US 20250265297A1

(19) United States

(12) Patent Application Publication (10) Pub. No.: US 2025/0265297 A1

WYLE et al. (43) Pub. Date: Aug. 21, 2025

(54) **DOCUMENT MATCHING**

(71) Applicant: **Sureprep, LLC**, Irvine, CA (US)

(72) Inventors: **DAVID A. WYLE**, Corona Del Mar, CA (US); **ALEXANDER JAMES SADOVSKY**, Denver, CO (US); **WILLIAM W. HOSEK**, Laguna Niguel, CA (US)

(73) Assignee: **Sureprep, LLC**, Irvine, CA (US)

(21) Appl. No.: **19/176,703**

(22) Filed: **Apr. 11, 2025**

Related U.S. Application Data

(63) Continuation of application No. 18/506,695, filed on Nov. 10, 2023, now Pat. No. 12,306,882, which is a continuation of application No. 17/217,917, filed on Mar. 30, 2021, now Pat. No. 11,860,950.

Publication Classification

(51) **Int. Cl.**

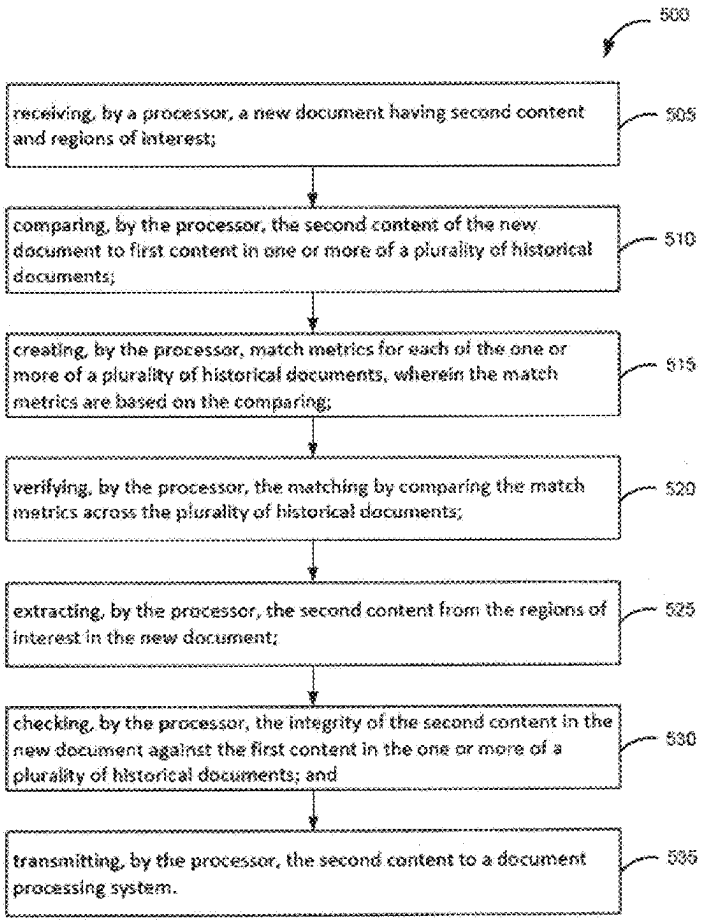
G06F 16/93	(2019.01)
G06N 3/08	(2023.01)
G06V 30/412	(2022.01)
G06V 30/416	(2022.01)
G06V 30/418	(2022.01)

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

CPC ***G06F 16/93*** (2019.01); ***G06N 3/08*** (2013.01); ***G06V 30/412*** (2022.01); ***G06V 30/416*** (2022.01); ***G06V 30/418*** (2022.01)

(57) **ABSTRACT**

The system is configured to create a generalized document automation framework that captures relevant data from documents based upon replicating historical human actions associated with a document. The system may use machine vision and natural language processing to match a new document to a document that was already human extracted in an existing corpus. This is accomplished by comparing both visual elements and textual elements. This match can be verified by statistical approaches by comparing the match metrics across multiple documents. After the match has been found and verified, the system then uses the historical extractions from the historical document and maps the extractions to similar regions in the new document based upon again both visual and text commonalities between documents. Data is then extracted from these regions of interest in the new document, sanity checked for data integrity against historical data, and then passed downstream for processing.



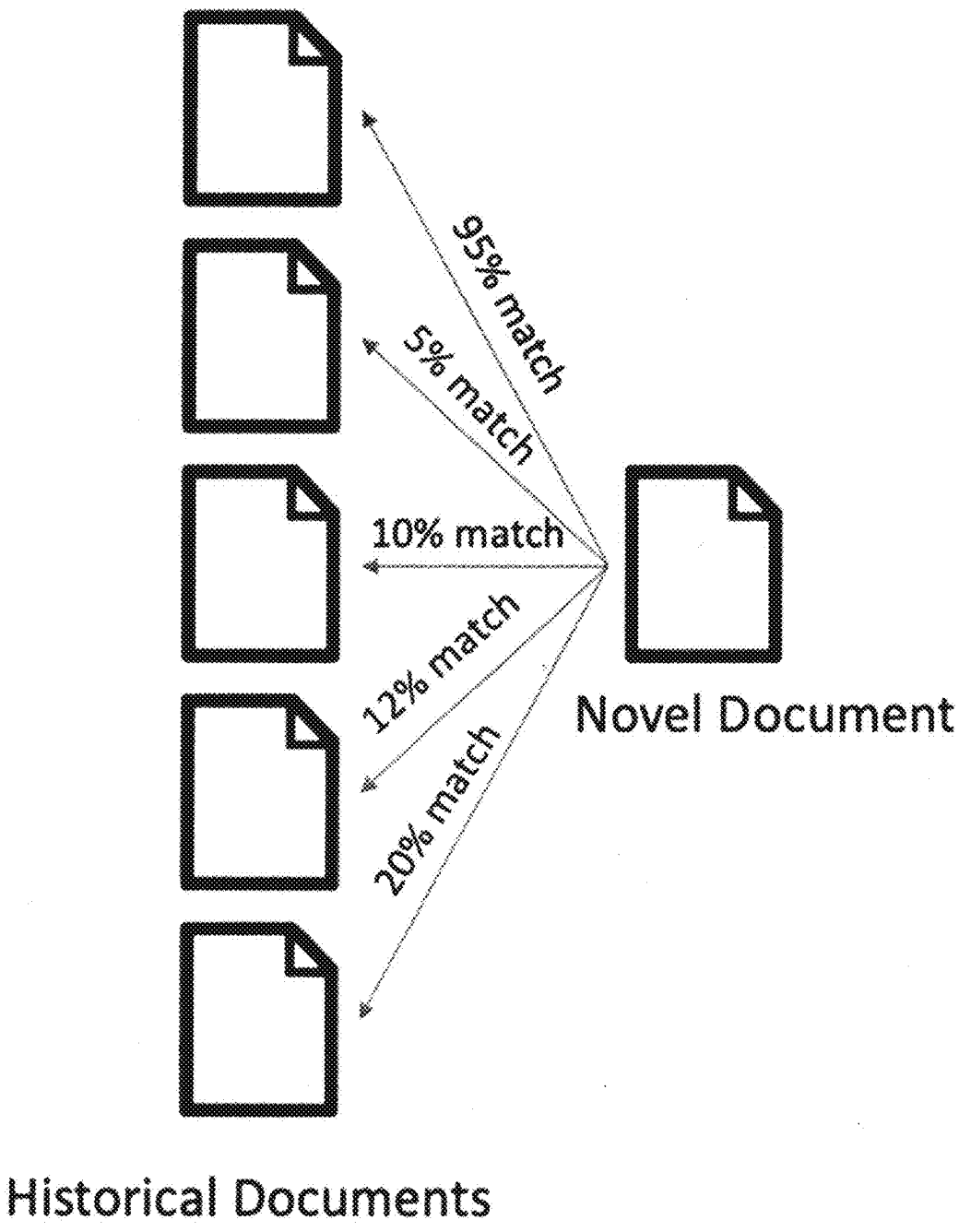

95% match
5% match
10% match
12% match
20% match
Novel Document
Historical Documents

