



(51) International Patent Classification:
G06F 17/27 (2006.01)

(21) International Application Number:
PCT/CN2018/123351

(22) International Filing Date:
25 December 2018 (25.12.2018)

(25) Filing Language: English

(26) Publication Language: English

(71) Applicant: **MICROSOFT TECHNOLOGY LICENSING, LLC** [US/US]; ONE MICROSOFT WAY, Redmond, Washington 98052 (US).

(72) Inventors: **WANG, Ying**; No. 999, Zi Xing Road, Min Hang District, Shanghai 200241 (CN). **LI, Min**; No. 999, Zi Xing Road, Min Hang District, Shanghai 200241 (CN). **LU, Mengyan**; No. 999, Zi Xing Road, Min Hang District, Shanghai 200241 (CN).

(74) Agent: **SHANGHAI PATENT & TRADEMARK LAW OFFICE, LLC**; 435 Guiping Road, Shanghai 200233 (CN).

(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).

Published:

— with international search report (Art. 21(3))

(54) Title: TECHNICAL DOCUMENT ISSUES SCANNER

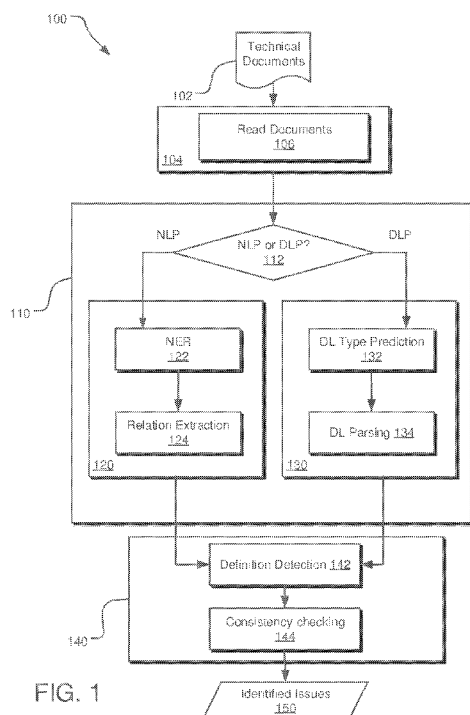


FIG. 1

(57) Abstract: A technical document scanner disclosed herein determines and categorizes various common issues among a large number of documents. An implementation of the technical document scanner is implemented using various computer process instructions including scanning a technical document to extract content, applying named entity recognition on the extracted content from the technical document to extract named entities, applying relation extraction on the named entities to extract relations between the named entities, and analyzing the relations between the entities to compose lists of high relevance entities for issue checking.

TECHNICAL DOCUMENT ISSUES SCANNER

Background

[0001] Technology companies publish documents online and that allow users to understand various products, such as databases, computer programming interfaces, word processing software, hardware products, network protocols, API documents, etc. Inspection of such technical document is an important procedure before publication. However, manually checking the documents for accuracy and completeness is time-consuming and tedious, especially when the documents have over hundreds of pages. Moreover, some issues (e.g. inconsistent or missing information, cross reference, etc.) are hard to be captured by human reviewers.

Summary

[0002] Implementations described herein discloses a technical document scanner determines and categorizes various common issues among a large number of documents. An implementation of the technical document scanner is implemented using various computer process instructions including scanning a technical document to extract content, applying named entity recognition on the extracted content to extract named entities from the technical document, applying relation extraction on the extracted entities to extract relations between the entities, and analyzing the relations between the named entities to compose lists of high relevance entities for issue checking.

[0003] This Summary is provided to introduce a selection of concepts in a simplified form that are further described below in the Detailed Description. This Summary is not

intended to identify key features or essential features of the claimed subject matter, nor is it intended to be used to limit the scope of the claimed subject matter.

[0004] Other implementations are also described and recited herein.

Brief Descriptions of the Drawings

[0005] A further understanding of the nature and advantages of the present technology may be realized by reference to the figures, which are described in the remaining portion of the specification.

[0006] FIG. 1 illustrates an example implementation of a system for technical document issues (TDI) scanner.

[0007] FIG. 2 illustrates an example implementation of natural language processing (NLP) operations used by the TDI scanner disclosed herein.

[0008] FIG. 3 illustrates example implementation of a relation extraction model of the TDI scanner disclosed herein.

[0009] FIG. 4 illustrates example operations for description language classification using machine learning (ML) according to implementations disclosed herein.

[0010] FIG. 5 illustrates example operations for definition detection using ML according to implementations disclosed herein.

[0011] FIG. 6 illustrates an example computing system that may be useful in implementing the described technology.

Detailed Descriptions

[0012] Implementations disclosed here provide a technical document scanner. The technical document scanner may use natural language processing (NLP) and machine learning (ML) approaches to scan the documents to categorize various common issues among

a large number of documents. For example, an implementation of the technology may use named entity recognition (NER) and relation extraction NLP processes to extract relations between named entities and to analyze the relations between the named entities. For example, some of the common technical document issues (TDIs) may include missing definition, inconsistent naming, wrong reference, inconsistent values, conflicting descriptions, etc.

[0013] The technology disclosed herein solves a technical problem of identifying issues in technical documents using technological solutions that include use of machine learning models. Specifically, the technology disclosed herein uses a NER ML model and a relation extraction model that includes a long short-term memory (LSTM) ML model. In one implementation, the LSTM model includes representation of one or more of named entities using bidirectional LSTM- recursive neural networks (RNNs). An implementation of the ML model includes a feature extraction operation using term frequency-inverse term frequency (TF-IDF) on unigrams scanned from the technical document and a classifier training operation using a support vector machine (SVM) classifier to classify the extracted features.

[0014] The technical document scanner disclosed herein may use ML models such as supervised learning models such as a support vector machine (SVM) model, deep learning LSTM models, or other deep learning models. The technology disclosed herein does not rely on hard coded validation rules by extracting information with self-defined named entities and their relations using NLP and ML methodologies. As the coded rules based approach is hard to maintain and it can only check hard-coded problems, the technical document scanner disclosed herein provides a better solution.

[0015] While the technology disclosed herein is disclosed in view of scanning and analyzing technical documents, it may also be used to scan and analyze other types of documents as well. For example, an alternative implementation of the document scanner disclosed herein may be used to scan and analyze legal documents, medical documents,

contracts, product descriptions, etc. Thus, the technology disclosed herein may be used by researchers/engineers in other communities.

