TEXT GENERATION FOR PATENT APPLICATIONS USING NATURAL LANGUAGE MODELING AND PROGRAMMABLE TEMPLATING LANGUAGE

[0001] A system may use a machine learning model to process input text and generate output text. For example, based on training a machine learning model on a particular corpus, such as a corpus of patent applications or sections thereof, the system may deploy the machine learning model for patent application drafting. In this case, the system may receive input text corresponding to a first portion of a patent application or information relating to a patent application and the system may use the machine learning model to generate a second portion of the patent application. However, such machine learning models may result in inconsistent output that fails to follow desired forms for the patent application. In another case, the system may use entity and semantic intent extraction techniques to extract entities and intents that can be used to generate portions of a patent application. However, such techniques may be limited in a level of semantic information that is extractable and usable. In yet another case, the system may use regular expressions to extract components of a corpus, such as components of a patent application, and generate new components for the patent application or other patent applications based on the extracted components. However, the use of regular expressions may limit a flexibility of an output from the system.

[0002] Some aspects described herein enable preparation of highly accurate output text in a patent application by using one or more natural language models coupled with a programmable templating language. For example, a system may perform natural language processing (NLP) using a plurality of component processes, such as a first natural language understanding (NLU) process and a second natural language generation (NLG) process. As a specific example, for

NLU, a system may use an NLP platform, such as a SpacyIO modularizable platform. The system may use a templating language (e.g., used for templating websites), such as Handlebars, and may embed logic in a generated template. Templating languages may be used for generating HTML, but may not be well suited to generated formatted patent document text. Accordingly, the system may deploy a set of helper functions and logic to translate the template text to a format useable in a word processing environment, such as in Microsoft Word. The system may extend a functionality of an underlying engine of the templating language with functions that manipulate the text based on a data model presented by the NLP platform. In this way, the system may generate precise text (e.g., more precisely output based on a given input text than is achieved with, for example, autoregressive language models, such as GPT-3 style free-form) that is more multi-dimensional than a template type output (e.g., in which words are filled into a static template). In some implementations, an output model of the system is stored in template space rather than in custom code, which enables a user of the system to directly manipulate the output of an NLG process of the system with all of the data provided by a natural language parser of the system.

[0003] Fig. 1 is a diagram of an example environment 100 in which systems and/or methods described herein may be implemented. Additional example environments, in which systems and/or methods described herein may be implemented, are described in the appendices. As shown in Fig. 1, environment 100 may include a computing system 101, which may include one or more elements of and/or may execute within a cloud computing system 102. The cloud computing system 102 may include one or more elements 103-113, as described in more detail below. As further shown in Fig. 1, environment 100 may include a network 120, a client device 130, a server device 140, and/or other devices described in the appendices. Devices and/or

elements of environment 100 may interconnect via wired connections and/or wireless connections.

[0004] The cloud computing system 102 includes computing hardware 103, a resource management component 104, a host operating system (OS) 105, and/or one or more virtual computing systems 106. The resource management component 104 may perform virtualization (e.g., abstraction) of computing hardware 103 to create the one or more virtual computing systems 106. Using virtualization, the resource management component 104 enables a single computing device (e.g., a computer, a server, and/or the like) to operate like multiple computing devices, such as by creating multiple isolated virtual computing systems 106 from computing hardware 103 of the single computing device. In this way, computing hardware 103 can operate more efficiently, with lower power consumption, higher reliability, higher availability, higher utilization, greater flexibility, and lower cost than using separate computing devices.

[0005] Computing hardware 103 includes hardware and corresponding resources from one or more computing devices. For example, computing hardware 103 may include hardware from a single computing device (e.g., a single server) or from multiple computing devices (e.g., multiple servers), such as multiple computing devices in one or more data centers. As shown, computing hardware 103 may include one or more processors 107, one or more memories 108, one or more storage components 109, and/or one or more networking components 110. Examples of a processor, a memory, a storage component, and a networking component (e.g., a communication component) are described elsewhere herein.

