

Figure 1

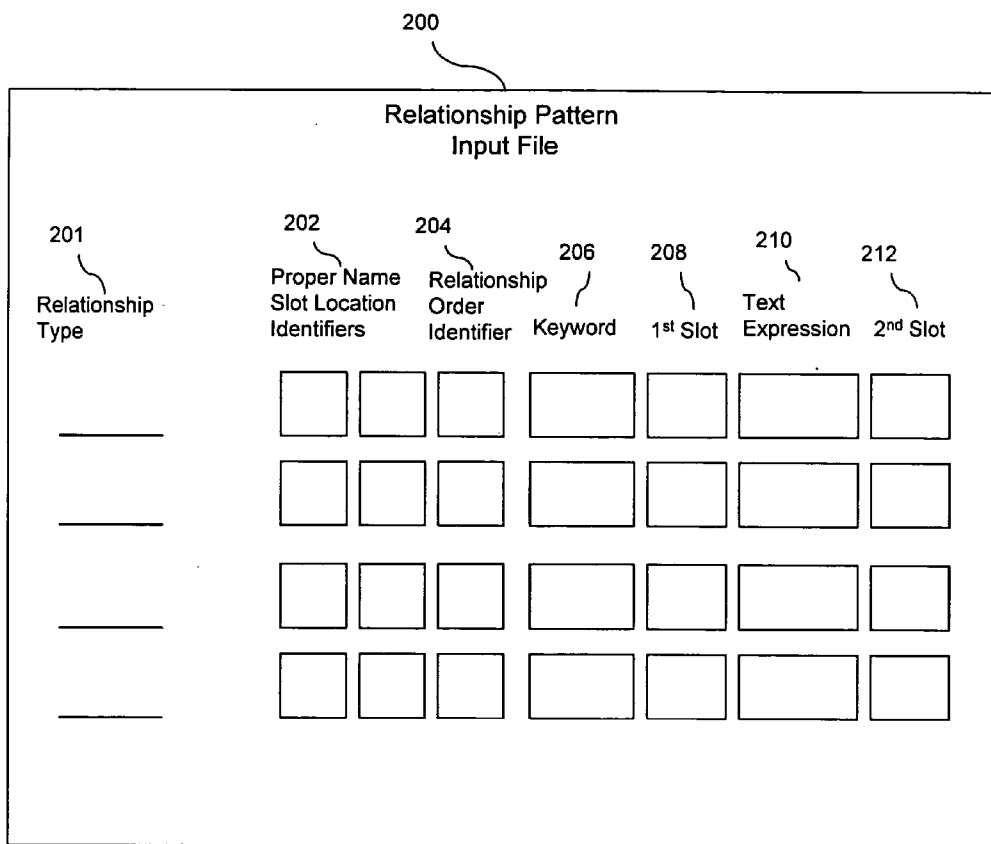


Figure 2

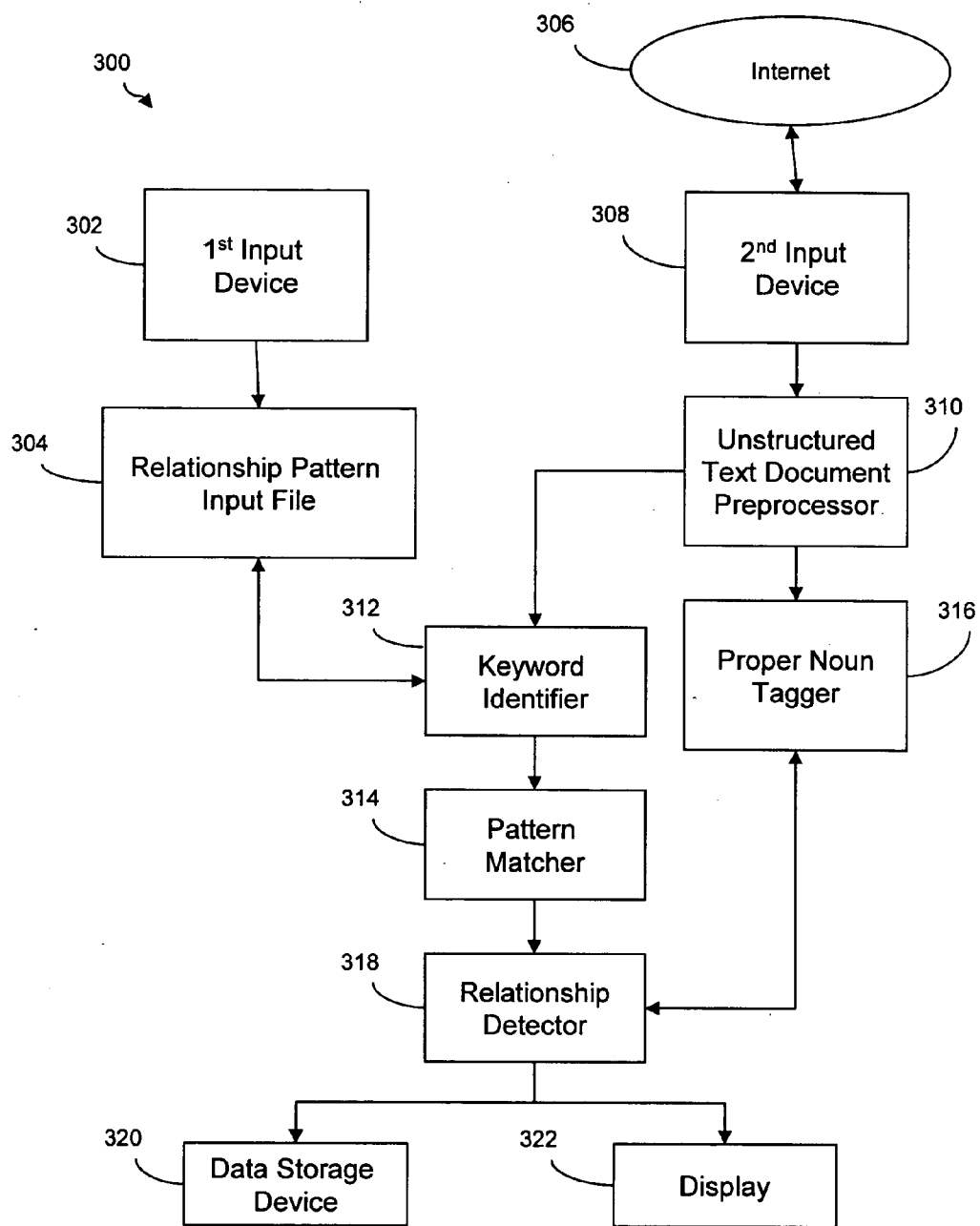


Figure 3

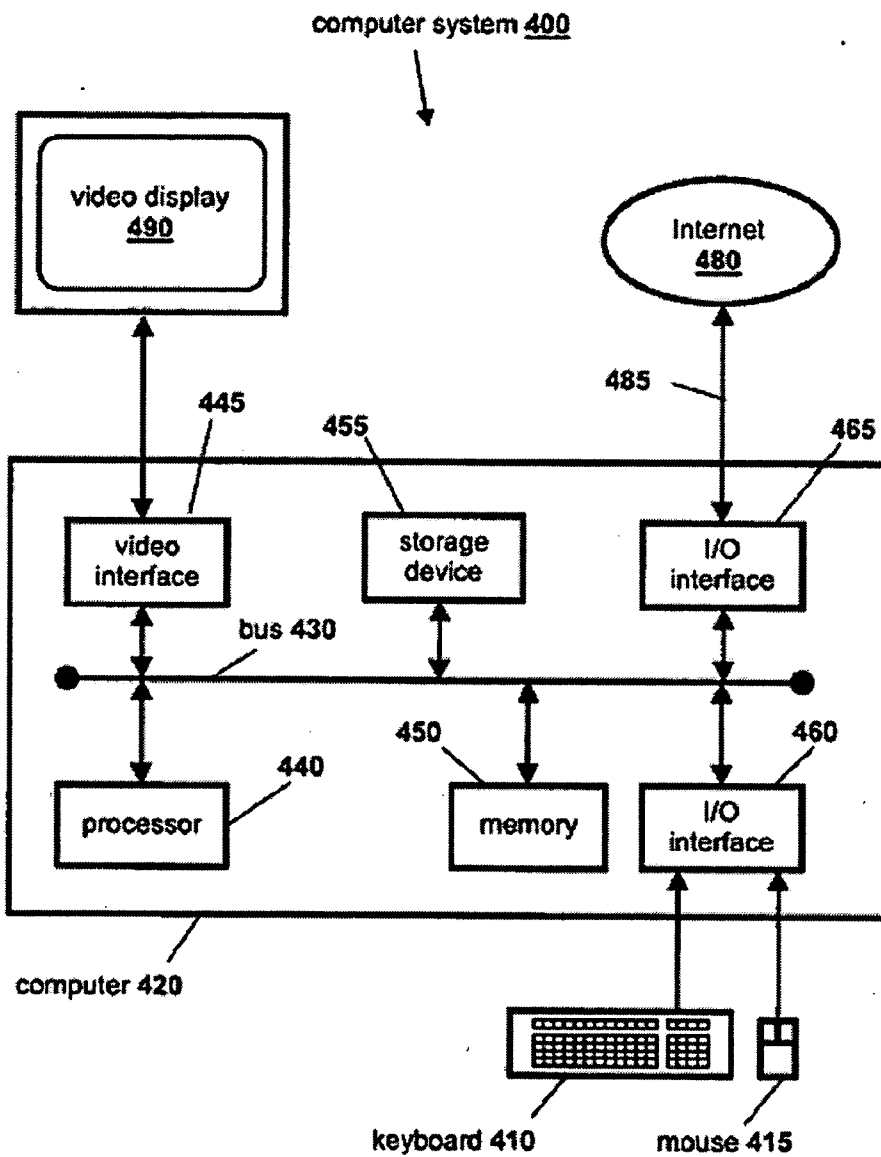


Figure 4

**DETECTING RELATIONSHIPS IN UNSTRUCTURED TEXT**

**BACKGROUND OF THE INVENTION**

**[0001]** 1. Field of the Invention

**[0002]** The invention generally relates to the field of data mining and, more particularly, to a system and a computer-implemented method of detecting relationships by creating input files of text patterns for each type of relationship, identifying a specific text pattern within a text-based document, tagging proper names in the text-based document, and extracting those proper names located within the specific text pattern so as to identify the two entities in the relationship.

**[0003]** 2. Description of the Related Art

**[0004]** Recently, there has been a rapid growth of on-line discussion groups and news websites on the World Wide Web (WWW). Detecting relationships between entities (e.g., buyer/seller, employee/employer, partnerships, parent/subsidiaries, etc.) discussed on those websites could prove to be a valuable resource (e.g., to a company investigating a rival company's business dealings, to a company or individual investigating a prospective client, employee, or contractor, etc.). However, the task of manually detecting such relationships from amongst the large corpus of documents contained on the Web is laborious. Thus, there is a need for a system and computer-implemented method for automatically and accurately detecting relationships in unstructured text contained within electronic documents with minimal processing times so as to be scalable to large document sets. The challenge is both in identifying entities in a document and in detecting the particular relationship, if any, between two entities.

