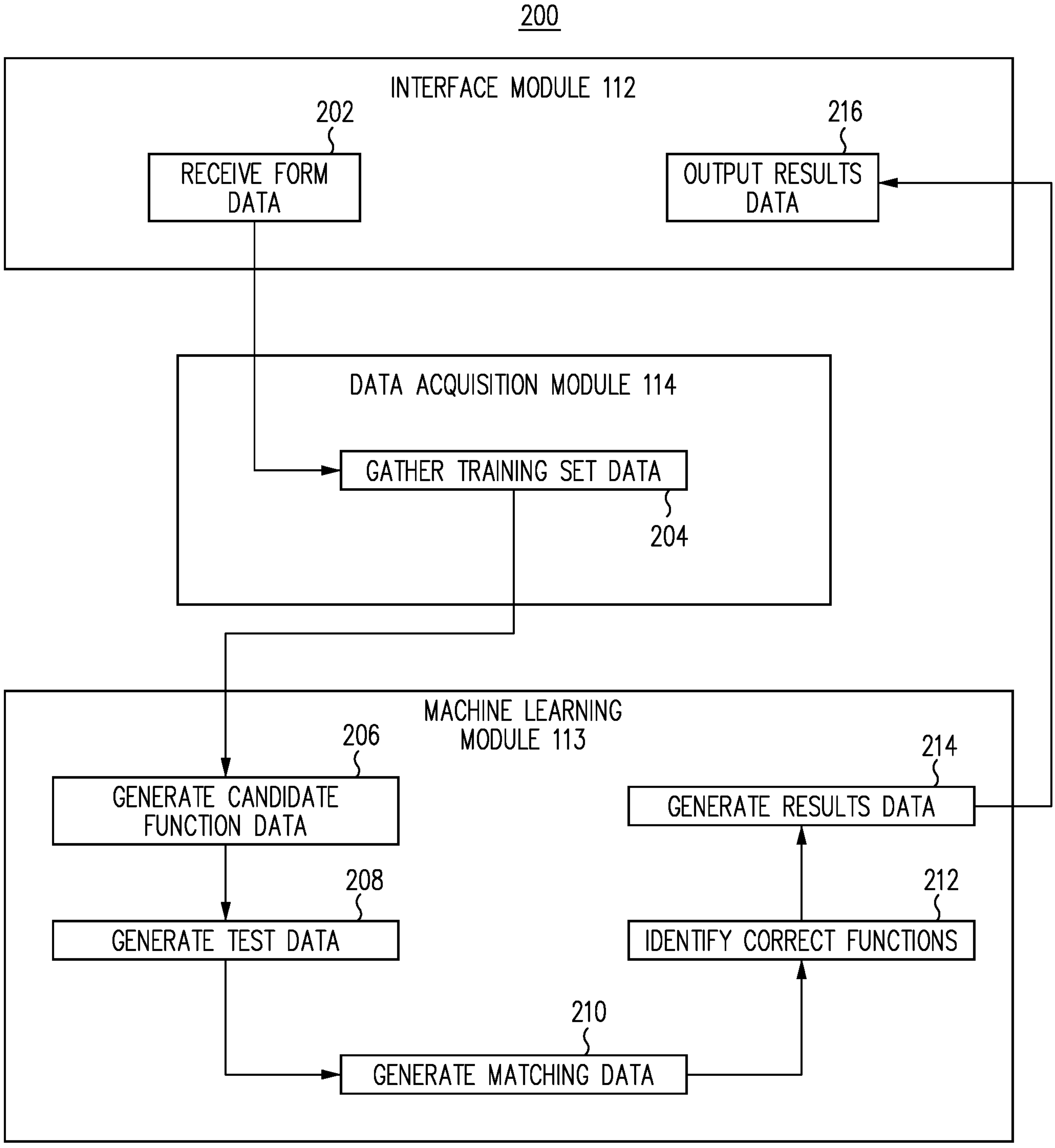

200
INTERFACE MODULE 112
202
RECEIVE FORM DATA
216
OUTPUT RESULTS DATA
DATA ACQUISITION MODULE 114
GATHER TRAINING SET DATA
204
MACHINE LEARNING MODULE 113
206
GENERATE CANDIDATE FUNCTION DATA
214
GENERATE RESULTS DATA
208
GENERATE TEST DATA
212
IDENTIFY CORRECT FUNCTIONS
210
GENERATE MATCHING DATA