[0016] Furthermore, the technology disclosed herein may also assist human reviewers of documents and in contrast with laborious and expensive manual inspection approaches, the technology disclosed herein provides advantage in both document checking efficiency and accuracy. Furthermore, the document scanner technology disclosed herein is an extendable solution in that over time its performance can be improved by training better ML models. In one implementation, the document scanner technology disclosed herein may be deployed on a cloud environment.

[0017] An implementation of the TDI scanner system disclosed herein is a collect-and-feedback system that operates by imitating a human being who has the background knowledge of the technical documents.

[0018] FIG. 1 illustrates an example implementation of a system for technical document issues (TDI) scanner 100. The TDI scanner 100 may be implemented on a computing device such as the computing device such as a laptop, a desktop, a server, or a mobile computing device. An example of a computing device and its components are further disclosed in FIG. 6 below. In one implementation, each component of the TDI scanner 100 may implemented on a separate computing device on a cloud.

[0019] The implementation of the TDI scanner 100 may be understood to be divided in three modules as disclosed in FIG. 1. Specifically, these three modules are a reader module 104, an information scanning module 110, and a checking module 140. The reader module 104 ingests technical documents 102, reads the technical documents 102, and stores the content in self-defined structures.

[0020] The information scanning module 110 extracts information from the stored content by reader module 104, in both natural languages and description languages.

Specifically, the information scanning module 110 uses NLP models for scanning natural language content such as description of structures, implementation details, etc. In one implementation, the NLP models extract important information from technical documents which will be used for completeness and consistency checking. On the other hand, the information scanning module 110 uses description language processing (DLP) models for scanning description language content, such as code.

[0021] Examples of the NLP models may include a named entity recognition (NER) model 122 that is configured to retrieve the entities of interest which represent information (e.g. size, type, etc.) and a relation extraction (RE) model (124) to associate the retrieved entities with the ones which represent object definitions or object references (e.g. field definition, structure reference, etc.).

[0022] Examples of the DLP models may include a description language (DL) type prediction module 132 that may be implemented using a support vector machine (SVM) classifier to predict the type of DL and a parsing module 134 to parse the content with regular expressions according to the type of DL. In one implementation, all the objects with the associated information from both NL and DL are inserted to either of a definition list or a reference list. An example of a definition list may include FieldDefName1, StructureDefName1, etc., while an example of a reference list may include FieldRefName1, FieldRefName2, StructureRefName1, etc.

[0023] The checking module 140 may include a definition detection module 142 that is implemented using an SVM classifier to locate the definition from the definition list for each entity in the reference list output by the information scanning module 110. A consistency checking module 144 may compare the extracted information contained by referred entities with related definitions for consistency check and generate identified issues 144. Various module of the TDI scanner 100 are disclosed in further detail below in FIGS. 2-5. In one

implementation, the checking module 140 analyzes the relations between the entities to compose lists of high relevance entities for issue checking. Such analysis may include inserting an entity into one of the lists to find related entities based on entity relations to compose a record for that list. Subsequently, information between various lists is compared to according to the named entity.

[0024] FIG. 2 illustrates an example implementation of natural language processing (NLP) operations 200 used by the TDI scanner disclosed herein. Specifically, the NLP operations 200 illustrate extracting information from a document 202. In the illustrated implementation, the document 202 includes the following content:

“Hdr (4 bytes): A TS_RAIL_PDU_HEADER structure.” Content A

[0025] Such content may be from a technical document such as a blog, user manual, online instructions, a protocol specification document, etc. An operation 204 tokenizes the content to generate a row of a content table 208. Specifically, the tokenizer breaks down each part of the content A in tokens 0 to 9.

[0026] Subsequently, an NER operation 206 categorizes the tokens 0 to 9 into various entities. For example, the token 0, “Hdr,” is categorized as *FieldDef*, whereas the token 7, “TS_RAIL_PDU_HEADER,” is categorized in the *StructureRef* category. For technical documents, the important entities could be field name, structure name, size, type, etc. In one implementation, an NER model used by operation 206 may be trained using a generally available named entity recognizer model such as the Stanford NER model. For example, the following seventeen (17) customized named entity labels may be used by the NER model:

StructureDef, StructureRef, FieldDef, FieldRef, TypeBasic, TypeModifier, FieldModifier, EnumOrFlag, Value, ValueModifierLevel, ValueModifierRestriction, Size, CollectionLength, SectionName, SectionNum, ReferredDoc, OperationRef, “O” (others).

[0027] In one implementation, the NER operation 206 may use a Conditional Random Field (CRF) sequence model.

[0028] A relation extraction operation 210 extracts the relations between various tokens 0 to 9 to generate the extracted information 220. Specifically, the relation extraction operation 210 retrieve the relation between entities recognized in NER operation 206 so that the information can be associated to the corresponding objects. In one implementation, an ML classifier may be used to predict relations between two entities. The relation extraction operation 210 is described in further detail in FIG. 3 below. Specifically, the extracted information 220 suggests that content A provides a definition as follows:

“Hdr, Type: TS_RAIL_PDU_HEADER, Size: 4 bytes”

[0029] FIG. 3 illustrates example implementation of a relation extraction model 300 of the TDI scanner disclosed herein. In particular, the relation extraction model 300 uses Long short-term memory (LSTM) layer that is capable of exploiting longer range of temporal dependencies in the sequences and avoiding gradient vanishing or exploding. the relation extraction model 300 consists of three layers, an input layer 302, an LSTM layer 304, and an output layer 306. The input layer 302 generates representation of each named entities, such as FieldDef, Size, etc., received from previous operations. The LSTM layer represents the named entity sequence of the sentence with bidirectional LSTM-recursive neural networks (RNNs). Specifically, each of the LSTM units at time step t receives the named entity embedding as input vector x_t , the previous hidden state h_{t-1} , the memory cell vector c_{t-1} , and produces the new vectors using the following equations:

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i)$$

$$f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f)$$

$$c_t = f_t c_{t-1} + i_t \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c)$$

$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_t + b_o)$$

$$h_t = o_t \tanh(c_t)$$

[0030] where σ denotes the logistic function, i , f , o , c and h are respectively the input gate, forget gate, output gate, cell activation vectors, and hidden state vector. W are weight matrices and b are bias vectors.

[0031] The output layer 306 employs a hidden layer and a softmax output layer to get the relation labels using the following equations:

$$h_t^{(r)} = \tanh(W_{rh}[y_{t-1}; h_t] + b_{rh})$$

$$y_t = \text{softmax}(W_{ry}h_t^{(r)} + b_y)$$

[0032] where, b and h are respectively the weight matrices, bias vectors, and hidden states. The output layer 306 outputs a relation label sequence that represents the relations between a current entity and a first named entity. For example, a named entity sequence may include entities A, B, C, and D and an output of relation sequence may include relations E, F, G, and H, where E represents a relation of entity A with itself, F represents a relation between the entity A and the entity B, G represents a relation between the entity A and the entity C, H represents a relation between the entity A and the entity D, etc.