**SUMMARY OF THE INVENTION**

**[0005]** In view of the foregoing, embodiments of the invention provide a system and a computer implemented method of detecting relationships in unstructured text.

**[0006]** An embodiment of a method of detecting relationships in unstructured text comprises first creating text patterns that represent different types of relationships and storing those text patterns in an input file. For example, the input file can store various text patterns representing employer/employee relationships, various patterns representing partnership relationships, etc. The text patterns may be custom-created by a user and input into the input file and/or pre-created and stored in the input file by a system manufacturer. A text pattern may be created by developing at least one regular text expression, comprising a plurality of words that describe the particular type of relationship. Additionally, the text pattern is developed with two or more slots positioned within, before, or after this regular text expression. These slots will be used in subsequent method steps, as described below, in order to identify the proper names of the entities involved in the relationship (e.g., a first slot for the name of the first entity and a second slot for the name of the second entity in the relationship). The text pattern can also be created with slot location identifiers which indicate a position of the first slot and/or a position of the second slot relative to the regular text expression. For example, the text pattern can be created with slot location

identifiers that indicate that the first slot should be located before the text expression and/or within a predetermined proximity from the text expression (e.g., within a predetermined number of words from the text expression). Similarly, the text pattern can be created with slot location identifiers to indicate that the second slot should be located after the text expression and/or within a predetermined proximity from the text expression. Additionally, the text pattern can be created with a relationship order identifier (i.e., an identifier that defines an order of the first and second entities in the relationship based on the locations of the proper names within the first and second slots). For example, if the type of relationship detected is a customer/seller relationship in which one entity is a "customer of" another entity, a relationship order identifier can be embedded in the text pattern to indicate that the proper name located in the first slot identifies the customer. Lastly, the text pattern can be created with a keyword for the particular type of relationship, and specifically, for the particular text pattern. This keyword may be used in subsequent method steps, as described below, to screen out documents prior to conducting a pattern matching analysis.

**[0007]** In addition to creating an input file, one or more text-based electronic documents (e.g., an unstructured text document (UTD)) are selected for processing by using an input device. The documents can be selected, for example, from the world wide web (WWW), from a wide area network (WAN), from a local area network, etc. The selection of documents can include a specific document, all documents in a specified category of documents, all documents having a specified date range, all documents matching a Boolean query of terms, etc. The selected unstructured text document(s) may be preprocessed, for example, by a pre-processor, in order to provide "noise free" text to either the proper noun tagger or the keyword identifier, described below.

**[0008]** Processing of a selected text-based document comprises analyzing the document in order to determine the location for each proper name occurring within the document. This can be accomplished using a multi-step process performed, for example, by a proper noun tagger. The tagger can be adapted to first scan the document in order to identify each of the proper names occurring within the document based on a predetermined set of matching rules. The set of matching rules can be based, for example, on word capitalization, sentence structure, sentence boundaries, excluded words, etc. The tagger can also be adapted to re-scan the document in order to tag and record each of the proper names found within the document along with their the locations.

**[0009]** Processing of a selected text-based document also comprises analyzing the document on a sentence by sentence basis so as to locate a text pattern within the document. This can also be accomplished using a multi-step process performed, for example, by a pattern keyword identifier and pattern matcher. The keyword identifier can be adapted to first scans the document in order to determine whether or not a keyword from one or more of the text patterns in the input file are located in the document. If a keyword for a particular text pattern is found, then a full text pattern matching process can be performed, for example, by a pattern matcher, to determine if the regular text expression defined in the particular text pattern is located in the document. If a full

text pattern is found within the document, the identity of the document is recorded and the location of the full text pattern is flagged.

[0010] Upon detection of a full text pattern with a document, a multi-step relationship detection process is performed, for example, by a relationship detector. The relationship detector refers to the list of proper names recorded by the proper noun tagger and determines if proper names are located within the first and second slots and extracts those proper names, thereby, identifying the first and second entities engaged in the relationship. Additionally, if an order for the relationship between the first and second entities is defined in the text pattern, then the relationship detector determines the order. Lastly, the relationship detector outputs the results of the relationship detection analysis. Specifically, the relationship detector can provide an output comprising the type of relationship, the names of the first and second entities engaged in the relationship, the order of the relationship (if applicable) and the identification of the document and the location in the document where the relationship was detected (i.e., the location of the text pattern), which can be stored and/or displayed.

[0011] An embodiment of a system for detecting relationships in one or more unstructured text documents comprises text pattern input files, a keyword identifier, a pattern matcher, a proper noun tagger and a relationship detector.

[0012] More specifically, the system can comprise text pattern input files stored in memory. These input files comprise text patterns that describe different types of relationships. The text patterns can be pre-created and input in the input file (e.g., by a system manufacturer) or custom developed and input into the input file by the user using an input device (e.g., a keyboard, disk, CD, internet link, hard drive, etc.). Each text pattern can comprise at least one regular text expression having a plurality of words that describe a particular relationship as well as two or more slots positioned within, before, or after this regular text expression. The slots will be used by system features, as described below, in order to identify the proper names of the entities involved in the relationship (e.g., a first slot for the name of the first entity and a second slot for the name of the second entity in the relationship). The text pattern can also comprise slot location identifiers that indicate a position of the first slot and/or a position of the second slot relative to the regular text expression, as described in detail above. Additionally, the text pattern can comprise a relationship order identifier that defines an order of the first and second entities in the relationship based on the locations of the proper names within the first and second slots, also as described in detail above. Lastly, the text pattern can comprise a keyword for the particular type of relationship and, specifically, for the particular text pattern. This keyword may be used by other features of the system, as described below, to screen out documents prior to conducting a pattern matching analysis.