FIG. 2

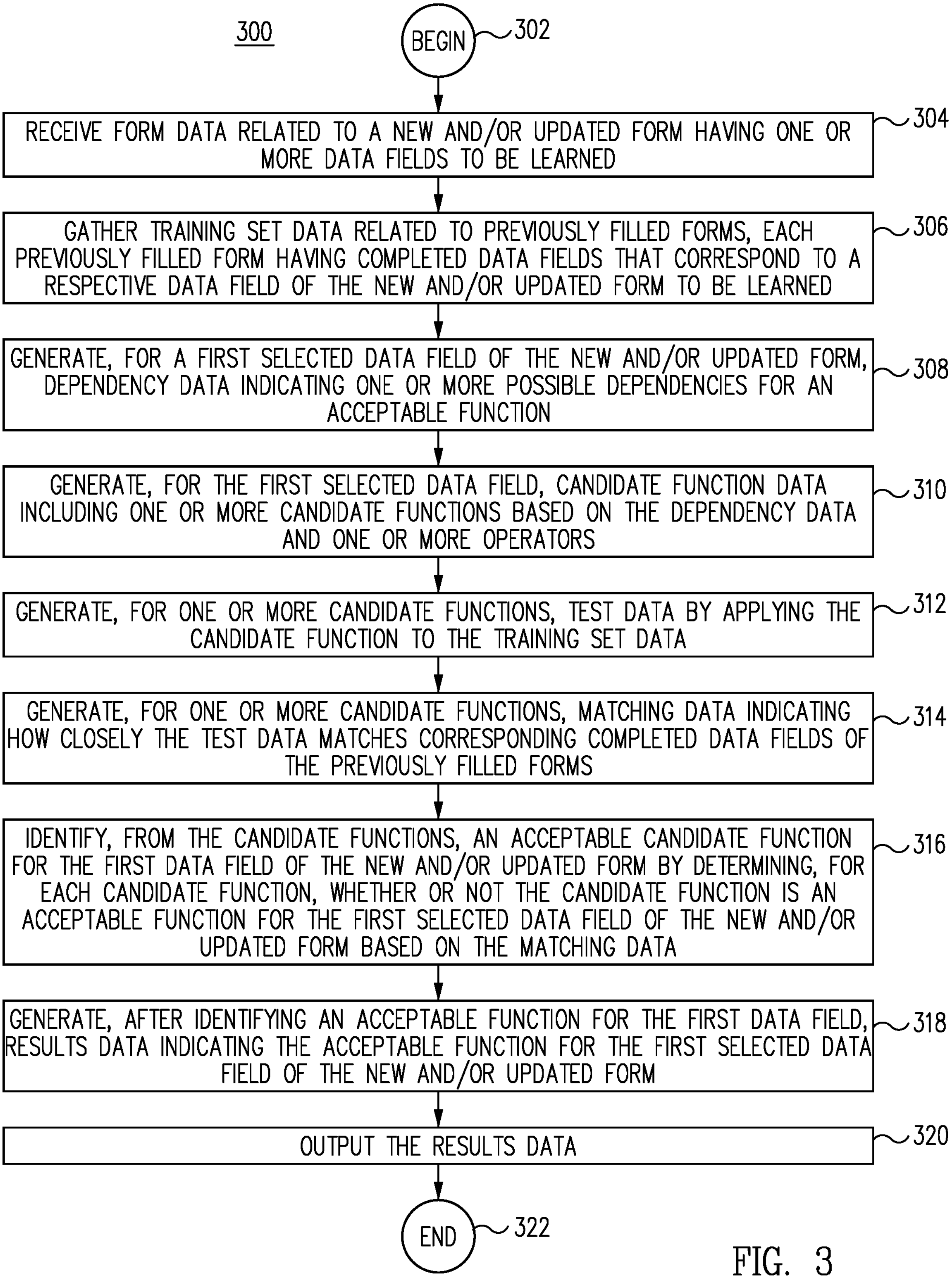

300
BEGIN
302
RECEIVE FORM DATA RELATED TO A NEW AND/OR UPDATED FORM HAVING ONE OR MORE DATA FIELDS TO BE LEARNED
304
GATHER TRAINING SET DATA RELATED TO PREVIOUSLY FILLED FORMS, EACH PREVIOUSLY FILLED FORM HAVING COMPLETED DATA FIELDS THAT CORRESPOND TO A RESPECTIVE DATA FIELD OF THE NEW AND/OR UPDATED FORM TO BE LEARNED
306
GENERATE, FOR A FIRST SELECTED DATA FIELD OF THE NEW AND/OR UPDATED FORM, DEPENDENCY DATA INDICATING ONE OR MORE POSSIBLE DEPENDENCIES FOR AN ACCEPTABLE FUNCTION
308
GENERATE, FOR THE FIRST SELECTED DATA FIELD, CANDIDATE FUNCTION DATA INCLUDING ONE OR MORE CANDIDATE FUNCTIONS BASED ON THE DEPENDENCY DATA AND ONE OR MORE OPERATORS
310
GENERATE, FOR ONE OR MORE CANDIDATE FUNCTIONS, TEST DATA BY APPLYING THE CANDIDATE FUNCTION TO THE TRAINING SET DATA
312
GENERATE, FOR ONE OR MORE CANDIDATE FUNCTIONS, MATCHING DATA INDICATING HOW CLOSELY THE TEST DATA MATCHES CORRESPONDING COMPLETED DATA FIELDS OF THE PREVIOUSLY FILLED FORMS
314
IDENTIFY, FROM THE CANDIDATE FUNCTIONS, AN ACCEPTABLE CANDIDATE FUNCTION FOR THE FIRST DATA FIELD OF THE NEW AND/OR UPDATED FORM BY DETERMINING, FOR EACH CANDIDATE FUNCTION, WHETHER OR NOT THE CANDIDATE FUNCTION IS AN ACCEPTABLE FUNCTION FOR THE FIRST SELECTED DATA FIELD OF THE NEW AND/OR UPDATED FORM BASED ON THE MATCHING DATA
316
GENERATE, AFTER IDENTIFYING AN ACCEPTABLE FUNCTION FOR THE FIRST DATA FIELD, RESULTS DATA INDICATING THE ACCEPTABLE FUNCTION FOR THE FIRST SELECTED DATA FIELD OF THE NEW AND/OR UPDATED FORM
318
OUTPUT THE RESULTS DATA
320
END
322