FIGURE 1

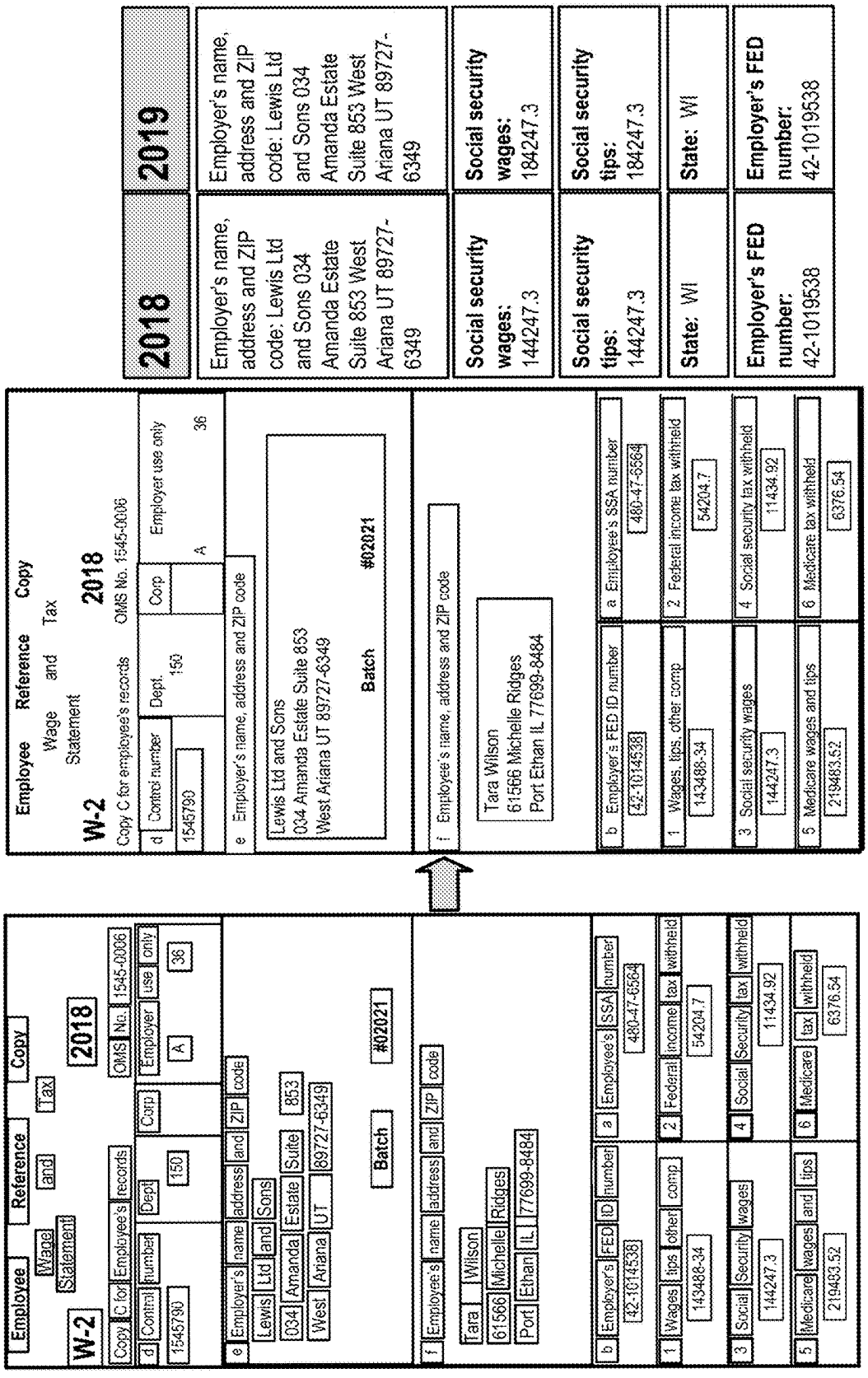

Raw Text Detection
Header/Value Identification
Document [Header:Value] Text Similarity
W-2
2018
Wage and Tax Statement
Copy C for employee's records
OMB No. 1545-0006
Employee Reference Copy
Employer's name, address and ZIP code: Lewis Ltd and Sons 034 Amanda Estate Suite 853 West Ariana UT 89727-6349
Employee's name, address and ZIP code: Tara Wilson 61566 Michelle Ridges Port Ethan IL 77699-8484
Batch #02021
Employer's FED ID number: 42-1014538
Employee's SSA number: 480-47-6564
Wages, tips, other comp: 143488.34
Federal income tax withheld: 54204.7
Social security wages: 144247.3
Social security tax withheld: 11434.92
Medicare wages and tips: 219483.52
Medicare tax withheld: 6376.54
Social security wages: 184247.3
Social security tips: 184247.3
State: WI
Employer's FED number: 42-1019538

FIG. 2

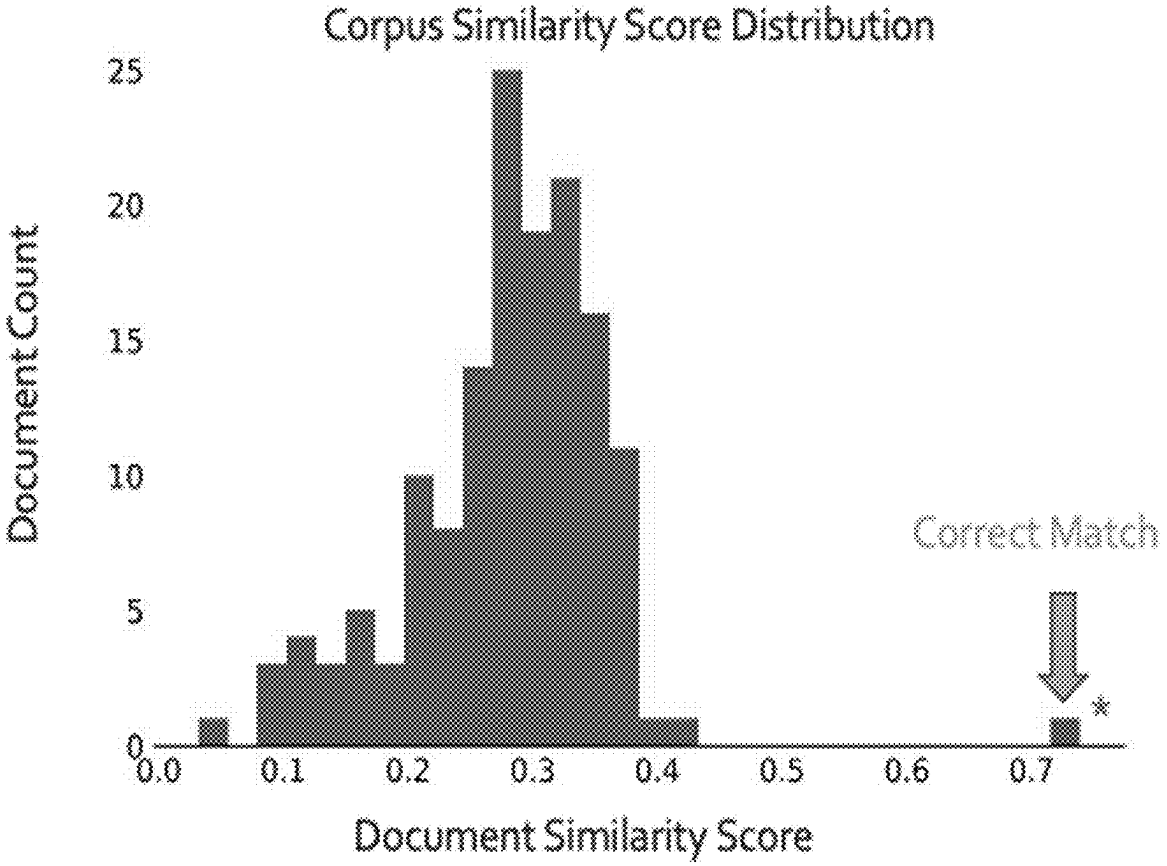

Corpus Similarity Score Distribution
Document Count
25
20
15
10
5
0
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
Correct Match
Document Similarity Score

FIGURE 3

Historical Document

1 Wages, tips, other compensation 228839.47	2 Federal income tax withheld 78836.81
3 Social security wages 234367.9	4 Social security tax withheld 17929.14
5 Medicare wages and tips 272320.44	6 Medicare tax withheld 7897.29
7 Social security tips 234367.9	8 Allocated tips 272320.44
9 Advance EIC payment	10 Dependent care benefits 133
11 Nonqualified plans 202	12a See instructions for box 12 H | 9473

Novel Document

1 Wages, tips, other compensation 232742.96	2 Federal income tax withheld 80767.81
3 Social security wages 164531.06	4 Social security tax withheld 12586.63
5 Medicare wages and tips 268492.56	6 Medicare tax withheld 7786.28
7 Social security tips 164531.06	8 Allocated tips 268492.56
9 Advance EIC payment	10 Dependent care benefits 176
11 Nonqualified plans 207	12a See instructions for box 12 | 3391

Manually Extracted Data	
Value 1	228839
Value 2	7897
Value 3	133

Automated Extracted Data	
Value 1	232743
Value 2	7786
Value 3	176

FIG. 4

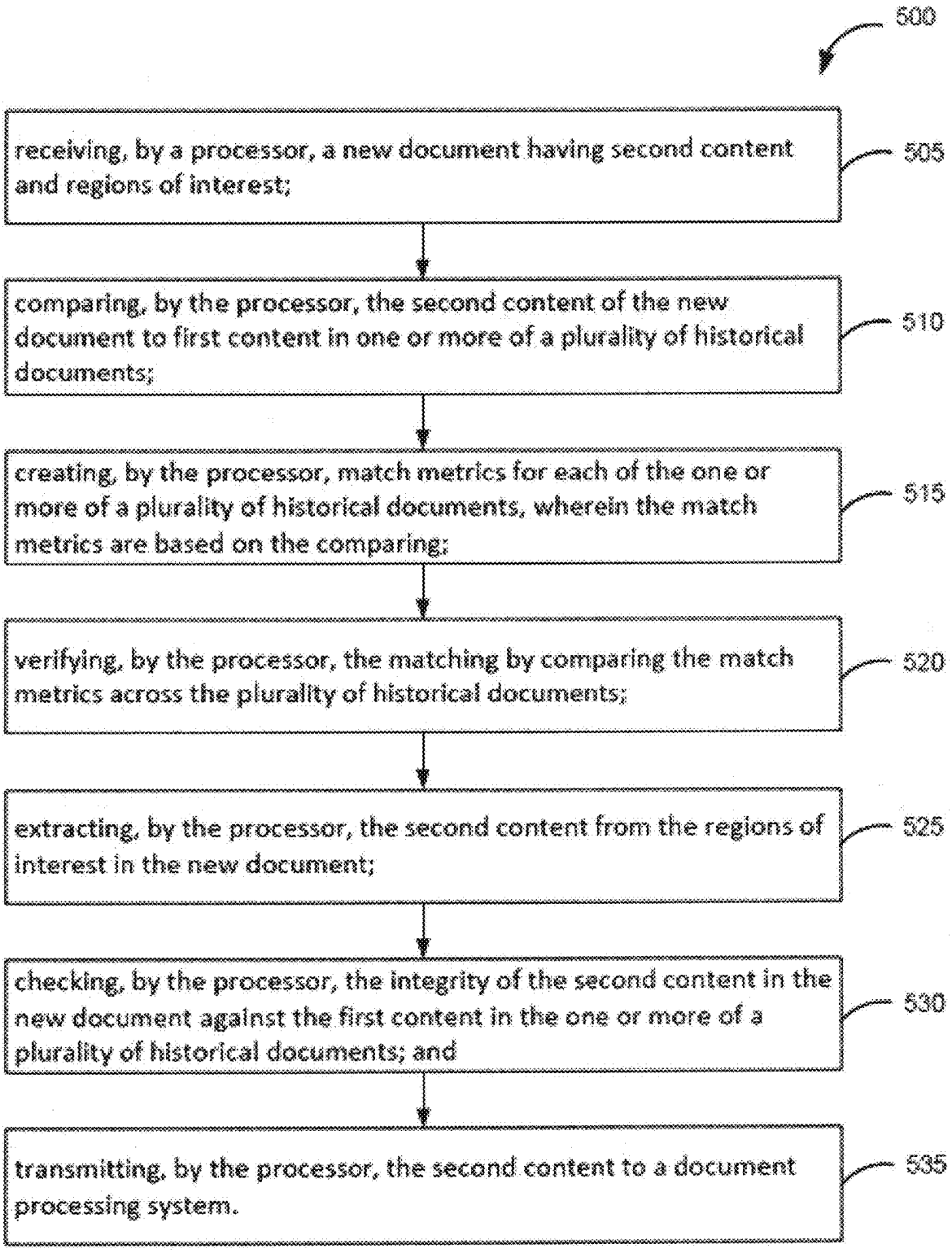

500
receiving, by a processor, a new document having second content and regions of interest;
505
comparing, by the processor, the second content of the new document to first content in one or more of a plurality of historical documents;
510
creating, by the processor, match metrics for each of the one or more of a plurality of historical documents, wherein the match metrics are based on the comparing;
515
verifying, by the processor, the matching by comparing the match metrics across the plurality of historical documents;
520
extracting, by the processor, the second content from the regions of interest in the new document;
525
checking, by the processor, the integrity of the second content in the new document against the first content in the one or more of a plurality of historical documents; and
530
transmitting, by the processor, the second content to a document processing system.
535