[0013] A communications link can be established between the system and a source for unstructured text documents (e.g., the world wide web (WWW), a wide area network (WAN), a local area network, etc.) so that a user of the system, using an input device (e.g., a keyboard, mouse, etc.) can select one or more text-based electronic documents for analysis. The documents may be selected such that they include a specific document, all documents in a specified

category of documents, all documents having a specified date range, all documents matching a Boolean query of terms, etc. The system may further comprise a pre-processor adapted to pre-process selected unstructured text document(s) prior to analysis in order to provide "noise free" text to either the proper noun tagger or the keyword identifier, described below.

[0014] The proper noun tagger can be adapted to receive the selected unstructured text document(s) and to perform a multi-step tagging process on the documents. Specifically, the tagger can be adapted to first scan each document in order to identify each occurrence of a proper name within the document based on a predetermined set of matching rules. The set of matching rules can be based, for example, on word capitalization, sentence structure, sentence boundaries, excluded words, etc. The tagger can also be adapted to re-scan the document in order to tag and record a list of each of the proper names found within the document along with their the locations.

[0015] The keyword identifier is in communication with the relationship pattern input file and is adapted to receive the selected unstructured text document(s) (e.g., before, after, or separate from the processing by the proper noun tagger) and to analyze the document(s). Specifically, the keyword identifier is adapted to scan each document sentence by sentence in order to determine whether or not a keyword from one or more of the text patterns in the input file is located in the document. If a keyword for a particular text pattern is found, the document is forwarded to a pattern matcher for further analysis.

[0016] The pattern matcher is adapted to perform a full text pattern matching process on the forwarded document. Specifically, the pattern matcher is adapted to scan the document sentence by sentence to determine if the regular text expression defined in the particular text pattern associated with the keyword is located in the document. If a full text pattern is found within the document, the identity of the document is recorded, the location of the full text pattern is flagged, and the document is forwarded to the relationship detector.

[0017] The relationship detector is in communication with the proper noun tagger and is adapted to analyze the document further in order to detect a relationship. Specifically, the relationship detector is adapted to refer to the list of proper names recorded by the proper noun tagger and determines if proper names are located within the first and second slots for the text pattern that was located in the document. If proper names are found in both slots, the relationship detector extracts those proper names, and thereby, identifies the first and second entities engaged in the relationship described by the text pattern. Additionally, if an order for the relationship between the first and second entities is defined in the text pattern, then the relationship detector determines the order of each named entity. Lastly, the relationship detector outputs the results of the relationship detection analysis. Specifically, the relationship detector can provide an output comprising the type of relationship, the names of the first and second entities engaged in the relationship, the order of the relationship (if applicable) and the identification of the document and the location in the document where the relationship was detected (i.e., the

location of the text pattern). This output can be stored (e.g., in a data storage device) and/or displayed on a display screen.

[0018] These and other aspects of embodiments of the invention will be better appreciated and understood when considered in conjunction with the following description and the accompanying drawings. It should be understood, however, that the following description, while indicating embodiments of the invention and numerous specific details thereof, is given by way of illustration and not of limitation. Many changes and modifications may be made within the scope of the embodiments of the invention without departing from the spirit thereof, and the invention includes all such modifications.

#### BRIEF DESCRIPTION OF THE DRAWINGS

[0019] The embodiments of the invention will be better understood from the following detailed description with reference to the drawings, in which:

[0020] FIG. 1 is a schematic flow diagram of an embodiment of a method of detecting relationships in unstructured text-based electronic documents;

[0021] FIG. 2 is a schematic block diagram of an exemplary relationship text pattern input file;

[0022] FIG. 3 is a schematic block diagram representing an embodiment of a system of detecting relationships in unstructured text-based electronic documents; and,

[0023] FIG. 4 is a schematic representation of a computer system suitable for use in detecting relationships in unstructured text-based electronic documents.

#### DETAILED DESCRIPTION OF PREFERRED EMBODIMENTS OF THE INVENTION

[0024] The embodiments of the invention and the various features and advantageous details thereof are explained more fully with reference to the non-limiting embodiments that are illustrated in the accompanying drawings and detailed in the following description. It should be noted that the features illustrated in the drawings are not necessarily drawn to scale. Descriptions of well-known components and processing techniques are omitted so as to not unnecessarily obscure the embodiments of the invention. The examples used herein are intended merely to facilitate an understanding of ways in which the embodiments of the invention may be practiced and to further enable those of skill in the art to practice the embodiments of the invention. Accordingly, the examples should not be construed as limiting the scope of the invention.

[0025] As mentioned above, there is need for a system and a computer-implemented method for automatically and accurately detecting relationships (e.g., a partner relationship between two corporations, an employee-employer relationship between two people, a seller-customer relationship, etc.) in unstructured text contained within electronic documents with minimal processing times so as to be scalable to large document sets. The challenge is both in identifying entities in a document and in detecting the particular relationship, if any, between two entities. Therefore, disclosed herein are embodiments of a system and method for detecting any type of relationship that is described in unstructured

text-based electronic documents. Specifically, the system and method each incorporate the use of an input file that contains one or more text patterns that represent particular relationships. The text patterns each include regular text expressions that describe the particular relationship and slots for the location of each entity in that relationship. Document(s) are selected by a user and scanned by a proper noun tagger that identifies and tags every occurrence of a proper name within the document(s). Then, a pattern matcher scans the document(s) to match text patterns from the input file. If a text pattern is matched a relationship detector extracts the proper names found in the slots for each matched text pattern. The output from the relationship detector includes the names for each entity in a relationship, the type of relationship, and the identity of the document and the location of the sentence describing the relationship in the document.