FIG. 3

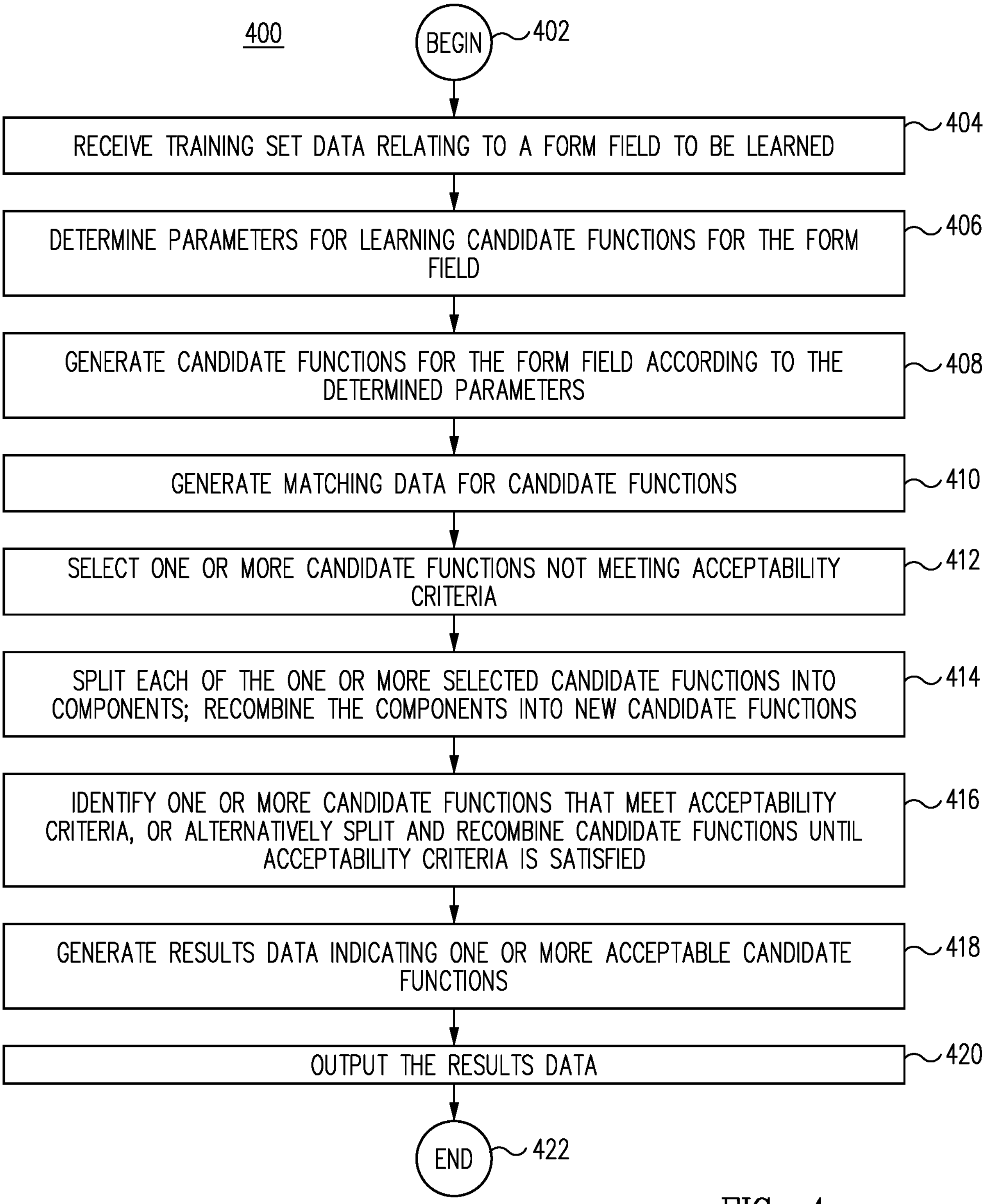

400
BEGIN
402
RECEIVE TRAINING SET DATA RELATING TO A FORM FIELD TO BE LEARNED
404
DETERMINE PARAMETERS FOR LEARNING CANDIDATE FUNCTIONS FOR THE FORM FIELD
406
GENERATE CANDIDATE FUNCTIONS FOR THE FORM FIELD ACCORDING TO THE DETERMINED PARAMETERS
408
GENERATE MATCHING DATA FOR CANDIDATE FUNCTIONS
410
SELECT ONE OR MORE CANDIDATE FUNCTIONS NOT MEETING ACCEPTABILITY CRITERIA
412
SPLIT EACH OF THE ONE OR MORE SELECTED CANDIDATE FUNCTIONS INTO COMPONENTS; RECOMBINE THE COMPONENTS INTO NEW CANDIDATE FUNCTIONS
414
IDENTIFY ONE OR MORE CANDIDATE FUNCTIONS THAT MEET ACCEPTABILITY CRITERIA, OR ALTERNATIVELY SPLIT AND RECOMBINE CANDIDATE FUNCTIONS UNTIL ACCEPTABILITY CRITERIA IS SATISFIED
416
GENERATE RESULTS DATA INDICATING ONE OR MORE ACCEPTABLE CANDIDATE FUNCTIONS
418
OUTPUT THE RESULTS DATA
420
END
422