[0033] As the relation extraction model 300 extracts relations between the first named entity in input and a current entity, in one implementation, the named entities are removed from the start so as to predict several relations with different inputs to get all the relations in an input sentence. As one can extract a relation between an input named entity and another entity from the relation extraction model 300, the relation is predicted several times with different input of named entity to extract all relations in a sentence. In one implementation, no relations existed between "O" (others) and other named entities so the relation extraction model 300 ignores entities that are tagged with "O."

[0034] For example, if the named entity input sequence is *[FieldDef, O, Size, O, Size, TypeModifier, TypeBasic, O]*, the relation extraction model 300 needs to predict the relations with following four input sequences:

[FieldDef, O, Size, O, Size, TypeModifier, TypeBasic],

[Size, O, Size, TypeModifier, TypeBasic],

[Size, TypeModifier, TypeBasic],

[TypeModifier, TypeBasic]

[0035] to get all the relations in the input sentence. Below is an example of relation extraction by the relation extraction model 300 from an input sentence:

[0036] Sentence:

length (2 bytes): A 16-bit, unsigned integer that specifies the packet size. This field MUST be set to 0x0008 (8 bytes).

[0037] Relation(s) extracted:

Field_Size: length, 2 bytes

Field_Size: length, 16-bit

Field_Type: length, unsigned integer

Field_Value: length, 0x0008

[0038] As a result, the following record is inserted in the definition list:

length, [Size: 2 bytes], [Size: 16-bit], [Type: unsigned integer], [Value: 0x0008]

[0039] FIG. 4 illustrates example operations 400 for description language (DL) processing (DL) using machine learning (ML) according to implementations disclosed herein. The DL processing operations 400 may include DL type prediction and DL parsing. For example, the operations 400 predicts the type of DL 404 using an ML model 410 that may be trained on training data 402. The feature extraction module 406 of the ML model may use

term frequency – inverse term frequency (TF-IDF) on unigrams scanned from the DL 404 to identify features from the DL 404.

[0040] Furthermore, the feature extraction module 406 also extracts features with conjunctions of characters, such as [], [{ }], [STRING, ..., etc. A classifier training module 408 using SVM allows generating prediction 420 of the type of the DL 404. In one implementation, the classifier training module 408 may be implemented using a library of SVM (LibSVM), however, other ML classifier models may also be used. For example, the DL classification operations 400 predicts that the type of DL 404 is JSON.

[0041] FIG. 5 illustrates alternative example operations 500 for definition detection using ML according to implementations disclosed herein. The operations 500 may use an ML model 510 with a feature extraction module 514 and a classifier training module 516. A list of candidate definition items 504 is selected from a set of definition items 502. In one example, an edit-distance algorithm may be used to generate the candidate definition items 504 the definition items 502. The edit-distance algorithm may include the following considerations:

[0042] 550: Split compound names according to case change or symbols. For example, “AutoDetectCliRequestPdu” is split as “Auto”, “Detect”, “Cli”, “Request”, “Pdu” whereas “TS_RAIL_PDU_Header” is split as “TS”, “RAIL”, “PDU”, “Header.” Subsequently, calculate distance based on segments.

[0043] 552: Consider the cost differences due to case. For example, a distance from “TS_RAIL_PDU_Header” to “ts_rail_pdu_header” may have a lower value than ten(10) characters.”

[0044] 554: Use add or delete operations, for example, the distance between “AutoDetectCliRequestPDU” and “AutoDetectCliReqPDU” will be lower than “AutoDetectCliRequestPDU” and “AutoDetectSrvRequestPDU.”

[0045] A set of definition items 506 is generated from the candidate definition items 504. The definition items 506 are input to the ML model 510 together with reference items 508. The feature extraction module 514 may use similarities between the definition items 506 and the reference items 508 to extract the features from the definition items. The ML model 510 generates a prediction 520 and a result selection module 522 selects the results of the prediction 520 to find the definition item 524.

[0046] FIG. 6 illustrates an example system 600 that may be useful in implementing the described technology for providing attestable and destructible device identity. The example hardware and operating environment of FIG. 6 for implementing the described technology includes a computing device, such as a general-purpose computing device in the form of a computer 20, a mobile telephone, a personal data assistant (PDA), a tablet, smart watch, gaming remote, or other type of computing device. In the implementation of FIG. 6, for example, the computer 20 includes a processing unit 21, a system memory 22, and a system bus 23 that operatively couples various system components including the system memory to the processing unit 21. There may be only one or there may be more than one processing unit 21, such that the processor of the computer 20 comprises a single central-processing unit (CPU), or a plurality of processing units, commonly referred to as a parallel processing environment. The computer 20 may be a conventional computer, a distributed computer, or any other type of computer; the implementations are not so limited.

[0047] The system bus 23 may be any of several types of bus structures including a memory bus or memory controller, a peripheral bus, a switched fabric, point-to-point connections, and a local bus using any of a variety of bus architectures. The system memory may also be referred to as simply the memory, and includes read only memory (ROM) 24 and random access memory (RAM) 25. A basic input/output system (BIOS) 26, containing the basic routines that help to transfer information between elements within the computer 20,

such as during start-up, is stored in ROM 24. The computer 20 further includes a hard disk drive 27 for reading from and writing to a hard disk, not shown, a magnetic disk drive 28 for reading from or writing to a removable magnetic disk 29, and an optical disk drive 30 for reading from or writing to a removable optical disk 31 such as a CD ROM, DVD, or other optical media.

[0048] The hard disk drive 27, magnetic disk drive 28, and optical disk drive 30 are connected to the system bus 23 by a hard disk drive interface 32, a magnetic disk drive interface 33, and an optical disk drive interface 34, respectively. The drives and their associated tangible computer-readable media provide non-volatile storage of computer-readable instructions, data structures, program modules and other data for the computer 20. It should be appreciated by those skilled in the art that any type of tangible computer-readable media may be used in the example operating environment.