[0026] More particularly, referring to FIG. 1, an embodiment of a method of detecting relationships in unstructured text comprises first creating text patterns 205 that represent different types of relationships 201 and storing those text patterns 205 in an input file 200, as illustrated in FIG. 2 (102-104). For example, the input file 200 can store various text patterns representing different types of relationships 201, such as employer/employee relationships, various patterns representing partnership relationships, etc. The text patterns 205 may be custom created and input into the input file 200 by a user and/or pre-created and stored in the input file 200 by a system manufacturer (e.g., as illustrated in FIG. 3 and discussed below). Any number of input files may be given as input with each file containing a list of patterns 205 for a particular relationship 201.

[0027] Specifically, the text patterns 205 may be created by developing at least one regular text expression 210, comprising a plurality of words that describe the particular type of relationship, and providing two or more slots 208, 212 positioned within, before, or after this regular text expression. These slots will be used in subsequent method steps, as described below, in order to identify the proper names of the entities involved in the relationship (e.g., a first slot for the name of the first entity and a second slot for the name of the second entity in the relationship). The text pattern 205 can also be created with slot location identifiers 202 that indicate a position of the first slot and/or a position of the second slot relative to the regular text expression. For example, the text pattern 205 can be created with a slot identifier 202a to indicate that the first slot 208 should be located before the text expression 210 and/or within a predetermined proximity from the text expression (e.g., within a predetermined number of words from the text expression). Similarly, the text pattern 205 can be created with a slot identifier 202b to indicate that the second slot 212 should be located after the text expression 210 and/or within a predetermined proximity from the text expression. Additionally, the text pattern 205 can be created with a relationship order identifier 204 (i.e., an identifier that defines an order of the first and second entities in the relationship that is not symmetric based on the locations of the proper names within the first and second slots 208 and 212). For example, if the type of relationship detected is a customer/seller relationship in which one entity is a "customer of" another entity, a relationship order identifier can be embedded in the text pattern to indicate that the proper name located in the first slot identifies the customer. Lastly, the text pattern 205



can also be created with a keyword **206** for the particular type of relationship **201**, and specifically, for the particular text pattern. This keyword **206** may be used in subsequent method steps, as described below, to screen out documents prior to conducting a pattern matching analysis and to, thereby, improve the speed of the pattern-matching.

[**0028**] For example, the text patterns **205** may be described in any language that supports regular expression matching (e.g., Perl) such that the slots **208** and **212** for the entities match the \$1 and \$2 variables after a successful match is performed. The following illustrates an exemplary text pattern for a customer relationship between two corporations:

[**0029**] O,1,1,awarded,(\*)(?:has)awarded (.\*) a (?:[<sup>^</sup>]\*){0,3}contract

[**0030**] This exemplary text pattern contains four comma-separated fields: (1) a first number (i.e., slot location identifier **202**), (2) a second number (i.e., another slot location identifier **202**), (3) a third number (i.e., a relationship order identifier **204**), (4) a keyword (**206**) for the pattern, and (5) a regular expression **210** with two slots **208** and **210**. Pursuant to Perl syntax, the text matching the first (.\*?) in the expression will be accessible via the \$1 variable after a successful match has been performed. Similarly, the second occurrence will be accessible via the \$2 variable. These two variables describe the location of the slot for each entity in the pattern. When a match is performed, these slots may contain an arbitrary amount of text. When the matching is performed, proper names are located within the slots. The first two numbers in this exemplary text pattern comprise the slot location identifiers **202** and refer to the text matched in the \$1 and \$2 slots, respectively. A 0 means that for a successful match, a proper name found within the slot must occur to the far right, a 1 means it must occur to the far left. The third number in the exemplary text pattern comprises the relationship order identifier which specifies the order of the entities in the relationship. For example, if the relationship is "customer of," a 1 in this field means that entity **1** (matched via \$1) is a customer of entity **2** (matched via \$2). A 2 in this field would mean that entity **2** is a customer of entity **1**. If this field is 0, the relationship is symmetric, as in a partnership relation.

[**0031**] At the start of the process, all input files and corresponding text patterns are loaded into memory and a mapping is created from relationship type **201** to the set of patterns **205** for that relationship.

[**0032**] Referring again to FIG. 1, in addition to creating an input file, one or more text-based electronic documents (e.g., an unstructured text document (UTD)) are selected for processing by using an input device (e.g., the same or a different input device than that used to create and input input files) (**106**). The documents can be selected, for example, from the world wide web (WWW), from a wide area network (WAN), from a local area network, etc. The selection of documents can include a specific document, all documents in a specified category of documents, all documents having a specified date range, all documents matching a Boolean query of terms, etc. Each selected unstructured text document may be preprocessed, for example, by a preprocessor, in order to provide "noise free" text to either the proper noun tagger or the keyword identifier, described below (**108**).

[**0033**] Processing of each selected text-based document comprises analyzing the document in order to determine the location for each proper name occurring within the document (**116**). This can be accomplished using a multi-step process performed, for example, by a proper noun tagger. The tagger can be adapted to first scan the document in order to identify each of the proper names occurring within each document based on a predetermined complex set of matching rules and lexicons. The set of matching rules define the proper nouns based, for example, on word capitalization, sentence structure, sentence boundaries, excluded words, etc. For example, the list of excluded words may include months, days of the week, words not capitalized in a title, etc. An exemplary rule may provide that all capitalized words, not located at the beginning of a sentence and not included on the list of excluded words, are identifiable as proper nouns. The tagger can also be adapted to re-scan the document in order to tag and record a list of each of the proper names found within the document along with their the locations.