FIG. 4

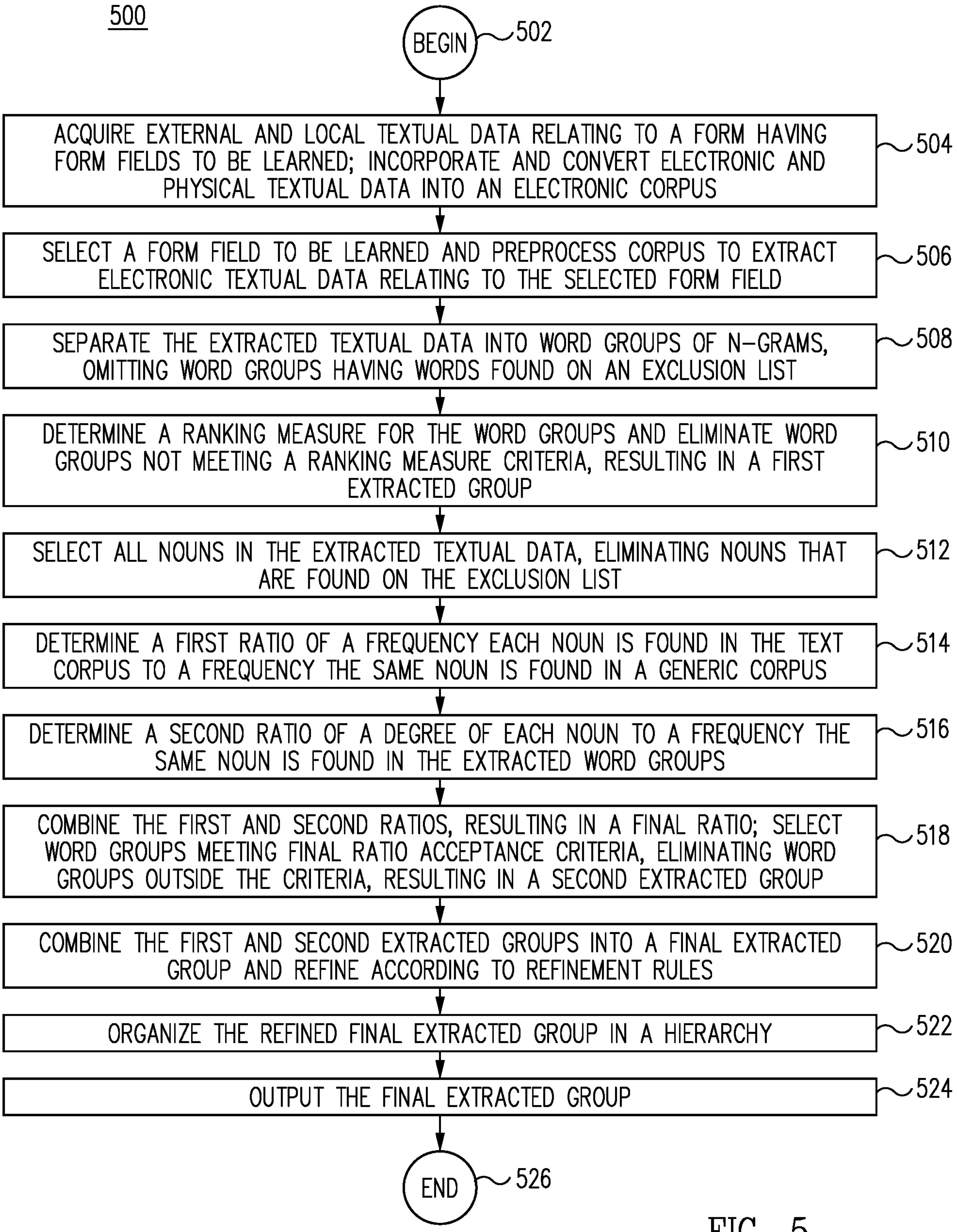
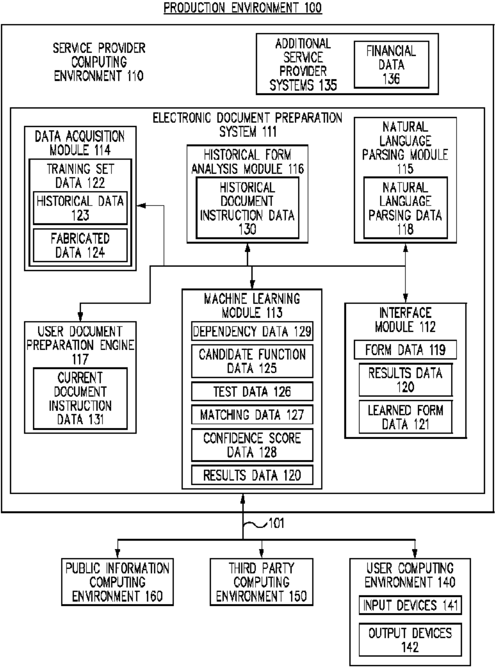