[0049] A number of program modules may be stored on the hard disk drive 27, magnetic disk 28, optical disk 30, ROM 24, or RAM 25, including an operating system 35, one or more application programs 36, other program modules 37, and program data 38. A user may generate reminders on the personal computer 20 through input devices such as a keyboard 40 and pointing device 42. Other input devices (not shown) may include a microphone (e.g., for voice input), a camera (e.g., for a natural user interface (NUI)), a joystick, a game pad, a satellite dish, a scanner, or the like. These and other input devices are often connected to the processing unit 21 through a serial port interface 46 that is coupled to the system bus 23, but may be connected by other interfaces, such as a parallel port, game port, or a universal serial bus (USB) (not shown). A monitor 47 or other type of display device is also connected to the system bus 23 via an interface, such as a video adapter 48. In addition to the monitor, computers typically include other peripheral output devices (not shown), such as speakers and printers.

[0050] The computer 20 may operate in a networked environment using logical connections to one or more remote computers, such as remote computer 49. These logical connections are achieved by a communication device coupled to or a part of the computer 20; the implementations are not limited to a particular type of communications device. The remote computer 49 may be another computer, a server, a router, a network PC, a client, a peer device or other common network node, and typically includes many or all of the elements described above relative to the computer 20. The logical connections depicted in FIG. 10 include a local-area network (LAN) 51 and a wide-area network (WAN) 52. Such networking environments are commonplace in office networks, enterprise-wide computer networks, intranets and the Internet, which are all types of networks.

[0051] When used in a LAN-networking environment, the computer 20 is connected to the local network 51 through a network interface or adapter 53, which is one type of communications device. When used in a WAN-networking environment, the computer 20 typically includes a modem 54, a network adapter, a type of communications device, or any other type of communications device for establishing communications over the wide area network 52. The modem 54, which may be internal or external, is connected to the system bus 23 via the serial port interface 46. In a networked environment, program engines depicted relative to the personal computer 20, or portions thereof, may be stored in the remote memory storage device. It is appreciated that the network connections shown are examples and other means of communications devices for establishing a communications link between the computers may be used.

[0052] In an example implementation, software or firmware instructions for providing attestable and destructible device identity may be stored in memory 22 and/or storage devices 29 or 31 and processed by the processing unit 21. One or more ML, NLP, or DLP models disclosed herein may be stored in memory 22 and/or storage devices 29 or 31 as

persistent datastores. For example, a TDI scanner 602 may be implemented on the computer 20 (alternatively, the TDI scanner 602 may be implemented on a server or in a cloud environment). The TDI scanner 602 may utilize one or more of the processing unit 21, the memory 22, the system bus 23, and other components of the personal computer 20.

[0053] In contrast to tangible computer-readable storage media, intangible computer-readable communication signals may embody computer readable instructions, data structures, program modules or other data resident in a modulated data signal, such as a carrier wave or other signal transport mechanism. The term "modulated data signal" means a signal that has one or more of its characteristics set or changed in such a manner as to encode information in the signal. By way of example, and not limitation, intangible communication signals include wired media such as a wired network or direct-wired connection, and wireless media such as acoustic, RF, infrared and other wireless media.

[0054] The implementations described herein are implemented as logical steps in one or more computer systems. The logical operations may be implemented (1) as a sequence of processor-implemented steps executing in one or more computer systems and (2) as interconnected machine or circuit modules within one or more computer systems. The implementation is a matter of choice, dependent on the performance requirements of the computer system being utilized. Accordingly, the logical operations making up the implementations described herein are referred to variously as operations, steps, objects, or modules. Furthermore, it should be understood that logical operations may be performed in any order, unless explicitly claimed otherwise or a specific order is inherently necessitated by the claim language.

[0055] The above specification, examples, and data provide a complete description of the structure and use of exemplary embodiments of the invention. Since many implementations of the invention can be made without departing from the spirit and scope of

the invention, the invention resides in the claims hereinafter appended. Furthermore, structural features of the different embodiments may be combined in yet another implementation without departing from the recited claims.

Claims

WHAT IS CLAIMED IS:

1. A physical article of manufacture including one or more tangible computer-readable storage media, encoding computer-executable instructions for executing on a computer system a computer process to provide attestable and destructible device identity, the computer process comprising:
 - scanning a document to extract content;
 - applying a named entity recognition (NER) model on the extracted content to extract named entities;
 - extracting relations between the named entities by applying a relation extraction model on the named entities; and
 - analyzing the relations between the named entities to compose lists of high relevance categories for issue checking.
2. The physical article of manufacture of claim 1, wherein the computer process further comprising extracting definitions for each of the named entities from the document.
3. The physical article of manufacture of claim 1, wherein the computer process further comprising extracting definitions for each of the named entities from the document using a machine learning (ML) algorithm.
4. The physical article of manufacture of claim 1, wherein the computer process further comprising checking the consistency for the list of high relevance categories.

5. The physical article of manufacture of claim 1, wherein scanning a document to extract content further comprising tokenizing parts of the document.

6. The physical article of manufacture of claim 5, wherein the computer process further comprising tagging one or more of the token using the NER model.

7. The physical article of manufacture of claim 1, wherein the relation extraction model is a long short-term memory (LSTM) model.

8. The physical article of manufacture of claim 7, wherein the LSTM model further comprises an input layer including representation of one or more of the named entities.

9. The physical article of manufacture of claim 8, wherein LSTM model further comprises an LSTM layer including representation of one or more of the named entities using bidirectional LSTM- recursive neural networks (RNNs).

10. In a computing environment, a method performed at least in part on at least one processor, the method comprising:
- scanning a technical document to extract content;
 - applying a named entity recognition (NER) model on the extracted content to extract named entities;
 - extracting relations between the named entities by applying a relation extraction model on the named entities; and
 - analyzing the relations between the named entities to compose lists of high relevance categories for issue checking.
11. The method of claim 10, further comprising extracting definitions for each of the named entities from the technical document.
12. The method of claim 10, further comprising extracting definitions for each of the named entities from the technical document using a machine learning (ML) algorithm.
13. The method of claim 10, wherein scanning a technical document to extract content further comprising tokenizing parts of the technical document.
14. The method of claim 13, further comprising tagging one or more of the token using the NER model.
15. The method of claim 10, wherein the relation extraction model is a long short term memory (LSTM) model.
16. The method of claim 15, wherein the LSTM model further comprises:

an input layer including representation of one or more of the named entities; and

an LSTM layer including representation of one or more of the named entities using
bidirectional LSTM- recursive neural networks (RNNs)

17. In a computing environment, a system comprising:

- memory;
- one or more processor units;
- a document issue scanner system stored in the memory and executable by the one or more processor units, the document issue scanner system encoding computer-executable instructions on the memory for executing on the one or more processor units a computer process, the computer process comprising:
 - scanning a technical document to extract content;
 - applying a named entity recognition (NER) model on the extracted content to extract named entities;
 - extracting relations between the named entities by applying a relation extraction model on the named entities; and
 - analyzing the relations between the entities to compose lists of high relevance named entities for issue checking.