FIGURE 5

DOCUMENT MATCHING

CROSS-REFERENCE TO RELATED APPLICATIONS

[0001] This application is a continuation of, claims priority to and the benefit of, U.S. application Ser. No. 18/506,695 filed Nov. 10, 2023 and entitled "Document Matching Using Artificial Intelligence." The '695 application is a continuation of, claims priority to and the benefit of, U.S. application Ser. No. 17/217,917 filed Mar. 30, 2021, now U.S. Pat. No. 11,860,950 issued Jan. 2, 2024 and entitled "Document Matching and Data Extraction," both of which are hereby incorporated by reference in its entirety for all purposes.

TECHNICAL FIELD

[0002] This disclosure generally relates to document matching and data extraction, and more particularly, to a system and method for recognizing regions of interest between similar documents by comparing visual elements and textual elements.

BACKGROUND

[0003] Different types of businesses often carefully curate and extract a large volume of documents. For example, a large set of accounting documents (in the form of images and/or portable document files) are typically sent to a tax preparer, who then has the task of identifying and extracting relevant information from the accounting documents.

[0004] To provide more efficiency, businesses have tried to automate this workflow by incorporating template-based optical character recognition (OCR). Businesses have also used rigid, specific rule-based methods. For example, businesses would often perform optical character recognition to use the expected positioning of text on a document to both identify the document type and to further extract and annotate data from that document.

[0005] Templated-based OCR often includes trained humans to create each template. A human with detailed knowledge of the OCR system and document variability must review every document to specifically create sets of rules detailing exactly how to extract data from each of the documents. Templated-based OCR also usually requires trained humans to maintain each template. However, templates often degrade in performance as documents change. While some variability can be explicitly declared in the template, any unaccounted-for changes usually require humans to modify a template to account for the differences or to create a new template.

[0006] Moreover, templated-based OCR approaches typically cannot adapt to dynamic documents. In particular, some documents may be structured, but the documents have variability in table length and data positioning. While a human is often able to detect these similar regions of interest from document to document, it is extremely difficult to create generic rules/templates that are able to capture these regions and specify how to process the regions.

[0007] Furthermore, resource constraints usually exist in template creation. For an organization that may process millions of documents, the cost/benefit tradeoff required for spending the time and money to create a templated-based OCR may not be sufficient to automate the analysis of documents that appear infrequently. Instead, these documents are typically curated by hand each time the documents are encountered.

[0008] Therefore, a long-felt need exists for a way to turn such hand curation into automation via artificial intelligence approaches that do not require human interaction.

SUMMARY

[0009] A system and method are disclosed for recognizing and extracting relevant data from regions of interest between similar documents by comparing visual elements and textual elements. In various embodiments, the system may include receiving, by a processor, a new document having second content and regions of interest; comparing, by the processor, the second content of the new document to first content in one or more of a plurality of historical documents; creating, by the processor, match metrics for each of the one or more of a plurality of historical documents, wherein the match metrics are based on the comparing; verifying, by the processor, the comparing by comparing the match metrics across the plurality of historical documents; extracting, by the processor, the second content from the regions of interest in the new document; checking, by the processor, the integrity of the second content in the new document against the first content in the one or more of a plurality of historical documents; and transmitting, by the processor, the second content to a document processing system.

[0010] The comparing may be based on at least one of visual elements or textual elements. The receiving may be via at least one of an image or a PDF. The verifying may be based on statistical properties of the match of the new document compared to population statistics of the plurality of historical documents. The comparing may further comprise comparing at least one of text headers, text values, X-Y coordinate positioning or visual elements. The method may further comprise transmitting, by the processor, the new document with the second content to a human for verification, in response to the checking the integrity of the second content in the new document being a failure.

[0011] The comparing may further comprise comparing at least one of visual elements or textual elements. The comparing may further comprise comparing at least one of logos, fonts, lines or layouts. The comparing may further comprise comparing text from at least one of headers, values or tables. The comparing may be accomplished by at least one of location un-aware OCR detected text or visual document similarity indexing. The comparing may further comprise detecting, by the processor, raw text of the second content in the new document. The comparing may further comprise detecting, by the processor, tables of the second content in the new document. The comparing may further comprise pulling, by the processor, at least one of headers or text values from raw text of the second content in the new document. The comparing may further comprise pulling, by the processor, at least one of headers or text values from raw text of the second content in the new document, based on at least one of proximity or formatting. The comparing may further comprise detecting, by the processor, raw text of the first content in the historical document. The comparing may further comprise detecting, by the processor, tables of the first content in the historical document. The comparing may further comprise pulling, by the processor, at least one of headers or text values from raw text of the first content in the historical document. The comparing may further comprise

pulling, by the processor, at least one of headers or text values from raw text of the first content in the historical document, based on at least one of proximity or formatting. The comparing may further comprise comparing, by the processor, visual elements between the new document and the historical document. The comparing may further comprise comparing, by the processor, visual elements between the new document and the historical document using at least one of mean square error (MSE), a structural similarity index measure (SSIM) or an image similarity metric as determined by a deep neural network.

BRIEF DESCRIPTION OF THE DRAWINGS

[0012] The accompanying drawings, wherein like numerals depict like elements, illustrate exemplary embodiments of the present disclosure, and together with the description, serve to explain the principles of the disclosure. In the drawings:

[0013] FIG. **1** is an exemplary graphic showing matching scores based on historical matching with a new document, in accordance with various embodiments.

[0014] FIG. **2** is an exemplary non-templated text comparison, in accordance with various embodiments.

[0015] FIG. **3** is an exemplary document match verification, in accordance with various embodiments.

[0016] FIG. **4** is an exemplary data extraction and classification using the detected regions of interest, in accordance with various embodiments.

[0017] FIG. **5** is an exemplary flow chart showing the method, in accordance with various embodiments.

DETAILED DESCRIPTION

[0018] The system directly removes the need for human maintained and rigid OCR templates by creating a dynamic, data extraction platform based around the ability to recognize regions of interest between similar documents. In various embodiments, the system is configured to create a generalized document automation framework that captures relevant data from documents based upon replicating historical human actions associated with a document. In general, in various embodiments and with respect to FIG. **5**, the system may receive a new document having second content and regions of interest (step **505**). The system may use machine vision and natural language processing to compare the new document (e.g., current year tax form) to an historical document (e.g., prior year tax form) (step **510**). The historical document may have already been subject to human data extraction in an existing corpus. This is accomplished by comparing visual elements (e.g., logos, fonts, lines, and layout), and textual elements (e.g., text contained in headers, values, and tables on the document). The system may create match metrics for each of the one or more of a plurality of historical documents, wherein the match metrics are based on the comparing (step **515**). The system reviews the historical documents and maps the extractions of first content to similar regions in the new document based upon both visual and text commonalities between documents. As used herein, a "match" may include an identical match, a partial match, a correlation, similar content, similar layout, etc. This match can be verified by statistical approaches by comparing the match metrics across multiple documents (step **520**). After the match has been found and verified, in various embodiments, the system extracts the second content from the new document (step **525**). In various embodiments, the data may also be extracted from these regions of interest in the new document, sanity checked for data integrity against historical data (step **530**), and then transmitted downstream for processing (step **535**). The further processing may be conducted by a document processing system. The document processing system may process any items that are generated periodically. The items may include, for example, health-care forms, billing forms, invoicing forms, audit forms, utility invoices, etc. Any system that uses multiple sources of documents. The system may also apply extraction of data in these regions, along with adapting and carrying forward historically human defined rules, to fully automate extraction in the new document.

[0019] The system is document agnostic in that the system provides a fuzzy mapping between historical document extractions and new documents. For near identical documents, the system may provide perfect fidelity in extraction. For a new document that is changed from an historical document, the fidelity of the approach is entirely focused on the ability to accurately determine the mapping between extractions from the historical document and corresponding regions of interest in the new document. However, due to multiple steps of verification, if a match cannot be made, regions of interest cannot be found or extracted data is different (e.g., different in expected format or type), the system can default to human extraction. Traditional templated OCR systems include human verification for uncertain values. This system is better than templated OCR because this system can handle document types that have traditionally defied template-based OCR. For example, with the automation of property tax documents, a template-based system would require the creation and maintenance of 3000+ variations, one for each of the numerous US counties.

[0020] More specifically, in various embodiments, the system may create a database of historical documents by analyzing each of the historical documents. The historical documents may be analyzed by acquiring regions of interest (e.g., coordinates) of certain fields, logos, shapes, lines, labels or other objects, obtaining target data types and/or using known key-value pairs. In various embodiments, such factors are initially obtained in a user-driven process that results in annotated documents, as further described below. The historical documents may also be analyzed or annotated by a human user, but the human user only needs to annotate the historical document once, then the historical document can be used many times to compare to various new documents.