[**0034**] Processing of each selected text-based document also comprises analyzing the document on a sentence by sentence basis so as to locate a text pattern within the document (**112-114**). This can also be accomplished using a multi-step process performed, for example, by a pattern keyword identifier and pattern matcher. The keyword identifier can be adapted to first scans the document in order to determine whether or not a keyword from one or more of the text patterns in the input file are located in the document (**112**). If a keyword for a particular text pattern is found, then a full text pattern matching process can be performed, for example, by a pattern matcher, to determine if the regular text expression defined in the particular text pattern is located in the document (**114**). If a full text pattern is found within the document, the identity of the document is recorded and the location of the full text pattern is flagged (**115**).

[**0035**] Upon detection of a full text pattern within a document, a multi-step relationship detection process is performed, for example, by a relationship detector. The relationship detector refers to the list of proper names recorded by the proper noun tagger and determines if proper names are located within the first and second slots and extracts those proper names, thereby, identifying the first and second entities engaged in the relationship (**119**). Additionally, if an order for the relationship between the first and second entities is defined in the text pattern, then the relationship detector determines the order. Lastly, the relationship detector outputs the results of the relationship detection analysis (**120**). Specifically, the relationship detector can provide an output comprising the type of relationship, the names of the first and second entities engaged in the relationship, the order of the relationship (if applicable) and the identification of the document and the location in the document where the relationship was detected (i.e., the location of the text pattern), which can be stored (**122**) and/or displayed (**124**).

[**0036**] Referring to FIG. 3, an embodiment of a system **300** for detecting relationships in one or more unstructured text documents comprises a text pattern input file **304**, a keyword identifier **312**, a pattern matcher **314**, a proper noun tagger **316** and a relationship detector **318**.

[0037] More specifically, the system 300 can comprise input files 304 stored in memory 305 (e.g., a hard drive, a disk, data storage device, etc.). The input files 304, as illustrated in FIG. 2 and discussed above, comprise a plurality of text patterns 205 that describe different types of relationships 201. These text patterns 205 can be pre-created and input in the input files 304 (e.g., by a system manufacturer) or custom developed and input into the input files 304 by the user using an input device 302 (e.g., a keyboard, disk, CD, internet link, hard drive, etc.).

[0038] Each text pattern 205 can comprise at least one text expression 210, discussed in detail above, having a plurality of words that describe a particular relationship 201 as well as two or more slots 208, 212 positioned within, before, or after this regular text expression 210. The slots 208, 212 will be used by the relationship detector 318, as described below, in order to identify the proper names of the entities involved in the relationship (e.g., a first slot 208 for the name of the first entity and a second slot 212 for the name of the second entity in the relationship). The text pattern 205 can also comprise slot location identifiers 202a-b that indicate a position of the first slot 208 and/or a position of the second slot 212 relative to the regular text expression 210, as described in detail above. Additionally, the text pattern 205 can comprise a relationship order identifier 204 that defines an order of the first and second entities in the relationship based on the locations of the proper names within the first and second slots 208, 212, also as described in detail above. Lastly, the text pattern 205 can comprise a keyword 206 for the particular type of relationship 201 and, specifically, for the particular text pattern 205. This keyword 206 may be used by the keyword identifier 312, as described below, to screen out documents prior to conducting a pattern matching analysis in order to improve processing speed.

[0039] A communications link 307 can be established between the system 300 and a source 306 for unstructured text documents (e.g., the Internet, the world wide web (WWW), a wide area network (WAN), a local area network, etc.) so that a user of the system 300 can select, using an input device 308 (e.g., a keyboard, a mouse, etc.) one or more text-based electronic documents 309 for analysis. The document(s) may be selected to include specific document(s), all documents in a specified category of documents, all documents having a specified date range, all documents matching a Boolean query of terms, etc. The system 300 may further comprise a pre-processor 310 adapted to pre-process each selected unstructured text document 309 prior to analysis in order to provide "noise free" text to either the proper noun tagger 315 or the keyword identifier 312, described below.

[0040] The proper noun tagger 315 can be adapted to receive each selected unstructured text document 309 and to perform a multi-step tagging process on the documents. Specifically, the tagger 315 can be adapted to first scan each document in order to identify each occurrence of a proper name within the document based on a predetermined and complex set of matching rules and lexicons. The set of matching rules can be based, for example, on at least one of word capitalization, sentence structure, sentence boundaries, and excluded words (e.g., as illustrated in the detail discussion above). The tagger 315 can also be adapted to re-scan the document(s) 309 in order to tag each proper name and

record a list of each of the proper names found within the document along with their the locations 317 in memory 318.

[0041] The keyword identifier 312 is in communication with (i.e., is adapted to access) the relationship pattern input file 304 and is further adapted to receive the selected unstructured text document(s) 309 from the preprocessor 310 (e.g., before, after, or separate from the processing by the proper noun tagger) and to analyze each document 309. Specifically, the keyword identifier 312 is adapted to scan each document 309 sentence by sentence in order to determine whether or not a keyword 206 from one or more of the text patterns 205 in the input file (as illustrated in FIG. 2) are located in the document 309. If a keyword 206 for a particular text pattern 205 is found, the document containing the keyword is forwarded to a pattern matcher 314 for further analysis.