[0006] The resource management component 104 includes a virtualization application (e.g., executing on hardware, such as computing hardware 103) capable of virtualizing computing hardware 103 to start, stop, and/or manage one or more virtual computing systems 106. For

example, the resource management component 104 may include a hypervisor (e.g., a bare-metal or Type 1 hypervisor, a hosted or Type 2 hypervisor, and/or the like) or a virtual machine monitor, such as when the virtual computing systems 106 are virtual machines 111. Additionally, or alternatively, the resource management component 104 may include a container manager, such as when the virtual computing systems 106 are containers 112. In some implementations, the resource management component 104 executes within and/or in coordination with a host operating system 105.

[0007] A virtual computing system 106 includes a virtual environment that enables cloud-based execution of operations and/or processes described herein using computing hardware 103. As shown, a virtual computing system 106 may include a virtual machine 111, a container 112, a hybrid environment 113 that includes a virtual machine and a container, and/or the like. A virtual computing system 106 may execute one or more applications using a file system that includes binary files, software libraries, and/or other resources required to execute applications on a guest operating system (e.g., within the virtual computing system 106) or the host operating system 105.

[0008] Although the computing system 101 may include one or more elements 103-113 of the cloud computing system 102, may execute within the cloud computing system 102, and/or may be hosted within the cloud computing system 103, in some implementations, the computing system 101 may not be cloud-based (e.g., may be implemented outside of a cloud computing system) or may be partially cloud-based. For example, the computing system 101 may include one or more devices that are not part of the cloud computing system 102, such as device 200 of Fig. 2, which may include a standalone server or another type of computing device. The

computing system 101 may perform one or more operations and/or processes described in more detail elsewhere herein.

[0009] Network 120 includes one or more wired and/or wireless networks. For example, network 120 may include a cellular network, a public land mobile network (PLMN), a local area network (LAN), a wide area network (WAN), a private network, the Internet, and/or the like, and/or a combination of these or other types of networks. The network 120 enables communication among the devices of environment 100.

[0010] The client device 130 includes one or more devices capable of receiving, generating, storing, processing, and/or providing information described herein and/or in the appendices. The client device 130 may include a communication device and/or a computing device. For example, the client device 130 may include a wireless communication device, a user equipment (UE), a mobile phone (e.g., a smart phone or a cell phone, among other examples), a laptop computer, a tablet computer, a handheld computer, a desktop computer, a gaming device, a wearable communication device (e.g., a smart wristwatch or a pair of smart eyeglasses, among other examples), an Internet of Things (IoT) device, or a similar type of device. The client device 130 may communicate with one or more other devices of environment 100, as described elsewhere herein.

[0011] The server device 140 includes one or more devices capable of receiving, generating, storing, processing, providing, and/or routing information described herein and/or in the appendices. The server device 140 may include a communication device and/or a computing device. For example, the server device 140 may include a server, an application server, a client server, a web server, a database server, a host server, a proxy server, a virtual server (e.g., executing on computing hardware), a server in a cloud computing system, a device that includes

computing hardware used in a cloud computing environment, or a similar type of device. The server device 140 may communicate with one or more other devices of environment 100, as described elsewhere herein.

[0012] The number and arrangement of devices and networks shown in Fig. 1 are provided as an example. In practice, there may be additional devices and/or networks, fewer devices and/or networks, different devices and/or networks, or differently arranged devices and/or networks than those shown in Fig. 1. Furthermore, two or more devices shown in Fig. 1 may be implemented within a single device, or a single device shown in Fig. 1 may be implemented as multiple, distributed devices. Additionally, or alternatively, a set of devices (e.g., one or more devices) of environment 100 may perform one or more functions described as being performed by another set of devices of environment 100.

[0013] Fig. 2 is a diagram of example components of a device 200, which may correspond to computing system 101, client device 130, server device 140, and/or other devices described in the appendices. In some implementations, computing system 101, client device 130, server device 140, and/or other devices described in the appendices may include one or more devices 200 and/or one or more components of device 200. As shown in Fig. 2, device 200 may include a bus 210, a processor 220, a memory 230, a storage component 240, an input component 250, an output component 260, and a communication component 270.

[0014] Bus 210 includes a component that enables wired and/or wireless communication among the components of device 200. Processor 220 includes a central processing unit, a graphics processing unit, a microprocessor, a controller, a microcontroller, a digital signal processor, a field-programmable gate array, an application-specific integrated circuit, and/or another type of processing component. Processor 220 is implemented in hardware, firmware, or

a combination of hardware and software. In some implementations, processor 220 includes one or more processors capable of being programmed to perform a function. Memory 230 includes a random access memory), a read only memory, and/or another type of memory (e.g., a flash memory, a magnetic memory, and/or an optical memory).