[0021] While other systems may try to determine the type of new document by comparing to a limited set of known document types, this system creates a database of many historical documents that are used to identify a new document. For example, each county in the U.S. may include a different property tax form, so it would be almost impossible to have a database of every type of property tax form. Thus, this system compares the new property tax form from this year to historical property tax forms in prior years that may have included the same format. For example, a taxpayer that previously submitted the prior year property tax form may live in the same county as a different taxpayer that submitted the new property tax form this year, so the system may recognize the new property tax form as matching the his-

torical property tax form from last year. As such, the system may compare forms from the same taxpayer or from different taxpayers.

[0022] Moreover, the system goes beyond just identifying tax documents because the system identifies particular types of tax documents. For example, the system may not only recognize that an historical document is a W-2 tax document, but the system may also recognize that the historical document is a W-2 tax document from a particular employer (e.g., a Burger King W-2) and/or for a particular employee.

[0023] The system receives a new document. The new document may be in the form of an electronic document such as, for example, an image, a photo, a scan, spreadsheets (e.g., MS Excel), written documents (e.g., MS Word or emails) and/or a portable document format (PDF). The system may load the new document for comparison against an historical text corpus of the historical documents. The new documents may be processed by an automated back-end workflow. In various embodiments, an exemplary workflow may include multiple steps of autonomous functionality that ingests documents, processes documents, performs data transforms, and delivers the output. In the course of this processing, the new document is automatically sent for storage to a file system (e.g., a cloud based system) and/or may be sent to one or more systems for further processing. The new document is compared against the historical documents. Historical documents may be stored in raw text/PDF formats. However, to improve speed of computation, cached metadata about the historical document may be stored in a relational database. To optimize computation, the first time an historical document is chosen as a candidate for comparison, the extraction algorithm extracts and stores the information that is contained in the historical document such as, for example, text, key-value pairs, named entities, etc. If the historical document is used in future comparisons, the extracted information can be used directly for comparison rather than needing to repeat the extraction process each time.

[0024] The new document may be compared against a subset or all of the candidate historical documents in a corpus. For example, the system may determine a subset by selecting only the historical documents that include a certain taxpayer's name. In that regard, by reducing the set of historical documents to a subset or by not needing to include a large set of all existing types of forms, the system is able to reduce processing time and conserve processing resources. Moreover, selecting a subset of historical documents also increases the accuracy of the results because the subset includes more relevant historical documents (e.g., only documents from the same taxpayer). While the process may operate on any computer, due to the computational power needed, the process may scale to high end server/cloud computing. Additionally, in various embodiments, many of the algorithms may be accelerated further via the use of Graphics Processing Units (GPUs).

[0025] The comparing may be conducted by any process such as, for example, being based on the location un-aware OCR detected text on the new document and/or visual document similarity indexing. With respect to the location un-aware OCR detected text, the image may be pre-processed to remove scanning noise. The pre-processing may include fixing any rotation/skew that may have been applied to the document, and using high and low pass filters to remove digital noise in the brightness/contrast of the page. In comparison to traditional templated OCR approaches, the software then scans a document in search of sequences of characters in a specific language subset (e.g., English) to form words. The detected characters, words, and their exact X, Y location in the document are then extracted and stored or processed, as further described below. Alternative or in tandem to the OCR approach, the process may look at the pixel information contained in a PDF to find image elements (e.g., motifs) that can be compared, as further described below.

[0026] The system may also conduct the comparison by using identification algorithms such as, for example, machine learning, artificial intelligence, expert systems logic, key-value pair matching (e.g., matching labels next to certain data), bag-of-words and/or identifier schema (e.g., matching W-2 from a particular employer).

[0027] More specifically, with respect to machine learning, the system may use specifically supervised learning to compute the probability of two pages being similar or the same. The document may be decomposed into features, which could be any content elements, specific words, visual logos or motifs, the presence of specific fonts, absence or presence of text at different coordinates, or any other element that can be observed from the document. These features may then be used to predict an outcome such as similarity probability. A model can be trained by looking at historical pairs of documents using those features to predict the outcome.

[0028] With respect to artificial intelligence/expert systems logic, the artificial intelligence approaches (e.g., expert systems) uses a long list of IF/THEN logic based upon business rules unique to documents. For example, IF the headers of two documents match, AND there is a unique identifier on both documents, they are the same document. With respect to key/value pair matching, an initial scan is completed on the document to determine headers (keys) and corresponding values (field data). For example, on a form, there could be the word "Name:" in bold as a key, next to the value "John" in regular text. This pattern could be repeated for other keys (address, zip code, etc.) throughout the form. Recognizing this pattern, and then extracting data in this way allows for every document to be recomposed into its corresponding key/value pairs. To determine document matches, the system may compare the overlap between keys in two documents, values in two documents, or a combination.

[0029] With respect to bag of words, documents are scanned for any text. The overlap between all text extracted between two documents is then compared. With respect to identifier schema, to uniquely identify a document, a set of data points (identifiers) are defined in a data dictionary. Depending on the document type, up to five identifiers may be used to uniquely identify a document, although there is no technical limit to the number of identifiers that can be defined. An example of identifier usage would be a W-2 document that includes two identifiers: an employer ID (EIN) and an employee ID (SSN).

[0030] As set forth in FIG. **1**, in various embodiments, the system determines a match score based on the comparison. The match score may be based on the percentage of content that may match between the historical documents and the new document. The content may include, for example, words, characters, headers, numbers, values, symbols, data, information, metadata, regions of interest, visual elements,

textual elements, logos, fonts, lines, layouts, headers, values, tables or other content. Percentage overlap may be determined by taking the Jaccardian distance between the two documents. In particular, the number of words that the documents share divided by the total number of words between both documents. As used herein, the term "match" may include identical match, substantial match, items being similar, match to a certain threshold, match based on an algorithm and/or the like. The system may also incorporate any other match metrics such as, for example, computer vision score, etc.

[0031] As set forth in FIG. **2**, in various embodiments, the system may conduct non-templated text comparison. The system may detect a subset or all of the raw text on the new document. Based upon proximity and formatting, the headers and text values are pulled from this new document raw text by, for example, implementing deep-learning, rule or natural language based relation extraction algorithms (e.g., non-exhaustive survey can be found at www.arxiv.org/pdf/2011.13534.pdf, which is hereby incorporated by reference). In addition, the system may identify any tables in the new document. The system may place the raw text from the new document in its context as well. Any text that is not part of a table or identified link to a header may be referred to as raw text. Raw text may be noted by its X, Y location on the document, and may be fed through an NLP algorithm for named entity recognition to see if it can be determined to be of a specific known entity (Dollar amount, Social Security Number, Address, etc.). The system may detect a subset or all of the raw text on the historical document. Based upon proximity and formatting, the headers and text values are pulled from this historical document raw text, as set forth above. In addition, the system may identify any tables in the historical document. The system may place the raw text from the historical document in their context as well, as set forth above. The system may conduct pairwise comparisons between similar regions of interest in the new and historical documents to identify areas of overlap and difference. Comparisons may be conducted by matching headers and corresponding text values between both documents. If headers cannot be found, then X,Y locations of nearest text is found between documents and comparisons are made. In addition, the entity of the text, if known, may be used to make equal comparisons in these cases as well. Such a process may be algorithmic.

[0032] In various embodiments, the system may conduct non-templated machine vision comparison (e.g., visual document similarity indexing). The system may attempt to match visual elements in the new document to visual elements in a subset or all of the historical documents in the corpus. For example, the logos may be similar, the layout may be similar and/or the font may be similar. The system may capture the result of these matches holistically via, for example, mean square error (MSE), a structural similarity index measure (SSIM), and/or an image similarity metric as determined by a deep neural network. Each of these algorithms may be applied to determine a match by determining a metric score for each metric. For MSE, this is obtained by going through each corresponding pixel of each image and calculating the difference between grayscale pixel intensities in each image. The system squares those differences, adds them, and divides by total number of pixels in the image. For SSIM, the system applies the method by Wang, et. al (www.cns.nyu.edu/pub/cero/wang03-reprint.pdf, which is hereby incorporated by reference). For deep learning, the system may train a decision network on known examples of matching and non-matching images to give a similarity score between any two images or may make use of "Siamese" networks or their variants. (www.proceedings.neurips.cc/paper/1993/file/288cc0ff022877bd3df94bc9360b9c5d-Paper.pdf). All of these approaches may output a score metric.

[0033] If the system finds an historical document or subset of historical documents that match the new document, in various embodiments, the automated system continues the matching process. The match may be based on a certain threshold percentage of matching items. If the system does not find a match, the system determines that a user should be consulted, so the system may notify a user or send the new document to a user. The user may then annotate the new document without using the automation features such that the system may more easily recognize the document. This is a user-interface-driven process in which the end-user identifies one or more data-points on a document and then inputs a value for each of those data points. For example, the end-user may identify an item on a document to be a property-tax payment, and then enter the amount of that payment. The user may annotate the new document by using a mouse to select and identify specific regions on the document and a keyboard to enter corresponding values, with the result being organized, labeled, tabular data extraction. This annotation process may provide the foundation and substrate for future use of the above automation process. Moreover, the system may still perform the above automated features on any additional new documents that are subsequently received by the system. As such, the system or the user may annotate or recognize regions of interest of the new document such that, in the next year, the new document is now a candidate historical document that is already annotated and can be more easily identified.

[0034] As set forth in FIG. **3**, in various embodiments, the system may also verify the match. The system may verify the match by understanding the statistical properties of the match of any particular new document compared to the historical document corpus population statistics. To determine a match, these metrics are applied to all possible matches. The scores are then compiled as a population, and a positive metric outlier is attempted to be detected. There should be two centers of mass in the distribution, one of non-matches and one outlier which is the true match. This can be completed on known match distribution to help "tune" the accuracy of what metric is determined to be a match. For example, historically, the system may know that any metric about "0.9" is 100% a match with this data. In that case, the system may make the match threshold for a document anything that is 0.9 or higher. The system may also use matching algorithms such as, for example, machine learning, artificial intelligence, expert systems logic, key-value pair matching, bag-of-words and/or identifier schema (as explained in more detail above).