500
BEGIN
502
ACQUIRE EXTERNAL AND LOCAL TEXTUAL DATA RELATING TO A FORM HAVING FORM FIELDS TO BE LEARNED; INCORPORATE AND CONVERT ELECTRONIC AND PHYSICAL TEXTUAL DATA INTO AN ELECTRONIC CORPUS
504
SELECT A FORM FIELD TO BE LEARNED AND PREPROCESS CORPUS TO EXTRACT ELECTRONIC TEXTUAL DATA RELATING TO THE SELECTED FORM FIELD
506
SEPARATE THE EXTRACTED TEXTUAL DATA INTO WORD GROUPS OF N-GRAMS, OMITTING WORD GROUPS HAVING WORDS FOUND ON AN EXCLUSION LIST
508
DETERMINE A RANKING MEASURE FOR THE WORD GROUPS AND ELIMINATE WORD GROUPS NOT MEETING A RANKING MEASURE CRITERIA, RESULTING IN A FIRST EXTRACTED GROUP
510
SELECT ALL NOUNS IN THE EXTRACTED TEXTUAL DATA, ELIMINATING NOUNS THAT ARE FOUND ON THE EXCLUSION LIST
512
DETERMINE A FIRST RATIO OF A FREQUENCY EACH NOUN IS FOUND IN THE TEXT CORPUS TO A FREQUENCY THE SAME NOUN IS FOUND IN A GENERIC CORPUS
514
DETERMINE A SECOND RATIO OF A DEGREE OF EACH NOUN TO A FREQUENCY THE SAME NOUN IS FOUND IN THE EXTRACTED WORD GROUPS
516
COMBINE THE FIRST AND SECOND RATIOS, RESULTING IN A FINAL RATIO; SELECT WORD GROUPS MEETING FINAL RATIO ACCEPTANCE CRITERIA, ELIMINATING WORD GROUPS OUTSIDE THE CRITERIA, RESULTING IN A SECOND EXTRACTED GROUP
518
COMBINE THE FIRST AND SECOND EXTRACTED GROUPS INTO A FINAL EXTRACTED GROUP AND REFINE ACCORDING TO REFINEMENT RULES
520
ORGANIZE THE REFINED FINAL EXTRACTED GROUP IN A HIERARCHY
522
OUTPUT THE FINAL EXTRACTED GROUP
524
END
526

FIG. 5


PRODUCTION ENVIRONMENT 100
SERVICE PROVIDER COMPUTING ENVIRONMENT 110
ADDITIONAL SERVICE PROVIDER SYSTEMS 135
FINANCIAL DATA 136
ELECTRONIC DOCUMENT PREPARATION SYSTEM 111
DATA ACQUISITION MODULE 114
TRAINING SET DATA 122
HISTORICAL DATA 123
FABRICATED DATA 124
HISTORICAL FORM ANALYSIS MODULE 116
HISTORICAL DOCUMENT INSTRUCTION DATA 130
NATURAL LANGUAGE PARSING MODULE 115
NATURAL LANGUAGE PARSING DATA 118
USER DOCUMENT PREPARATION ENGINE 117
CURRENT DOCUMENT INSTRUCTION DATA 131
MACHINE LEARNING MODULE 113
DEPENDENCY DATA 129
CANDIDATE FUNCTION DATA 125
TEST DATA 126
MATCHING DATA 127
CONFIDENCE SCORE DATA 128
RESULTS DATA 120
INTERFACE MODULE 112
FORM DATA 119
RESULTS DATA 120
LEARNED FORM DATA 121
101
PUBLIC INFORMATION COMPUTING ENVIRONMENT 160
THIRD PARTY COMPUTING ENVIRONMENT 150
USER COMPUTING ENVIRONMENT 140
INPUT DEVICES 141
OUTPUT DEVICES 142