[0015] Storage component 240 stores information and/or software related to the operation of device 200. For example, storage component 240 may include a hard disk drive, a magnetic disk drive, an optical disk drive, a solid state disk drive, a compact disc, a digital versatile disc, and/or another type of non-transitory computer-readable medium. Input component 250 enables device 200 to receive input, such as user input and/or sensed inputs. For example, input component 250 may include a touch screen, a keyboard, a keypad, a mouse, a button, a microphone, a switch, a sensor, a global positioning system component, an accelerometer, a gyroscope, an actuator, and/or the like. Output component 260 enables device 200 to provide output, such as via a display, a speaker, and/or one or more light-emitting diodes. Communication component 270 enables device 200 to communicate with other devices, such as via a wired connection and/or a wireless connection. For example, communication component 270 may include a receiver, a transmitter, a transceiver, a modem, a network interface card, an antenna, and/or the like.

[0016] Device 200 may perform one or more processes described herein. For example, a non-transitory computer-readable medium (e.g., memory 230 and/or storage component 240) may store a set of instructions (e.g., one or more instructions, code, software code, program code, and/or the like) for execution by processor 220. Processor 220 may execute the set of instructions to perform one or more processes described herein. In some implementations, execution of the set of instructions, by one or more processors 220, causes the one or more processors 220 and/or the device 200 to perform one or more processes described herein. In

some implementations, hardwired circuitry may be used instead of or in combination with the instructions to perform one or more processes described herein. Thus, implementations described herein are not limited to any specific combination of hardware circuitry and software.

[0017] The number and arrangement of components shown in Fig. 2 are provided as an example. Device 200 may include additional components, fewer components, different components, or differently arranged components than those shown in Fig. 2. Additionally, or alternatively, a set of components (e.g., one or more components) of device 200 may perform one or more functions described as being performed by another set of components of device 200.

[0018] Fig. 3 is a flowchart of an example process 300 associated with text generation for patent applications using natural language modeling and programmable templating language. In some implementations, one or more process blocks of Fig. 3 may be performed by a device (e.g., a device of cloud computing system 102). In some implementations, one or more process blocks of Fig. 3 may be performed by another device or a group of devices separate from or including the device. Additionally, or alternatively, one or more process blocks of Fig. 3 may be performed by one or more components of device 200, such as processor 220, memory 230, storage component 240, input component 250, output component 260, and/or communication component 270.

[0019] As shown in Fig. 3, process 300 may include identifying a text from a document includes structured information, wherein the structured information is a set of patent claims (block 310). For example, the device may identify a text from a document includes structured information, wherein the structured information is a set of patent claims, as described above.

[0020] As further shown in Fig. 3, process 300 may include extracting a first one or more parts and a second one or more parts, wherein the first one or more parts are one or more grammatical

parts and the second one or more parts are one or more legal parts (block 320). For example, the device may extract a first one or more parts and a second one or more parts, wherein the first one or more parts are one or more grammatical parts and the second one or more parts are one or more legal parts, as described above.

[0021] As further shown in Fig. 3, process 300 may include inserting the first one or more parts and the second one or more parts into a structured model data object, wherein the structured model data object is a claim model data object (block 330). For example, the device may insert the first one or more parts and the second one or more parts into a structured model data object, wherein the structured model data object is a claim model data object, as described above.

[0022] As further shown in Fig. 3, process 300 may include generating structured output text from the structured model data object, wherein the structured output text corresponds to one or more patent document parts (block 340). For example, the device may generate structured output text from the structured model data object, wherein the structured output text corresponds to one or more patent document parts, as described above.

[0023] As further shown in Fig. 3, process 300 may include providing the structured output text (block 350). For example, the device may provide the structured output text, as described above.

[0024] Process 300 may include additional implementations, such as any single implementation or any combination of implementations described below and/or in connection with one or more other processes described elsewhere herein.

[0025] In a first implementation, process 300 includes querying for an output template, determining whether the claim model data object is missing any information for the output template, selectively requesting missing information for the output template based at least in part

on a result of determining whether the claim model data object is missing any information for the output template, selectively including the missing information in the claim model data object based at least in part on a result of selectively requesting the missing information, and wherein generating the structured output text comprises generating the structured output text using the output template.