[0035] In response to matching the new document with the historical document, as set forth in FIG. **4**, in various embodiments, the system may extract the data from the historical document. The data is not just acquired, but the data is also classified such that the data has more meaning. All or any subset of data points in the system may be classified by way of a comprehensive data dictionary which describes all relevant features of any given data point. The

system may map extractions from a region of interest in the historical document to regions of interest in the new document. The system may detect a region of interest in the new document. For example, the system may compare text headers, text values, XY coordinate positioning, visual elements, target data types and/or known key-value matching between the new document and the historical document to automatically match corresponding regions of interest between the new document and the historical document. With respect to text headers, many regions are identified by a descriptive label (e.g., "Box 12"). With respect to XY coordinates, for documents with fixed formats year-to-year, prior-year XY coordinates can be used to identify the corresponding current-year region. With respect to visual elements, items such as logos, lines, boxes and other geometric shapes can indicate regions of interest via proximity. With respect to key-value pairs, when the keys of key-value pairs can be matched between prior-year and current-year documents, the corresponding XY coordinates of the key-value pair in the prior year can determine the region of interest for the current year. The system may use the detected regions of interest in the new document to establish the new document as a new, dynamic OCR template, such that the system can extract data from the new document using the template as a guide to find the specific data. More specifically, for each comparing (or mapping) between the historical document and the new document, in the new document, OCR is performed on the defined regions identified in the mapping. For example, if there was a mapping in the upper left of a document that found the field "Name", and it was determined that area of interest is now in the lower right of the new document, when the field "Name" is needed in the new document, it will be extracted in the lower right but treated as equal to the historical field.

[0036] In various embodiments, the system may conduct data verification. The system may verify the extracted data by automatically identifying and comparing the entry in the new document (name, address, SSN, currency, number, etc.) to the entry extracted from the historical document. For improving the data accuracy, the system may also use accuracy algorithms such as, for example, machine learning, artificial intelligence, expert systems logic, fuzzy text matching and/or text layer extraction. These algorithms may be domain specific. For example, in accounting, one may have prior knowledge that certain numbers on a form are related (sum of a column being a total, differences being taken, etc.) and this logic can be applied to verify that extracted data is accurate. Similarly, machine learning or fuzzy logic can confirm that values are within expected ranges. For example, someone making $80,000 a year is unlikely to have $2,000,000 in stock dividends and that data should be flagged for review. In fact, the exact numbers and ranges for flagging can be found using population statistics from known forms. Text layer extraction may be a special case, where the text layer of the PDF (if it exists) is extracted to see if it is in concordance with the scanned value. For more information related to text layer extraction, see U.S. patent application Ser. No. 16/047,346, entitled "System and Method for Automatic Detection and Verification of Optical Character Recognition Data," filed Jul. 27, 2018 and U.S. patent application Ser. No. 15/922,821 entitled "System and Method for Automatic Detection and Verification of Optical Character Recognition Data," filed Mar. 15, 2018, the contents of which are herein incorporated by reference in their entirety for all purposes. If the system determines that the extracted data from the historical document does not match the data in the new document, the system determines that a user should be consulted. For example, the system may determine that the extracted data from the historical document does not match the data in the new document because headers do not match, entity type does not match, key known elements from the document are missing (as discovered via an expert systems/explicit check). The system may pass the extracted data to the user for verification. In response to the system or user verifying the extracted data, in various embodiments, the system may categorize the extracted, auto-verified data in the same way in the new document as it was classified in the historical document. As mentioned above, all or any subset of data points in the system may be classified by way of a comprehensive data dictionary which describes all relevant features of any given data point. The system may run the categorized data through various processes. For example, in the context specifically of tax automation, the categorized data from the new document may be automatically processed by tax preparation software. In the realm of invoice automation, whatever actions were taken on an historical invoice (or data that was captured and transferred to other systems), can be repeated on new documents.

[0037] The system may review the new document and historical document within overlapping windows of a graphical user interface. The new document may be within a first window and the historical document with historical extractions in a second window that overlaps with the first window. The system may dynamically relocate the textual information and/or historical extractions within the first window displayed within the graphical user interface based upon a detected overlap condition. The relocation may be based upon visual elements (logos, fonts, lines, and layout), and textual elements (the text contained in headers, values, and tables on the document). When the windows overlap, textual information and/or historical extractions are reformatted and relocated to an unobscured portion of the underlying second window. The system may determine that the textual information and/or historical extractions are too large to fit in the unobstructed portion of the underlying second window. In response to being too large, the system may scale the textual information and/or historical extractions based upon a calculated scaling factor. The system may then automatically relocate the scaled textual information and/or historical extractions to the unobstructed portion of the underlying second window. When the first and second windows no longer overlap, the textual information and/or historical extractions are returned to its original format and location, and un-scaled if needed. The detailed description of various embodiments herein makes reference to the accompanying drawings, which show various embodiments by way of illustration. While these various embodiments are described in sufficient detail to enable those skilled in the art to practice the disclosure, it should be understood that other embodiments may be realized and that logical and mechanical changes may be made without departing from the spirit and scope of the disclosure. Thus, the detailed description herein is presented for purposes of illustration only and not of limitation. For example, the steps recited in any of the method or process descriptions may be executed in any order and are not limited to the order presented. Moreover, any of the functions or steps may be outsourced to or

performed by one or more third parties. Modifications, additions, or omissions may be made to the systems, apparatuses, and methods described herein without departing from the scope of the disclosure. For example, the components of the systems and apparatuses may be integrated or separated. Moreover, the operations of the systems and apparatuses disclosed herein may be performed by more, fewer, or other components and the methods described may include more, fewer, or other steps. Additionally, steps may be performed in any suitable order. As used in this document, "each" refers to each member of a set or each member of a subset of a set. Furthermore, any reference to singular includes plural embodiments, and any reference to more than one component may include a singular embodiment. Although specific advantages have been enumerated herein, various embodiments may include some, none, or all of the enumerated advantages.

[0038] In the detailed description herein, references to "various embodiments," "one embodiment," "an embodiment," "an example embodiment," etc., indicate that the embodiment described may include a particular feature, structure, or characteristic, but every embodiment may not necessarily include the particular feature, structure, or characteristic. Moreover, such phrases are not necessarily referring to the same embodiment. Further, when a particular feature, structure, or characteristic is described in connection with an embodiment, it is submitted that it is within the knowledge of one skilled in the art to affect such feature, structure, or characteristic in connection with other embodiments whether or not explicitly described. After reading the description, it will be apparent to one skilled in the relevant art(s) how to implement the disclosure in alternative embodiments.

[0039] Benefits, other advantages, and solutions to problems have been described herein with regard to specific embodiments. However, the benefits, advantages, solutions to problems, and any elements that may cause any benefit, advantage, or solution to occur or become more pronounced are not to be construed as critical, required, or essential features or elements of the disclosure. The scope of the disclosure is accordingly limited by nothing other than the appended claims, in which reference to an element in the singular is not intended to mean "one and only one" unless explicitly so stated, but rather "one or more." Moreover, where a phrase similar to 'at least one of A, B, and C' or 'at least one of A, B, or C' is used in the claims or specification, it is intended that the phrase be interpreted to mean that A alone may be present in an embodiment, B alone may be present in an embodiment, C alone may be present in an embodiment, or that any combination of the elements A, B and C may be present in a single embodiment; for example, A and B, A and C, B and C, or A and B and C. Although the disclosure includes a method, it is contemplated that it may be embodied as computer program instructions on a tangible computer-readable carrier, such as a magnetic or optical memory or a magnetic or optical disk. All structural, chemical, and functional equivalents to the elements of the above-described various embodiments that are known to those of ordinary skill in the art are expressly incorporated herein by reference and are intended to be encompass 140ed by the present claims. Moreover, it is not necessary for a device or method to address each and every problem sought to be solved by the present disclosure, for it to be encompass 140ed by the present claims. Furthermore, no clement, component, or method step in the present disclosure is intended to be dedicated to the public regardless of whether the element, component, or method step is explicitly recited in the claims. No claim element is intended to invoke 35 U.S.C. § 112(f) unless the element is expressly recited using the phrase "means for" or "step for". As used herein, the terms "comprises," "comprising," or any other variation thereof, are intended to cover a non-exclusive inclusion, such that a process, method, article, or apparatus that comprises a list of elements does not include only those elements but may include other elements not expressly listed or inherent to such process, method, article, or apparatus.

[0040] Terms and phrases similar to "associate" and/or "associating" may include tagging, flagging, correlating, using a look-up table or any other method or system for indicating or creating a relationship between elements, such as, for example, (i) a transaction account and (ii) an item (e.g., offer, reward, discount) and/or digital channel. Moreover, the associating may occur at any point, in response to any suitable action, event, or period of time. The associating may occur at pre-determined intervals, periodically, randomly, once, more than once, or in response to a suitable request or action. Any of the information may be distributed and/or accessed via a software enabled link, wherein the link may be sent via an email, text, post, social network input, and/or any other method known in the art.

[0041] Computer programs (also referred to as computer control logic) are stored in main memory and/or secondary memory. Computer programs may also be received via communications interface. Such computer programs, when executed, enable the computer system to perform the features as discussed herein. In particular, the computer programs, when executed, enable the processor to perform the features of various embodiments. Accordingly, such computer programs represent controllers of the computer system.

[0042] These computer program instructions may be loaded onto a general purpose computer, special purpose computer, or other programmable data processing apparatus to produce a machine, such that the instructions that execute on the computer or other programmable data processing apparatus create means for implementing the functions specified in the flowchart block or blocks. These computer program instructions may also be stored in a computer-readable memory that can direct a computer or other programmable data processing apparatus to function in a particular manner, such that the instructions stored in the computer-readable memory produce an article of manufacture including instruction means which implement the function specified in the flowchart block or blocks. The computer program instructions may also be loaded onto a computer or other programmable data processing apparatus to cause a series of operational steps to be performed on the computer or other programmable apparatus to produce a computer-implemented process such that the instructions which execute on the computer or other programmable apparatus provide steps for implementing the functions specified in the flowchart block or blocks.