18. The system of claim 17, wherein the computer process further comprising extracting definitions for each of the named entities from the technical document using a machine learning (ML) algorithm.

19. The system of claim 18, wherein the ML algorithm comprises:

- a feature extraction operation using term frequency-inverse term frequency (TF-IDF) on unigrams scanned from the technical document; and
- a classifier training operation.

20. The system of claim 19, wherein the classifier training operation uses a support vector machine (SVM) classifier.

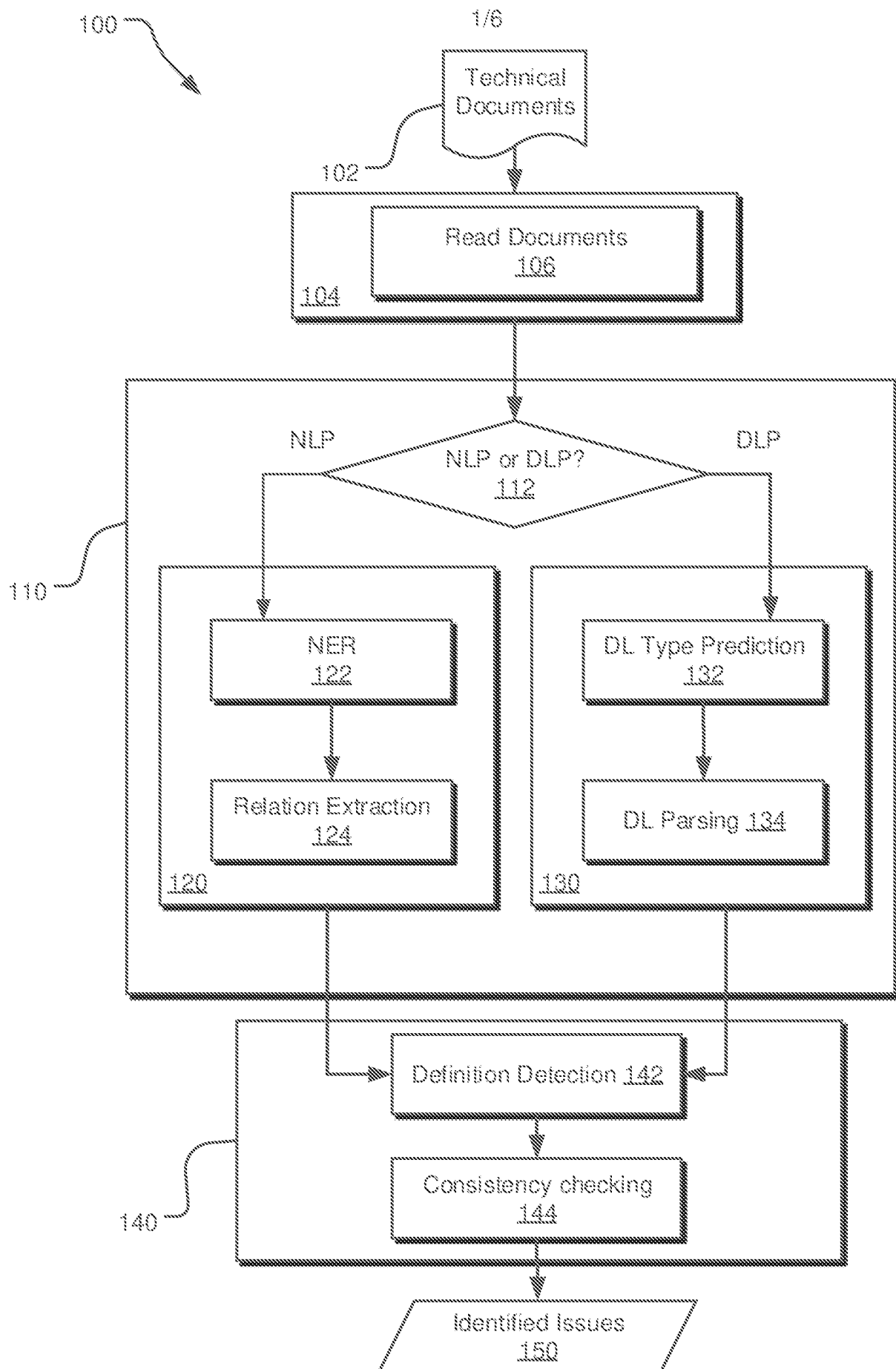


FIG. 1

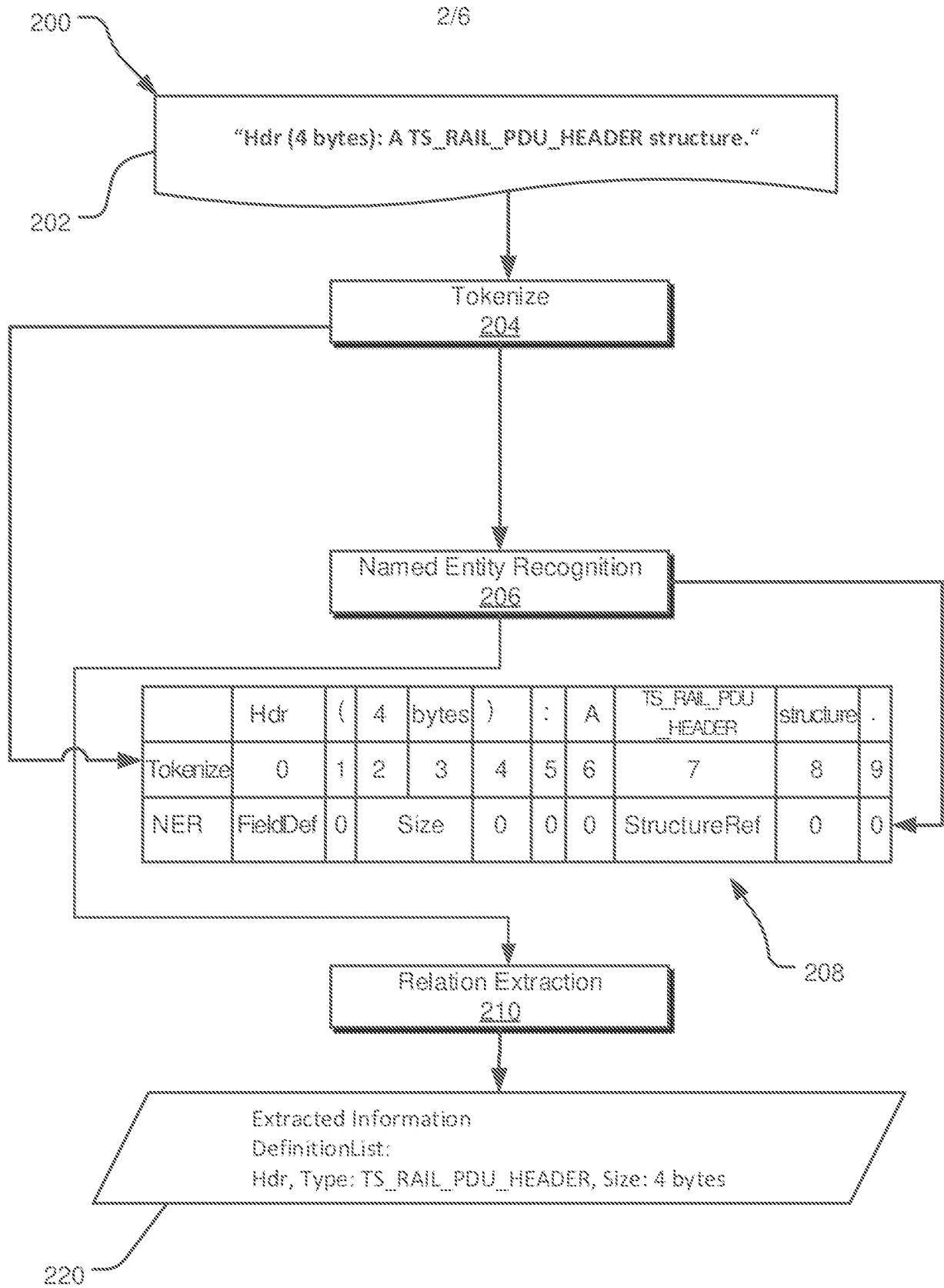


FIG. 2

300

3/6

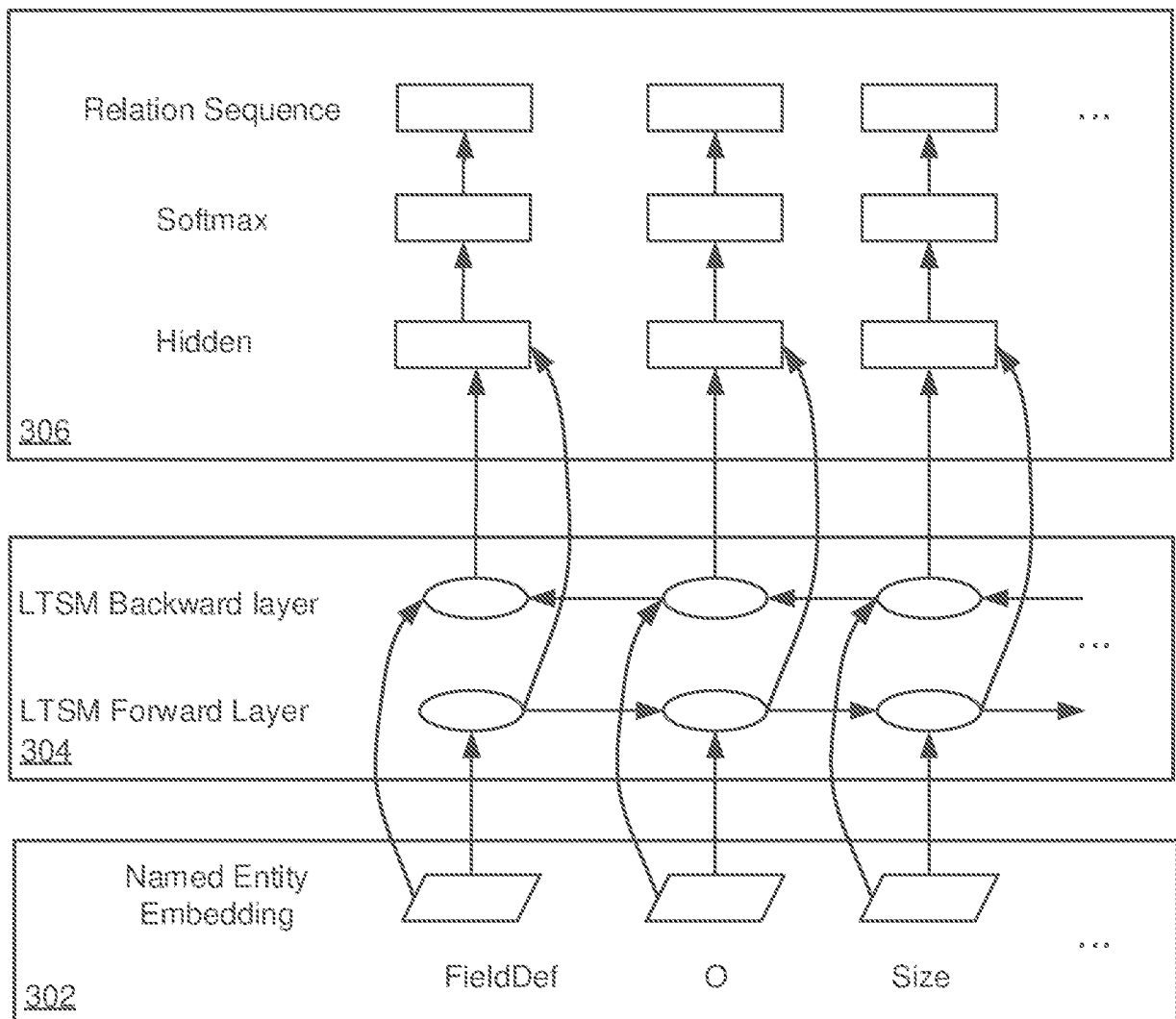


FIG. 3

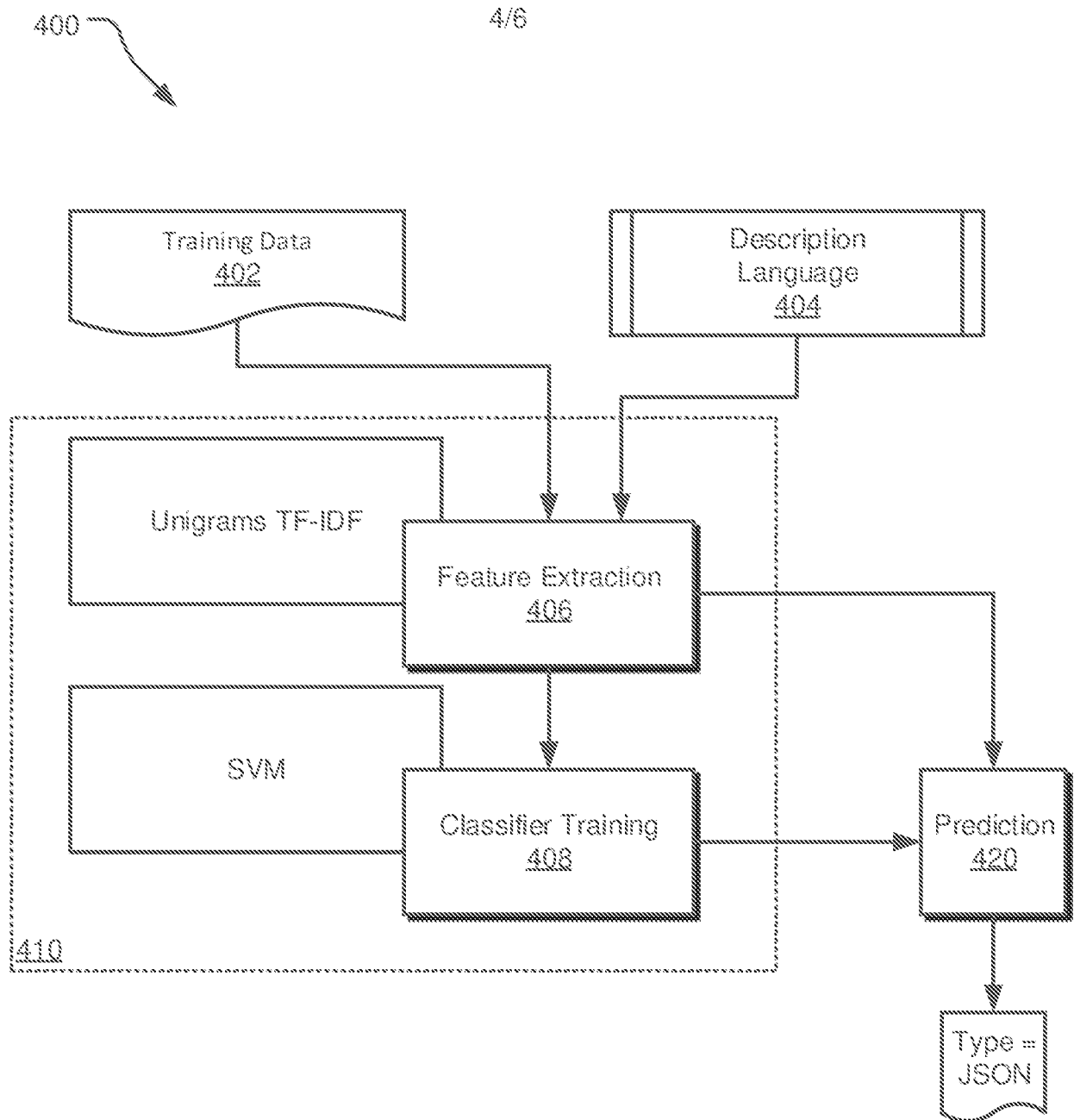


FIG. 4

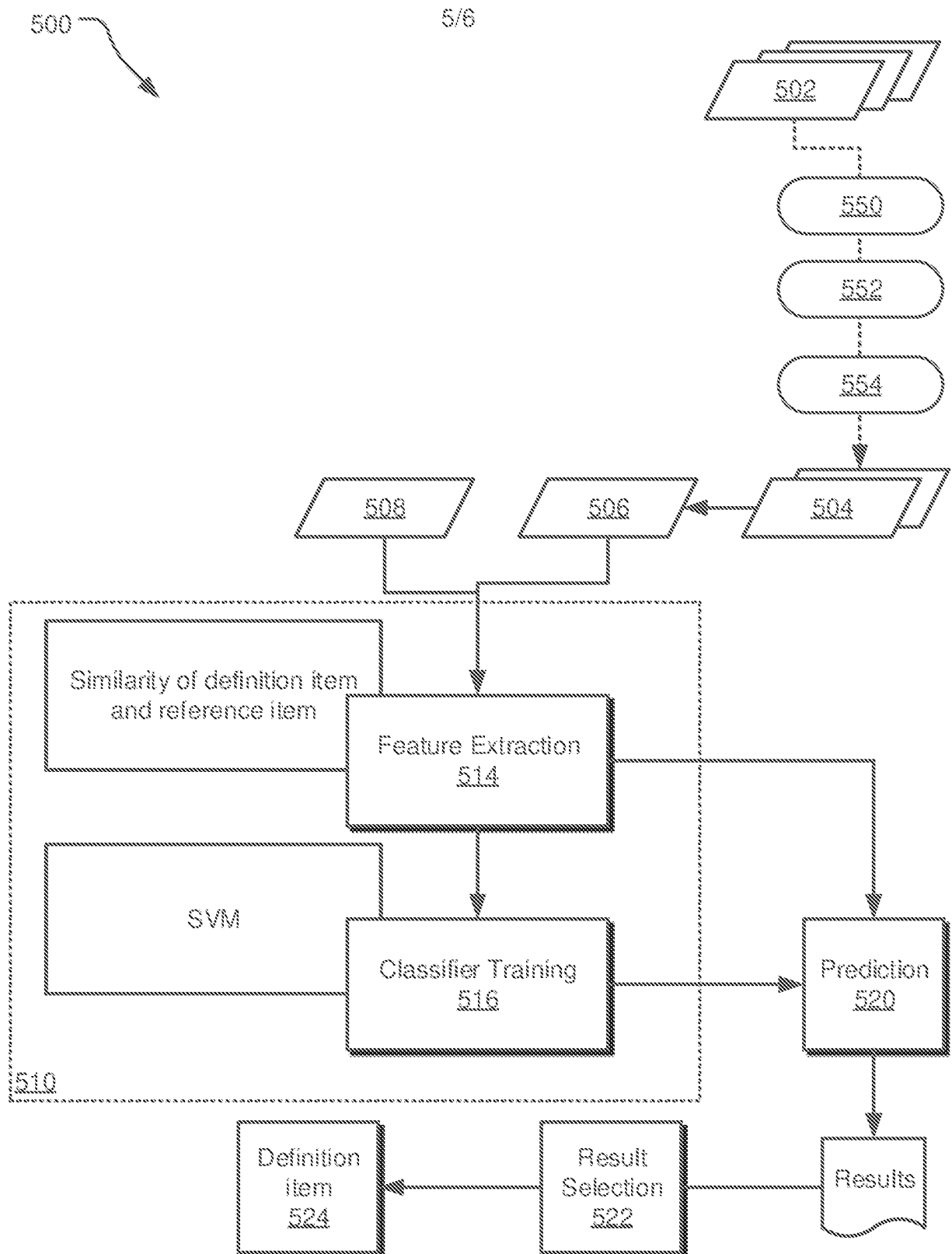