[0026] In a second implementation, alone or in combination with the first implementation, process 300 includes determining, using the claim model data object and an output template, a number of new claims to create, and generating claim language, for the number of new claims, by running template logic on the first one or more parts and the second one or more parts.

[0027] In a third implementation, alone or in combination with one or more of the first and second implementations, the template logic includes one or more looping functions.

[0028] In a fourth implementation, alone or in combination with one or more of the first through third implementations, the template logic includes one or more logical operators that evaluate conditions and, wherein generating the claim language comprises running the template logic based on the logical operators.

[0029] In a fifth implementation, alone or in combination with one or more of the first through fourth implementations, the template logic includes one or more helper functions that extend a template parsing functionality of a base engine of the device.

[0030] In a sixth implementation, alone or in combination with one or more of the first through fifth implementations, the output template is user-creatable.

[0031] In a seventh implementation, alone or in combination with one or more of the first through sixth implementations, the output template includes one or more sub-templates according to one or more logic rules of the output template.

[0032] In an eighth implementation, alone or in combination with one or more of the first through seventh implementations, generating the structured output text comprises converting the generated text into one or more paragraphs, wherein the one or more paragraphs are insertable into a text processing application, and wherein providing the structured output text comprises inserting the one or more paragraphs into the text processing application.

[0033] In a ninth implementation, alone or in combination with one or more of the first through eighth implementations, inserting the one or more paragraphs comprises inserting the one or more paragraphs into the document that includes the structured information.

[0034] In a tenth implementation, alone or in combination with one or more of the first through ninth implementations, the text processing application is Microsoft Word.

[0035] Although Fig. 3 shows example blocks of process 300, in some implementations, process 300 may include additional blocks, fewer blocks, different blocks, or differently arranged blocks than those depicted in Fig. 3. Additionally, or alternatively, two or more of the blocks of process 300 may be performed in parallel.

[0036] The foregoing disclosure provides illustration and description, but is not intended to be exhaustive or to limit the implementations to the precise form disclosed. Modifications may be made in light of the above disclosure or may be acquired from practice of the implementations.

[0037] As used herein, the term "component" is intended to be broadly construed as hardware, firmware, or a combination of hardware and software. It will be apparent that systems and/or methods described herein may be implemented in different forms of hardware, firmware, and/or a combination of hardware and software. The actual specialized control hardware or software code used to implement these systems and/or methods is not limiting of the implementations. Thus, the operation and behavior of the systems and/or methods are described herein without

reference to specific software code - it being understood that software and hardware can be used to implement the systems and/or methods based on the description herein.

[0038] Although particular combinations of features are recited in the claims and/or disclosed in the specification, these combinations are not intended to limit the disclosure of various implementations. In fact, many of these features may be combined in ways not specifically recited in the claims and/or disclosed in the specification. Although each dependent claim listed below may directly depend on only one claim, the disclosure of various implementations includes each dependent claim in combination with every other claim in the claim set.

[0039] No element, act, or instruction used herein should be construed as critical or essential unless explicitly described as such. Also, as used herein, the articles "a" and "an" are intended to include one or more items, and may be used interchangeably with "one or more." Further, as used herein, the article "the" is intended to include one or more items referenced in connection with the article "the" and may be used interchangeably with "the one or more." Furthermore, as used herein, the term "set" is intended to include one or more items (e.g., related items, unrelated items, a combination of related and unrelated items, etc.), and may be used interchangeably with "one or more." Where only one item is intended, the phrase "only one" or similar language is used. Also, as used herein, the terms "has," "have," "having," or the like are intended to be open-ended terms. Further, the phrase "based on" is intended to mean "based, at least in part, on" unless explicitly stated otherwise. Also, as used herein, the term "or" is intended to be inclusive when used in a series and may be used interchangeably with "and/or," unless explicitly stated otherwise (e.g., if used in combination with "either" or "only one of").

[0040] Further disclosure is included in the appendices. The appendices are provided as an example only, and is to be considered part of the specification. A definition, illustration, or other

description in the appendices does not supersede or override similar information included in the detailed description or figures. Furthermore, a definition, illustration, or other description in the detailed description or figures does not supersede or override similar information included in the appendices. Furthermore, the appendices are not intended to limit the disclosure of possible aspects.