[0043] In various embodiments, software may be stored in a computer program product and loaded into a computer system using a removable storage drive, hard disk drive, or communications interface. The control logic (software), when executed by the processor, causes the processor to perform the functions of various embodiments as described

herein. In various embodiments, hardware components may take the form of application specific integrated circuits (ASICs). Implementation of the hardware so as to perform the functions described herein will be apparent to persons skilled in the relevant art(s).

[0044] As will be appreciated by one of ordinary skill in the art, the system may be embodied as a customization of an existing system, an add-on product, a processing apparatus executing upgraded software, a stand-alone system, a distributed system, a method, a data processing system, a device for data processing, and/or a computer program product. Accordingly, any portion of the system or a module may take the form of a processing apparatus executing code, an internet based embodiment, an entirely hardware embodiment, or an embodiment combining aspects of the internet, software, and hardware. Furthermore, the system may take the form of a computer program product on a computer-readable storage medium having computer-readable program code means embodied in the storage medium. Any suitable computer-readable storage medium may be utilized, including hard disks, CD-ROM, BLU-RAY DISC®, optical storage devices, magnetic storage devices, and/or the like.

[0045] In various embodiments, components, modules, and/or engines of system **100** may be implemented as micro-applications or micro-apps. Micro-apps are typically deployed in the context of a mobile operating system, including for example, a WINDOWS® mobile operating system, an ANDROID® operating system, an APPLE® iOS operating system, a BLACKBERRY® company's operating system, and the like. The micro-app may be configured to leverage the resources of the larger operating system and associated hardware via a set of predetermined rules which govern the operations of various operating systems and hardware resources. For example, where a micro-app desires to communicate with a device or network other than the mobile device or mobile operating system, the micro-app may leverage the communication protocol of the operating system and associated device hardware under the predetermined rules of the mobile operating system. Moreover, where the micro-app desires an input from a user, the micro-app may be configured to request a response from the operating system which monitors various hardware components and then communicates a detected input from the hardware to the micro-app.

[0046] The system and method may be described herein in terms of functional block components, screen shots, optional selections, and various processing steps. It should be appreciated that such functional blocks may be realized by any number of hardware and/or software components configured to perform the specified functions. For example, the system may employ various integrated circuit components, e.g., memory elements, processing elements, logic elements, look-up tables, and the like, which may carry out a variety of functions under the control of one or more microprocessors or other control devices. Similarly, the software elements of the system may be implemented with any programming or scripting language such as C, C++, C#, JAVA®, JAVASCRIPT®, JAVASCRIPT® Object Notation (JSON), VBScript, Macromedia COLD FUSION, COBOL, MICROSOFT® company's Active Server Pages, assembly, PERL®, PHP, awk, PYTHON®, Visual Basic, SQL Stored Procedures, PL/SQL, any UNIX® shell script, and extensible markup language (XML) with the various algorithms being implemented with any combination of data structures, objects, processes, routines or other programming elements. Further, it should be noted that the system may employ any number of conventional techniques for data transmission, signaling, data processing, network control, and the like. Still further, the system could be used to detect or prevent security issues with a client-side scripting language, such as JAVASCRIPT®, VBScript, or the like.

[0047] The system and method are described herein with reference to screen shots, block diagrams and flowchart illustrations of methods, apparatus, and computer program products according to various embodiments. It will be understood that each functional block of the block diagrams and the flowchart illustrations, and combinations of functional blocks in the block diagrams and flowchart illustrations, respectively, can be implemented by computer program instructions.

[0048] Accordingly, functional blocks of the block diagrams and flowchart illustrations support combinations of means for performing the specified functions, combinations of steps for performing the specified functions, and program instruction means for performing the specified functions. It will also be understood that each functional block of the block diagrams and flowchart illustrations, and combinations of functional blocks in the block diagrams and flowchart illustrations, can be implemented by either special purpose hardware-based computer systems which perform the specified functions or steps, or suitable combinations of special purpose hardware and computer instructions. Further, illustrations of the process flows and the descriptions thereof may make reference to user WINDOWS® applications, webpages, websites, web forms, prompts, etc. Practitioners will appreciate that the illustrated steps described herein may comprise, in any number of configurations, including the use of WINDOWS® applications, webpages, web forms, popup WINDOWS® applications, prompts, and the like. It should be further appreciated that the multiple steps as illustrated and described may be combined into single webpages and/or WINDOWS® applications but have been expanded for the sake of simplicity. In other cases, steps illustrated and described as single process steps may be separated into multiple webpages and/or WINDOWS® applications but have been combined for simplicity.

[0049] In various embodiments, the software elements of the system may also be implemented using a JAVASCRIPT® run-time environment configured to execute JAVASCRIPT® code outside of a web browser. For example, the software elements of the system may also be implemented using NODE.JS® components. NODE.JS® programs may implement several modules to handle various core functionalities. For example, a package management module, such as NPM®, may be implemented as an open source library to aid in organizing the installation and management of third-party NODE.JS® programs. NODE.JS® programs may also implement a process manager, such as, for example, Parallel Multithreaded Machine ("PM2"); a resource and performance monitoring tool, such as, for example, Node Application Metrics ("appmetrics"); a library module for building user interfaces, and/or any other suitable and/or desired module.

[0050] Middleware may include any hardware and/or software suitably configured to facilitate communications and/or process transactions between disparate computing systems. Middleware components are commercially available and known in the art. Middleware may be implemented

through commercially available hardware and/or software, through custom hardware and/or software components, or through a combination thereof. Middleware may reside in a variety of configurations and may exist as a standalone system or may be a software component residing on the internet server. Middleware may be configured to process transactions between the various components of an application server and any number of internal or external systems for any of the purposes disclosed herein. WEBSPHERE® MQTM (formerly MQSeries) by IBM®, Inc. (Armonk, NY) is an example of a commercially available middleware product. An Enterprise Service Bus ("ESB") application is another example of middleware.

[0051] The computers discussed herein may provide a suitable website or other internet-based graphical user interface which is accessible by users. In one embodiment, MICROSOFT® company's Internet Information Services (IIS), Transaction Server (MTS) service, and an SQL SERVER® database, are used in conjunction with MICROSOFT® operating systems, WINDOWS NT® web server software, SQL SERVER® database, and MICROSOFT® Commerce Server. Additionally, components such as ACCESS® software, SQL SERVER® database, ORACLE® software, SYBASE® software, INFORMIX® software, MYSQL® software, INTERBASE® software, etc., may be used to provide an Active Data Object (ADO) compliant database management system. In one embodiment, the APACHE® web server is used in conjunction with a LINUX® operating system, a MYSQL® database, and PERL®, PHP, Ruby, and/or PYTHON® programming languages.

[0052] For the sake of brevity, conventional data networking, application development, and other functional aspects of the systems (and components of the individual operating components of the systems) may not be described in detail herein. Furthermore, the connecting lines shown in the various figures contained herein are intended to represent exemplary functional relationships and/or physical couplings between the various elements. It should be noted that many alternative or additional functional relationships or physical connections may be present in a practical system.

[0053] In various embodiments, the methods described herein are implemented using the various particular machines described herein. The methods described herein may be implemented using the below particular machines, and those hereinafter developed, in any suitable combination, as would be appreciated immediately by one skilled in the art. Further, as is unambiguous from this disclosure, the methods described herein may result in various transformations of certain articles.

[0054] In various embodiments, the system and various components may integrate with one or more smart digital assistant technologies. For example, exemplary smart digital assistant technologies may include the ALEXA® system developed by the AMAZON® company, the GOOGLE HOME® system developed by Alphabet, Inc., the HOMEPOD® system of the APPLE® company, and/or similar digital assistant technologies. The ALEXA® system, GOOGLE HOME® system, and HOMEPOD® system, may each provide cloud-based voice activation services that can assist with tasks, entertainment, general information, and more. All the ALEXA® devices, such as the AMAZON ECHO®, AMAZON ECHO DOT®, AMAZON TAP®, and AMAZON FIRE® TV, have access to the ALEXA® system. The ALEXA® system, GOOGLE HOME® system, and HOMEPOD® system may receive voice commands via its voice activation technology, activate other functions, control smart devices, and/or gather information. For example, the smart digital assistant technologies may be used to interact with music, emails, texts, phone calls, question answering, home improvement information, smart home communication/activation, games, shopping, making to-do lists, setting alarms, streaming podcasts, playing audiobooks, and providing weather, traffic, and other real time information, such as news. The ALEXA®, GOOGLE HOME®, and HOMEPOD® systems may also allow the user to access information about eligible transaction accounts linked to an online account across all digital assistant-enabled devices.

[0055] The various system components discussed herein may include one or more of the following: a host server or other computing systems including a processor for processing digital data; a memory coupled to the processor for storing digital data; an input digitizer coupled to the processor for inputting digital data; an application program stored in the memory and accessible by the processor for directing processing of digital data by the processor; a display device coupled to the processor and memory for displaying information derived from digital data processed by the processor; and a plurality of databases. Various databases used herein may include: client data; merchant data; financial institution data; and/or like data useful in the operation of the system. As those skilled in the art will appreciate, user computer may include an operating system (e.g., WINDOWS®, UNIX®, LINUX®, SOLARIS®, MACOS®, etc.) as well as various conventional support software and drivers typically associated with computers.

[0056] The present system or any part(s) or function(s) thereof may be implemented using hardware, software, or a combination thereof and may be implemented in one or more computer systems or other processing systems. However, the manipulations performed by embodiments may be referred to in terms, such as matching or selecting, which are commonly associated with mental operations performed by a human operator. No such capability of a human operator is necessary, or desirable, in most cases, in any of the operations described herein. Rather, the operations may be machine operations or any of the operations may be conducted or enhanced by artificial intelligence (AI) or machine learning. Al may refer generally to the study of agents (e.g., machines, computer-based systems, etc.) that perceive the world around them, form plans, and make decisions to achieve their goals. Foundations of Al include mathematics, logic, philosophy, probability, linguistics, neuroscience, and decision theory. Many fields fall under the umbrella of AI, such as computer vision, robotics, machine learning, and natural language processing. Useful machines for performing the various embodiments include general purpose digital computers or similar devices.