[0042] The pattern matcher 314 is adapted to perform a full text pattern matching process on the forwarded document. Specifically, the pattern matcher 314 is adapted to scan the document sentence by sentence to determine if the regular text expression defined in the particular text pattern associated with the keyword is located in the document. If a full text pattern is found within the document, the identity of the document and the location of the full text pattern 320 is recorded in a memory 319 that is accessible by the relationship detector 318. The document that contains the full text pattern is then forwarded to the relationship detector 318 for further analysis.

[0043] The relationship detector 318 is adapted to further analyze the document that contains the full text pattern in order to detect a relationship and, particularly, the entities engaged in the relationship. Specifically, the relationship detector 318 is adapted to access the memory 316 in order to refer to the list of proper names 317 recorded by the proper noun tagger 315. The relationship detector 318 then reviews the document and determines if proper names are located within the first and second slots for the text pattern that was located within the document. If proper names are found in both slots, the relationship detector 318 extracts those proper names, and thereby, identifies the first and second entities engaged in the relationship described by the text pattern. Additionally, if an order for the relationship between the first and second entities is defined in the text pattern, then the relationship detector 318 determines the order of each named entity. Lastly, the relationship detector outputs the results of the relationship detection analysis. Specifically, the relationship detector 318 can provide an output comprising the type of relationship (as defined by the text pattern), the names of the first and second entities engaged in the relationship, the order of the relationship (if applicable) and the identification of the document and the location in the document where the relationship was detected (i.e., the location of the text pattern). This output can be stored (e.g., in a data storage device 322) and/or displayed on a display screen 324.

[0044] Embodiments of the system 300, described above, can take the form of an entirely hardware embodiment, an entirely software embodiment or an embodiment including both hardware and software elements. In a preferred embodiment, the invention is implemented using software, which includes but is not limited to firmware, resident software, microcode, etc. Furthermore, embodiments of the

system **300** can take the form of a computer program product accessible from a computer-usable or computer-readable medium providing program code for use by or in connection with a computer or any instruction execution system. For the purposes of this description, a computer-usable or computer readable medium can be any apparatus that can comprise, store, communicate, propagate, or transport the program for use by or in connection with the instruction execution system, apparatus, or device. The medium can be an electronic, magnetic, optical, electromagnetic, infrared, or semiconductor system (or apparatus or device) or a propagation medium. Examples of a computer-readable medium include a semiconductor or solid state memory, magnetic tape, a removable computer diskette, a random access memory (RAM), a read-only memory (ROM), a rigid magnetic disk and an optical disk. Current examples of optical disks include compact disk—read only memory (CD-ROM), compact disk—read/write (CD-R/W) and DVD. A data processing system suitable for storing and/or executing program code will include at least one processor coupled directly or indirectly to memory elements through a system bus. The memory elements can include local memory employed during actual execution of the program code, bulk storage, and cache memories which provide temporary storage of at least some program code in order to reduce the number of times code must be retrieved from bulk storage during execution.

[0045] FIG. 4 is a schematic representation of an exemplary computer system **400** suitable for use in detecting relationships as described herein. Computer software executes under a suitable operating system installed on the computer system **400** to assist in performing the described techniques. This computer software is programmed using any suitable computer programming language, and may be thought of as comprising various software code means for achieving particular steps. The components of the computer system **400** include a computer **420**, a keyboard **410** and a mouse **415**, and a video display **490**. The computer **420** includes a processor **440**, a memory **450**, input/output (I/O) interfaces **460**, **465**, a video interface **445**, and a storage device **455**. The processor **440** is a central processing unit (CPU) that executes the operating system and the computer software executing under the operating system. The memory **450** includes random access memory (RAM) and read-only memory (ROM), and is used under direction of the processor **440**. The video interface **445** is connected to video display **490**. User input to operate the computer **420** is provided from the keyboard **410** and mouse **415**. The storage device **455** can include a disk drive or any other suitable storage medium. Each of the components of the computer **420** is connected to an internal bus **430** that includes data, address, and control buses, to allow components of the computer **420** to communicate with each other via the bus **430**. The computer system **400** can be connected to one or more other similar computers via input/output (I/O) interface **465** using a communication channel **465** to a network, represented as the Internet **480**. The computer software may be recorded on a portable storage medium, in which case, the computer software program is accessed by the computer system **400** from the storage device **455**. Alternatively, the computer software can be accessed directly from the Internet **480** by the computer **420**. In either case, a user can interact with the computer system **400** using the keyboard **410** and mouse **415** to operate the programmed computer software execut-

ing on the computer **420**. Other configurations or types of computer systems can be equally well used to implement the described techniques. The computer system **400** described above is described only as an example of a particular type of system suitable for implementing the described techniques.

[0046] Therefore, disclosed above are embodiments of a system and a method for detecting relationships described in unstructured text-based electronic documents. The system and method incorporate the use of an input file that contains one or more text patterns that represent particular relationships. The text patterns each include regular text expressions that describe the particular relationship and slots for the location of each entity in that relationship. Document(s) are selected by a user and scanned by a proper noun tagger that identifies and tags every occurrence of proper names within the document(s). Then, a pattern matcher scans the document(s) to match text patterns. If a text pattern is matched within a document a relationship detector extracts all pairs of proper names found in the slots for each matched text pattern. The output from the relationship detector includes the names for each entity in the relationship, the type of relationship, and the identity of the document and the location of the sentence describing the relationship in the document. This method and associated system are extremely cost and time efficient because they avoid the need of natural language processing or parsing (i.e., running expensive machines such as parsers and parts-of-speech taggers is unnecessary), so that they are scalable to a large number of documents. Additionally, because a user may define the text patterns with regular text expressions (as opposed to a single word or simple phrase) describing each relationship, the system and method are applicable to any type of relationship and are very precise in detecting particular relationships.