FIG. 5

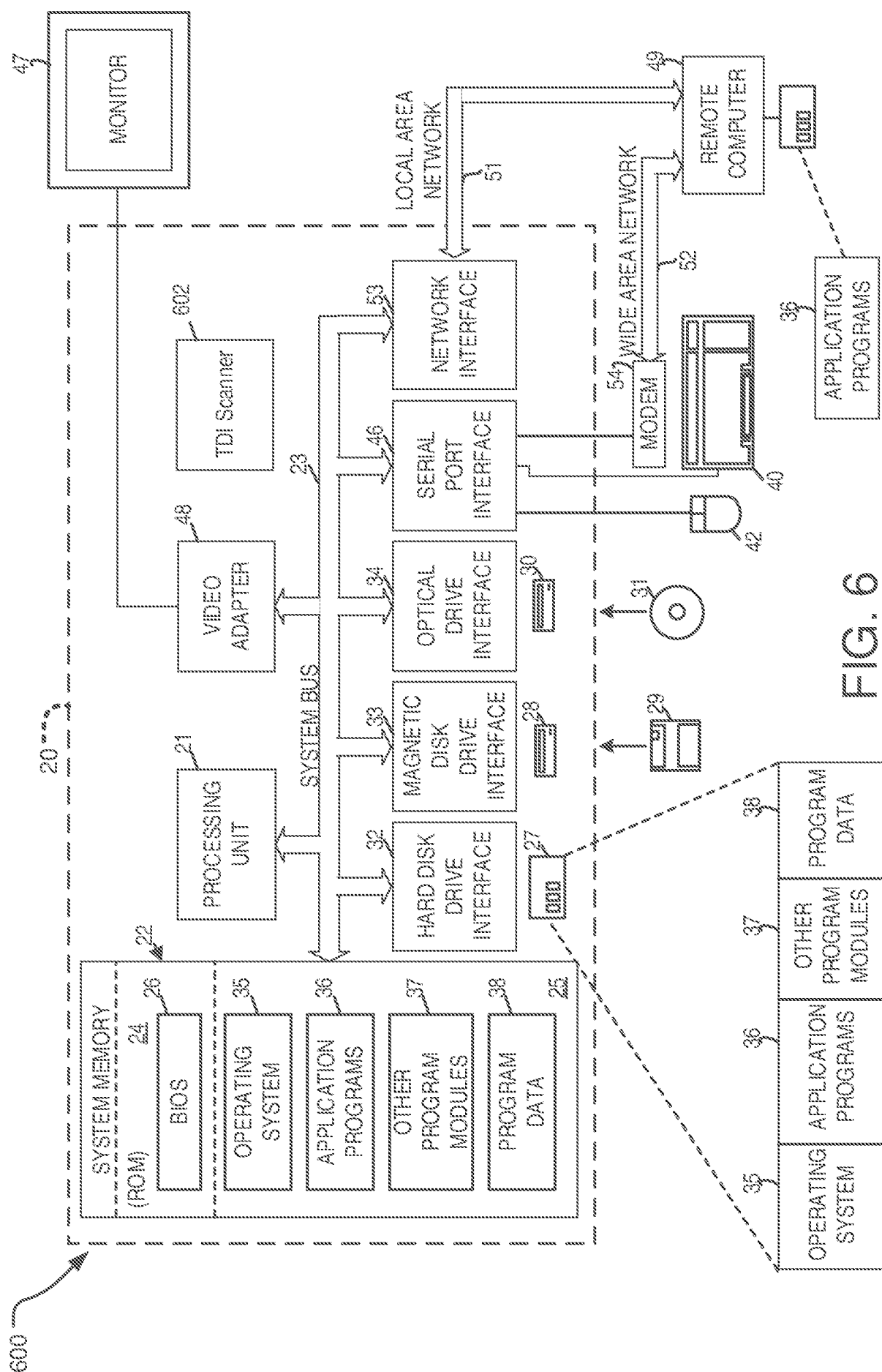


FIG. 6

INTERNATIONAL SEARCH REPORT

International application No.

PCT/CN2018/123351

A. CLASSIFICATION OF SUBJECT MATTER

G06F 17/27(2006.01)i

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

G06F; H04L

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

CNPAT;CNKI;WPI;EPODOC:scan+,extract+,named entity recognition,NER,named entity,NE,correlated entity,CE,relation,analy+,relavance,machine learning,ML

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	CN 105468605 A (JINAN ZHONGLIN INFORMATION TECHNOLOGY CO., LTD.) 06 April 2016 (2016-04-06) description, paragraphs[0057]-[0126]	1-20
A	CN 105512197 A (GUANGZHOU BAOSTEEL SOUTHERN TRADE CO., LTD.) 20 April 2016 (2016-04-20) the whole document	1-20
A	US 2017060835 A1 (XEROX CORPORATION) 02 March 2017 (2017-03-02) the whole document	1-20



Further documents are listed in the continuation of Box C.



See patent family annex.

* Special categories of cited documents:

“A” document defining the general state of the art which is not considered to be of particular relevance

“E” earlier application or patent but published on or after the international filing date

“L” document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

“O” document referring to an oral disclosure, use, exhibition or other means

“P” document published prior to the international filing date but later than the priority date claimed

“T” later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

“X” document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

“Y” document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

“&” document member of the same patent family

Date of the actual completion of the international search

09 August 2019

Date of mailing of the international search report

28 August 2019

Name and mailing address of the ISA/CN

National Intellectual Property Administration, PRC
6, Xitucheng Rd., Jimen Bridge, Haidian District, Beijing
100088
China

Authorized officer

CHEN,Xiyuan

Facsimile No. (86-10)62019451

Telephone No. 86-(10)-53961594

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.

PCT/CN2018/123351

Patent document cited in search report			Publication date (day/month/year)	Patent family member(s)			Publication date (day/month/year)
CN	105468605	A	06 April 2016	None			
CN	105512197	A	20 April 2016	None			
US	2017060835	A1	02 March 2017	EP	3136257	A2	01 March 2017
				JP	2017045453	A	02 March 2017