[0057] In various embodiments, the embodiments are directed toward one or more computer systems capable of carrying out the functionalities described herein. The computer system includes one or more processors. The processor is connected to a communication infrastructure (e.g., a communications bus, cross-over bar, network, etc.). Various software embodiments are described in terms of this exemplary computer system. After reading this description, it will become apparent to a person skilled in the relevant art(s) how to implement various embodiments using other com-

puter systems and/or architectures. The computer system can include a display interface that forwards graphics, text, and other data from the communication infrastructure (or from a frame buffer not shown) for display on a display unit.

[0058] The computer system also includes a main memory, such as random access memory (RAM), and may also include a secondary memory. The secondary memory may include, for example, a hard disk drive, a solid-state drive, and/or a removable storage drive. The removable storage drive reads from and/or writes to a removable storage unit in a well-known manner. As will be appreciated, the removable storage unit includes a computer usable storage medium having stored therein computer software and/or data.

[0059] In various embodiments, secondary memory may include other similar devices for allowing computer programs or other instructions to be loaded into a computer system. Such devices may include, for example, a removable storage unit and an interface. Examples of such may include a program cartridge and cartridge interface (such as that found in video game devices), a removable memory chip (such as an erasable programmable read only memory (EPROM), programmable read only memory (PROM)) and associated socket, or other removable storage units and interfaces, which allow software and data to be transferred from the removable storage unit to a computer system.

[0060] The terms "computer program medium," "computer usable medium," and "computer readable medium" are used to generally refer to media such as removable storage drive and a hard disk installed in hard disk drive. These computer program products provide software to a computer system.

[0061] The computer system may also include a communications interface. A communications interface allows software and data to be transferred between the computer system and external devices. Examples of such a communications interface may include a modem, a network interface (such as an Ethernet card), a communications port, etc. Software and data transferred via the communications interface are in the form of signals which may be electronic, electromagnetic, optical, or other signals capable of being received by communications interface. These signals are provided to communications interface via a communications path (e.g., channel). This channel carries signals and may be implemented using wire, cable, fiber optics, a telephone line, a cellular link, a radio frequency (RF) link, wireless and other communications channels.

[0062] As used herein an "identifier" may be any suitable identifier that uniquely identifies an item. For example, the identifier may be a globally unique identifier ("GUID"). The GUID may be an identifier created and/or implemented under the universally unique identifier standard. Moreover, the GUID may be stored as 128-bit value that can be displayed as 32 hexadecimal digits. The identifier may also include a major number, and a minor number. The major number and minor number may each be 16-bit integers.

[0063] The firewall may include any hardware and/or software suitably configured to protect CMS components and/or enterprise computing resources from users of other networks. Further, a firewall may be configured to limit or restrict access to various systems and components behind the firewall for web clients connecting through a web server. Firewall may reside in varying configurations including Stateful Inspection, Proxy based, access control lists, and Packet Filtering among others. Firewall may be integrated within a web server or any other CMS components or may further reside as a separate entity. A firewall may implement network address translation ("NAT") and/or network address port translation ("NAPT"). A firewall may accommodate various tunneling protocols to facilitate secure communications, such as those used in virtual private networking. A firewall may implement a demilitarized zone ("DMZ") to facilitate communications with a public network such as the internet. A firewall may be integrated as software within an internet server or any other application server components, reside within another computing device, or take the form of a standalone hardware component.

[0064] Any databases discussed herein may include relational, hierarchical, graphical, blockchain, object-oriented structure, and/or any other database configurations. Any database may also include a flat file structure wherein data may be stored in a single file in the form of rows and columns, with no structure for indexing and no structural relationships between records. For example, a flat file structure may include a delimited text file, a CSV (comma-separated values) file, and/or any other suitable flat file structure. Common database products that may be used to implement the databases include DB2® by IBM® (Armonk, NY), various database products available from ORACLE® Corporation (Redwood Shores, CA), MICROSOFT ACCESS® or MICROSOFT SQL SERVER® by MICROSOFT® Corporation (Redmond, Washington), MYSQL® by MySQL AB (Uppsala, Sweden), MONGODB®, Redis, APACHE CASSANDRA®, HBASE® by APACHE®, MapR-DB by the MAPR® corporation, or any other suitable database product. Moreover, any database may be organized in any suitable manner, for example, as data tables or lookup tables. Each record may be a single file, a series of files, a linked series of data fields, or any other data structure.

[0065] As used herein, big data may refer to partially or fully structured, semi-structured, or unstructured data sets including millions of rows and hundreds of thousands of columns. A big data set may be compiled, for example, from a history of purchase transactions over time, from web registrations, from social media, from records of charge (ROC), from summaries of charges (SOC), from internal data, or from other suitable sources. Big data sets may be compiled without descriptive metadata such as column types, counts, percentiles, or other interpretive-aid data points.

[0066] Association of certain data may be accomplished through any desired data association technique such as those known or practiced in the art. For example, the association may be accomplished either manually or automatically. Automatic association techniques may include, for example, a database search, a database merge, GREP, AGREP, SQL, using a key field in the tables to speed searches, sequential searches through all the tables and files, sorting records in the file according to a known order to simplify lookup, and/or the like. The association step may be accomplished by a database merge function, for example, using a "key field" in pre-selected databases or data sectors. Various database tuning steps are contemplated to optimize database performance. For example, frequently used files such as indexes may be placed on separate file systems to reduce In/Out ("I/O") bottlenecks.

[0067] More particularly, a "key field" partitions the database according to the high-level class of objects defined by the key field. For example, certain types of data may be designated as a key field in a plurality of related data tables and the data tables may then be linked on the basis of the type of data in the key field. The data corresponding to the key field in each of the linked data tables is preferably the same or of the same type. However, data tables having similar, though not identical, data in the key fields may also be linked by using AGREP, for example. In accordance with one embodiment, any suitable data storage technique may be utilized to store data without a standard format. Data sets may be stored using any suitable technique, including, for example, storing individual files using an ISO/IEC 7816-4 file structure; implementing a domain whereby a dedicated file is selected that exposes one or more elementary files containing one or more data sets; using data sets stored in individual files using a hierarchical filing system; data sets stored as records in a single file (including compression, SQL accessible, hashed via one or more keys, numeric, alphabetical by first tuple, etc.); data stored as Binary Large Object (BLOB); data stored as ungrouped data elements encoded using ISO/IEC 7816-6 data elements; data stored as ungrouped data elements encoded using ISO/IEC Abstract Syntax Notation (ASN.1) as in ISO/IEC 8824 and 8825; other proprietary techniques that may include fractal compression methods, image compression methods, etc.

[0068] In various embodiments, the ability to store a wide variety of information in different formats is facilitated by storing the information as a BLOB. Thus, any binary information can be stored in a storage space associated with a data set. As discussed above, the binary information may be stored in association with the system or external to but affiliated with the system. The BLOB method may store data sets as ungrouped data elements formatted as a block of binary via a fixed memory offset using either fixed storage allocation, circular queue techniques, or best practices with respect to memory management (e.g., paged memory, least recently used, etc.). By using BLOB methods, the ability to store various data sets that have different formats facilitates the storage of data, in the database or associated with the system, by multiple and unrelated owners of the data sets. For example, a first data set which may be stored may be provided by a first party, a second data set which may be stored may be provided by an unrelated second party, and yet a third data set which may be stored may be provided by a third party unrelated to the first and second party. Each of these three exemplary data sets may contain different information that is stored using different data storage formats and/or techniques. Further, each data set may contain subsets of data that also may be distinct from other subsets.

[0069] As stated above, in various embodiments, the data can be stored without regard to a common format. However, the data set (e.g., BLOB) may be annotated in a standard manner when provided for manipulating the data in the database or system. The annotation may comprise a short header, trailer, or other appropriate indicator related to each data set that is configured to convey information useful in managing the various data sets. For example, the annotation may be called a "condition header," "header," "trailer," or "status," herein, and may comprise an indication of the status of the data set or may include an identifier correlated to a specific issuer or owner of the data. In one example, the first three bytes of each data set BLOB may be configured or configurable to indicate the status of that particular data set; e.g., LOADED, INITIALIZED, READY, BLOCKED, REMOVABLE, or DELETED. Subsequent bytes of data may be used to indicate for example, the identity of the issuer, user, transaction/membership account identifier or the like. Each of these condition annotations are further discussed herein.

[0070] The data set annotation may also be used for other types of status information as well as various other purposes. For example, the data set annotation may include security information establishing access levels. The access levels may, for example, be configured to permit only certain individuals, levels of employees, companies, or other entities to access data sets, or to permit access to specific data sets based on the transaction, merchant, issuer, user, or the like. Furthermore, the security information may restrict/permit only certain actions, such as accessing, modifying, and/or deleting data sets. In one example, the data set annotation indicates that only the data set owner or the user are permitted to delete a data set, various identified users may be permitted to access the data set for reading, and others are altogether excluded from accessing the data set. However, other access restriction parameters may also be used allowing various entities to access a data set with various permission levels as appropriate.

[0071] The data, including the header or trailer, may be received by a standalone interaction device configured to add, delete, modify, or augment the data in accordance with the header or trailer. As such, in one embodiment, the header or trailer is not stored on the transaction device along with the associated issuer-owned data, but instead the appropriate action may be taken by providing to the user, at the standalone device, the appropriate option for the action to be taken. The system may contemplate a data storage arrangement wherein the header or trailer, or header or trailer history, of the data is stored on the system, device or transaction instrument in relation to the appropriate data.

[0072] One skilled in the art will also appreciate that, for security reasons, any databases, systems, devices, servers, or other components of the system may consist of any combination thereof at a single location or at multiple locations, wherein each database or system includes any of various suitable security features, such as firewalls, access codes, encryption, decryption, compression, decompression, and/or the like.

[0073] Practitioners will also appreciate that there are a number of methods for displaying data within a browser-based document. Data may be represented as standard text or within a fixed list, scrollable list, drop-down list, editable text field, fixed text field, pop-up window, and the like. Likewise, there are a number of methods available for modifying data in a web page such as, for example, free text entry using a keyboard, selection of menu items, check boxes, option boxes, and the like.