[0047] The foregoing description of the specific embodiments will so fully reveal the general nature of the invention that others can, by applying current knowledge, readily modify and/or adapt for various applications such specific embodiments without departing from the generic concept, and, therefore, such adaptations and modifications should and are intended to be comprehended within the meaning and range of equivalents of the disclosed embodiments. It is to be understood that the phraseology or terminology employed herein is for the purpose of description and not of limitation. Therefore, while the invention has been described in terms of preferred embodiments, those skilled in the art will recognize that the invention can be practiced with modification within the spirit and scope of the appended claims.

What is claimed is:

1. A computer implemented method of detecting a relationship between a first entity and a second entity, said method comprising:

- creating a text pattern that represents a type of relationship, wherein said text pattern comprises a first slot for said first entity and a second slot for said second entity;
- analyzing a text-based document so as to locate said text pattern within said document;
- determining a location for each proper name occurring within said document; and
- extracting proper names located within said first slot and said second slot of said text pattern within said docu-

ment, wherein said proper names located within said first slot and said second slot identify said first entity and said second entity.

2. The method of claim 1, wherein said creating of said text pattern further comprises identifying a keyword describing said relationship and wherein said method further comprises before said analyzing of said document, reviewing said document to determine if said keyword is located in said document.

3. The method of claim 1, wherein said creating of said text pattern further comprises:

creating at least one text expression comprising a plurality of words that describe said type of said relationship; and

setting a position of said first slot and said second slot relative to said at least one text expression.

4. The method of claim 1, wherein said determining of said location of each of said proper names comprises:

scanning said document to identify each of said proper names occurring within said document based on a set of matching rules;

re-scanning said document to tag said location for each of said proper names identified; and

recording said location for each of said proper names.

5. The method of claim 4, wherein said set of matching rules is based on at least one of word capitalization, sentence structure, sentence boundaries, and excluded words.

6. The method of claim 1, wherein said creating of said text pattern further comprises defining an order of said first entity and said second entity in said relationship based on said locations of said proper names within said first slot and said second slot.

7. The method of claim 1, further comprising storing a record of said relationship comprising at least one of said proper name of said first entity, said proper name of said second entity, said type of relationship between said first entity and said second entity, said order of said first entity and said second entity in said relationship, and an identifier for said document and a location in said document where said relationship is detected.

8. A system for detecting a relationship between a first entity and a second entity, said system comprising:

an input file adapted to store a text pattern that describes a type of relationship, wherein said text pattern comprises a first slot for said first entity and a second slot for said second entity;

a pattern matcher in communication with said input file and adapted to analyze a text-based document so as to locate said text pattern within said document;

a proper noun tagger adapted to locate and record occurrences of proper names within said document; and

a relationship detector in communication with said pattern matcher and said proper noun tagger and adapted to extract said proper names located within said first slot and said second slot of said text pattern within said document so as to identify said first entity and said second entity and, thereby, detect said relationship.

9. The system of claim 8, further comprising a keyword identifier in communication with said input file and adapted

to review said document for said keyword and to forward said document to said pattern matcher only if said keyword is located in said document.

10. The system of claim 8, wherein said text pattern further comprises:

at least one text expression comprising a plurality of words that describe said relationship; and

positions for said first slot and said second slot relative to said at least one text expression.

11. The system of claim 8, wherein said text pattern further comprises an order of said first entity and said second entity in said relationship based said locations of said proper names within said first slot and said second slot.

12. The method of claim 8, wherein said proper noun tagger is further adapted to scan said document to identify each of said proper names occurring within said document based on a set of matching rules, to re-scan said document to tag said location for each of said proper names, and to record said location for each of said proper names within said document.

13. The system of claim 12, wherein said set of matching rules is based on at least one of word capitalization, sentence structure, sentence boundaries, and excluded words.

14. The system of claim 11, further comprising at least one of a data storage device adapted to store at least one of said proper name of said first entity, said proper name of said second entity, said relationship between said first entity and said second entity, said order of said first entity and said second entity in said relationship, and a record of said document in which said relationship is detected.

15. A program storage device readable by computer and tangibly embodying a program of instructions executable by said computer to perform a method of detecting a relationship between a first entity and a second entity, said method comprising:

creating a text pattern that represents a type of relationship, wherein said text pattern comprises a first slot for said first entity and a second slot for said second entity;

analyzing a text-based document so as to locate said text pattern within said document;

determining a location for each proper name occurring within said document; and

extracting proper names located within said first slot and said second slot of said text pattern within said document, wherein said proper names located within said first slot and said second slot identify said first entity and said second entity

16. The program storage device of claim 15, wherein said creating of said text pattern further comprises identifying a keyword describing said relationship and wherein said method further comprises before said analyzing of said document, reviewing said document to determine if said keyword is located in said document.

17. The program storage device of claim 15, wherein said creating of said text pattern further comprises:

creating at least one text expression comprising a plurality of words that describe said type of said relationship; and

setting a position of said first slot and said second slot relative to said at least one text expression.

**18.** The program storage device of claim 15, wherein said determining of said location for each of said proper names comprises:

scanning said document to identify each of said proper names occurring within said document based on a set of matching rules;

re-scanning said document to tag said location for each of said proper names; and

recording said location for each of said proper names.

**19.** The program storage device of claim 18, wherein said set of matching rules is based on at least one of word capitalization, sentence structure, sentence boundaries, and excluded words.

**20.** The program storage device of claim 15, wherein said creating of said text pattern further comprises defining an order of said first entity and said second entity in said relationship based on said locations of said proper names within said first slot and said second slot.

\* \* \* \* \*