[0074] The data may be big data that is processed by a distributed computing cluster. The distributed computing cluster may be, for example, a HADOOP® software cluster configured to process and store big data sets with some of nodes comprising a distributed storage system and some of nodes comprising a distributed processing system. In that regard, distributed computing cluster may be configured to support a HADOOP® software distributed file system (HDFS) as specified by the Apache Software Foundation at www.hadoop.apache.org/docs.

[0075] As used herein, the term "network" includes any cloud, cloud computing system, or electronic communications system or method which incorporates hardware and/or software components. Communication among the parties may be accomplished through any suitable communication channels, such as, for example, a telephone network, an extranet, an intranet, internet, point of interaction device (point of sale device, personal digital assistant (e.g., an IPHONE® device, a BLACKBERRY® device), cellular phone, kiosk, etc.), online communications, satellite communications, off-line communications, wireless communications, transponder communications, local area network (LAN), wide area network (WAN), virtual private network (VPN), networked or linked devices, keyboard, mouse, and/or any suitable communication or data input modality. Moreover, although the system is frequently described herein as being implemented with TCP/IP communications protocols, the system may also be implemented using IPX, APPLETALK® program, IP-6, NetBIOS, OSI, any tunneling protocol (e.g. IPsec, SSH, etc.), or any number of existing or future protocols. If the network is in the nature of a public network, such as the internet, it may be advantageous to presume the network to be insecure and open to eavesdroppers. Specific information related to the protocols, standards, and application software utilized in connection with the internet is generally known to those skilled in the art and, as such, need not be detailed herein.

[0076] "Cloud" or "Cloud computing" includes a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. Cloud computing may include location-independent computing, whereby shared servers provide resources, software, and data to computers and other devices on demand.

[0077] As used herein, "transmit" may include sending electronic data from one system component to another over a network connection. Additionally, as used herein, "data" may include encompassing information such as commands, queries, files, data for storage, and the like in digital or any other form.

[0078] Any database discussed herein may comprise a distributed ledger maintained by a plurality of computing devices (e.g., nodes) over a peer-to-peer network. Each computing device maintains a copy and/or partial copy of the distributed ledger and communicates with one or more other computing devices in the network to validate and write data to the distributed ledger. The distributed ledger may use features and functionality of blockchain technology, including, for example, consensus-based validation, immutability, and cryptographically chained blocks of data. The blockchain may comprise a ledger of interconnected blocks containing data. The blockchain may provide enhanced security because each block may hold individual transactions and the results of any blockchain executables. Each block may link to the previous block and may include a timestamp. Blocks may be linked because each block may include the hash of the prior block in the blockchain. The linked blocks form a chain, with only one successor block allowed to link to one other predecessor block for a single chain. Forks may be possible where divergent chains are established from a previously uniform blockchain, though typically only one of the divergent chains will be maintained as the consensus chain. In various embodiments, the blockchain may implement smart contracts that enforce data workflows in a decentralized manner. The system may also include applications deployed on user devices such as, for example, computers, tablets, smartphones, Internet of Things devices ("IoT" devices), etc. The applications may communicate with the blockchain (e.g., directly or via a blockchain node) to transmit and retrieve data. In various embodiments, a governing organization or consortium may control access to data stored on the blockchain. Registration with the managing organization(s) may enable participation in the blockchain network.

[0079] Data transfers performed through the blockchain-based system may propagate to the connected peers within the blockchain network within a duration that may be determined by the block creation time of the specific blockchain technology implemented. For example, on an ETHEREUM®-based network, a new data entry may become available within about 13-20 seconds as of the writing. On a HYPERLEDGER® Fabric 1.0 based platform, the duration is driven by the specific consensus algorithm that is chosen, and may be performed within seconds. In that respect, propagation times in the system may be improved compared to existing systems, and implementation costs and time to market may also be drastically reduced. The system also offers increased security at least partially due to the immutable nature of data that is stored in the blockchain, reducing the probability of tampering with various data inputs and outputs. Moreover, the system may also offer increased security of data by performing cryptographic processes on the data prior to storing the data on the blockchain. Therefore, by transmitting, storing, and accessing data using the system described herein, the security of the data is improved, which decreases the risk of the computer or network from being compromised.

[0080] In various embodiments, the system may also reduce database synchronization errors by providing a common data structure, thus at least partially improving the integrity of stored data. The system also offers increased reliability and fault tolerance over traditional databases (e.g., relational databases, distributed databases, etc.) as each node operates with a full copy of the stored data, thus at least partially reducing downtime due to localized network outages and hardware failures. The system may also increase the reliability of data transfers in a network environment having reliable and unreliable peers, as each node broadcasts messages to all connected peers, and, as each block comprises a link to a previous block, a node may quickly detect a missing block and propagate a request for the missing block to the other nodes in the blockchain network.

[0081] The particular blockchain implementation described herein provides improvements over conventional technology by using a decentralized database and improved processing environments. In particular, the blockchain implementation improves computer performance by, for example, leveraging decentralized resources (e.g., lower latency). The distributed computational resources improves computer performance by, for example, reducing processing times. Furthermore, the distributed computational resources improves computer performance by improving security using, for example, cryptographic protocols.

[0082] Any communication, transmission, and/or channel discussed herein may include any system or method for delivering content (e.g. data, information, metadata, etc.),

and/or the content itself. The content may be presented in any form or medium, and in various embodiments, the content may be delivered electronically and/or capable of being presented electronically. For example, a channel may comprise a website, mobile application, or device (e.g., FACEBOOK®, YOUTUBE®, PANDORA®, APPLE TV®, MICROSOFT® XBOX®, ROKU®, AMAZON FIRE®, GOOGLE CHROMECAST™, SONY® PLAYSTATION®, NINTENDO® SWITCH®, etc.) a uniform resource locator ("URL"), a document (e.g., a MICROSOFT®) Word or EXCEL™, an ADOBE® Portable Document Format (PDF) document, etc.), an "ebook," an "emagazine," an application or microapplication (as described herein), an short message service (SMS) or other type of text message, an email, a FACEBOOK® message, a TWITTER® tweet, multimedia messaging services (MMS), and/or other type of communication technology. In various embodiments, a channel may be hosted or provided by a data partner. In various embodiments, the distribution channel may comprise at least one of a merchant website, a social media website, affiliate or partner websites, an external vendor, a mobile device communication, social media network, and/or location based service. Distribution channels may include at least one of a merchant website, a social media site, affiliate or partner websites, an external vendor, and a mobile device communication. Examples of social media sites include FACEBOOK®, FOURSQUARE®, TWITTER®, LINKEDIN®, INSTAGRAM®, PINTEREST®, TUMBLR®, REDDIT®, SNAPCHAT®, WHATSAPP®, FLICKR®, VK®, QZONE®, WECHAT®, and the like. Examples of affiliate or partner websites include AMERICAN EXPRESS®, GROUPON®, LIVINGSOCIAL®, and the like. Moreover, examples of mobile device communications include texting, email, and mobile applications for smartphones.

What is claimed is:

1. A method comprising:

determining, by a one or more processors, match metrics based on a percentage of a second content of a new document that matches first content in one or more of a plurality of historical documents by decomposing the first content and the second content into features and using machine learning to predict a similarity probability based on the features;

extracting, by the one or more processors, the second content from regions of interest in the new document based on the match metrics; and

preparing, by the one or more processors, documents using the second content.

2. The method of claim **1**, wherein the features include at least one of content elements, words, visual logos, motifs, fonts, presence of text at different coordinates, or absence of text at different coordinates.

3. The method of claim **1**, further comprising training, by the one or more processors, a model by looking at historical pairs of documents and using the features to predict the outcome.

4. The method of claim **1**, wherein the machine learning includes supervised learning.

5. The method of claim **1**, wherein the processor uses identification algorithms.

6. The method of claim **1**, wherein the processor uses at least one of artificial intelligence, expert systems logic, key-value pair matching, bag-of-words or identifier schema.

7. The method of claim **1**, wherein the processor uses if/then logic based on business rules for at least one of the new document or one or more of the plurality of historical documents.

8. The method of claim **1**, further comprising scanning, by the one or more processors, the new document to determine headers and corresponding values.

9. The method of claim **1**, further comprising comparing, by the one or more processors, at least one of keys or values in the new document and one or more of the plurality of historical documents.

10. A method comprising:

determining, by one or more processors, match metrics based on a percentage of a second content of a new document that matches first content in one or more of a plurality of historical documents by pulling headers and text values from raw text of the second content in the new document, based on at least one of proximity or formatting of raw text of the second content in the new document;

extracting, by the one or more processors, the second content from regions of interest in the new document based on the match metrics; and

preparing, by the one or more processors, documents using the second content.

11. The method of claim **10**, further comprising conducting, by the one or more processors, pairwise comparisons between similar of the regions of interest in the historical document and the new document to identify areas of overlap and areas of difference.

12. The method of claim **10**, wherein the determining match metrics includes non-templated text comparison.

13. The method of claim **10**, further comprising detecting, by the one or more processors, at least a subset of raw text in at least one of the first content of the historical document or the second content of the new document.

14. The method of claim **10**, wherein the pulling the headers and the text values is accomplished by implementing at least one of deep learning, rule based relation extraction algorithms or natural language based relation extraction algorithms.

15. The method of claim **10**, further comprising identifying, by the one or more processors, tables in at least one of the new document or the historical document.

16. The method of claim **10**, further comprising placing, by the one or more processors, at least one of the raw text from the new document in context or the raw text from the historical document in context.

17. The method of claim **10**, wherein raw text includes text that is at least one of not part of a table or does not have an identified link to a header.

18. The method of claim **10**, wherein raw text is determined by x, y location on the new document.

19. The method of claim **10**, further comprising feeding, by the one or more processors, the raw text through an NLP algorithm for named entity recognition to determine if the raw text is of a specific known entity.

20. The method of claim **10**, further comprising pulling, by the one or more processors, headers and text values from the raw text of the first content in the historical document, based on at least one of proximity or formatting of raw text of the first content in the historical document.

* * * * *