US 20250265305A1

(19) **United States**

(12) **Patent Application Publication** (10) **Pub. No.: US 2025/0265305 A1**

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

(54) **METHOD AND SYSTEM FOR WEB DATA EXTRACTION USING META-PATH GRAPH**

(71) Applicant: **S&P Global Inc.**, New York, NY (US)

(72) Inventors: **Saeed Shoaraee**, Burlington (CA); **Akshay Sehgal**, Gurugram (IN); **Mrinal Gupta**, Jammu (IN); **Ankur Debnath**, Hyderabad (IN)

(21) Appl. No.: **18/444,999**

(22) Filed: **Feb. 19, 2024**

Publication Classification

(51) **Int. Cl.**

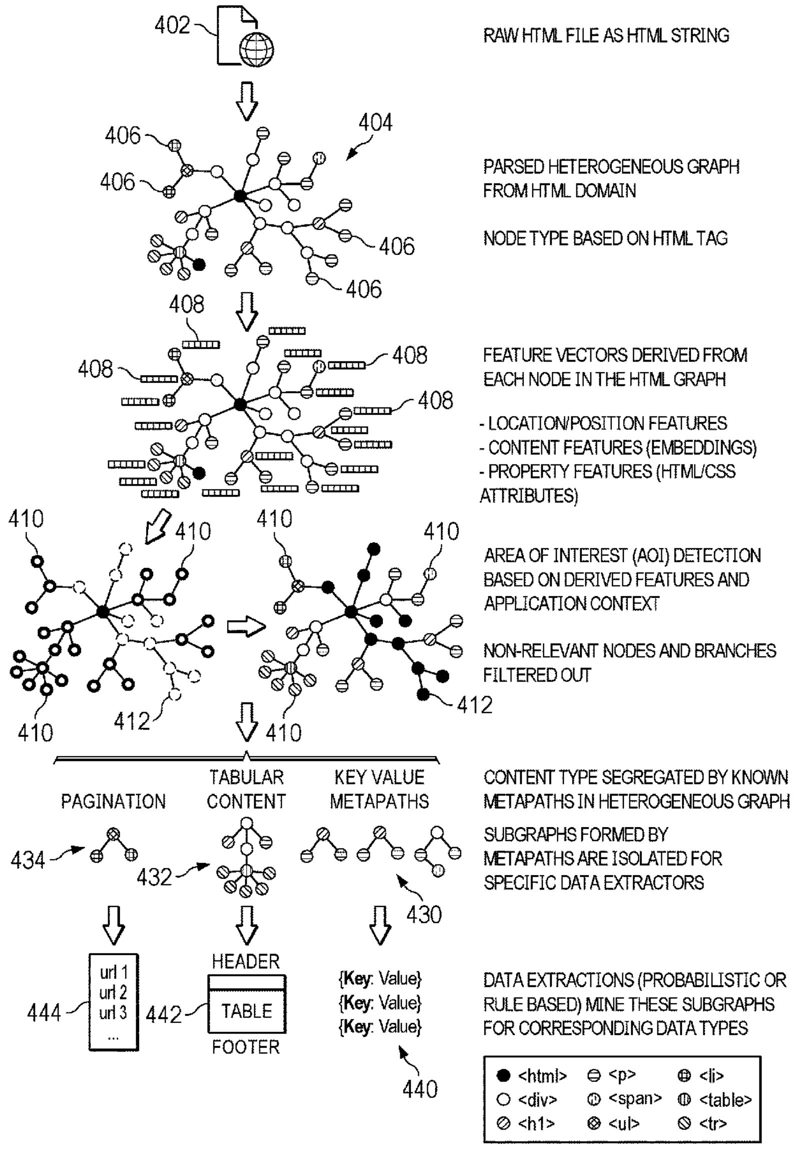
G06F 16/958	(2019.01)
G06F 18/2415	(2023.01)

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

CPC ***G06F 16/986*** (2019.01); ***G06F 18/2415*** (2023.01)

(57) **ABSTRACT**

A computer-implemented method for web data extraction is provided. The method includes receiving an HTML file containing HTML data and converting the HTML data into an HTML graph. Elements in the HTML file are represented by nodes in the HTML graph and relationships among the elements are represented by meta-paths. The method includes generating feature sets for the nodes in the HTML graph and identifying areas of interest in the HTML graph based on the feature sets of the nodes. The method includes refining the identified areas of interest by segregating sub-structures having recurring patterns or sequences. The method includes extracting data items from the segregated sub-structures, storing the extracted data items and monitoring them over time for any significant updates.


402
RAW HTML FILE AS HTML STRING
404
406
406
406
406
PARSED HETEROGENEOUS GRAPH FROM HTML DOMAIN
NODE TYPE BASED ON HTML TAG
408
408
408
408
FEATURE VECTORS DERIVED FROM EACH NODE IN THE HTML GRAPH
- LOCATION/POSITION FEATURES
- CONTENT FEATURES (EMBEDDINGS)
- PROPERTY FEATURES (HTML/CSS ATTRIBUTES)
410
410
410
410
410
410
412
412
AREA OF INTEREST (AOI) DETECTION BASED ON DERIVED FEATURES AND APPLICATION CONTEXT
NON-RELEVANT NODES AND BRANCHES FILTERED OUT
PAGINATION
TABULAR CONTENT
KEY VALUE METAPATHS
CONTENT TYPE SEGREGATED BY KNOWN METAPATHS IN HETEROGENEOUS GRAPH
434
432
430
SUBGRAPHS FORMED BY METAPATHS ARE ISOLATED FOR SPECIFIC DATA EXTRACTORS
url 1
url 2
url 3
...
444
HEADER
TABLE
FOOTER
442
{Key: Value}
{Key: Value}
{Key: Value}
440
DATA EXTRACTIONS (PROBABILISTIC OR RULE BASED) MINE THESE SUBGRAPHS FOR CORRESPONDING DATA TYPES
● <html>
○ <div>
⊘ <h1>
⊖ <p>
<span>
<ul>
<li>
<table>
<tr>

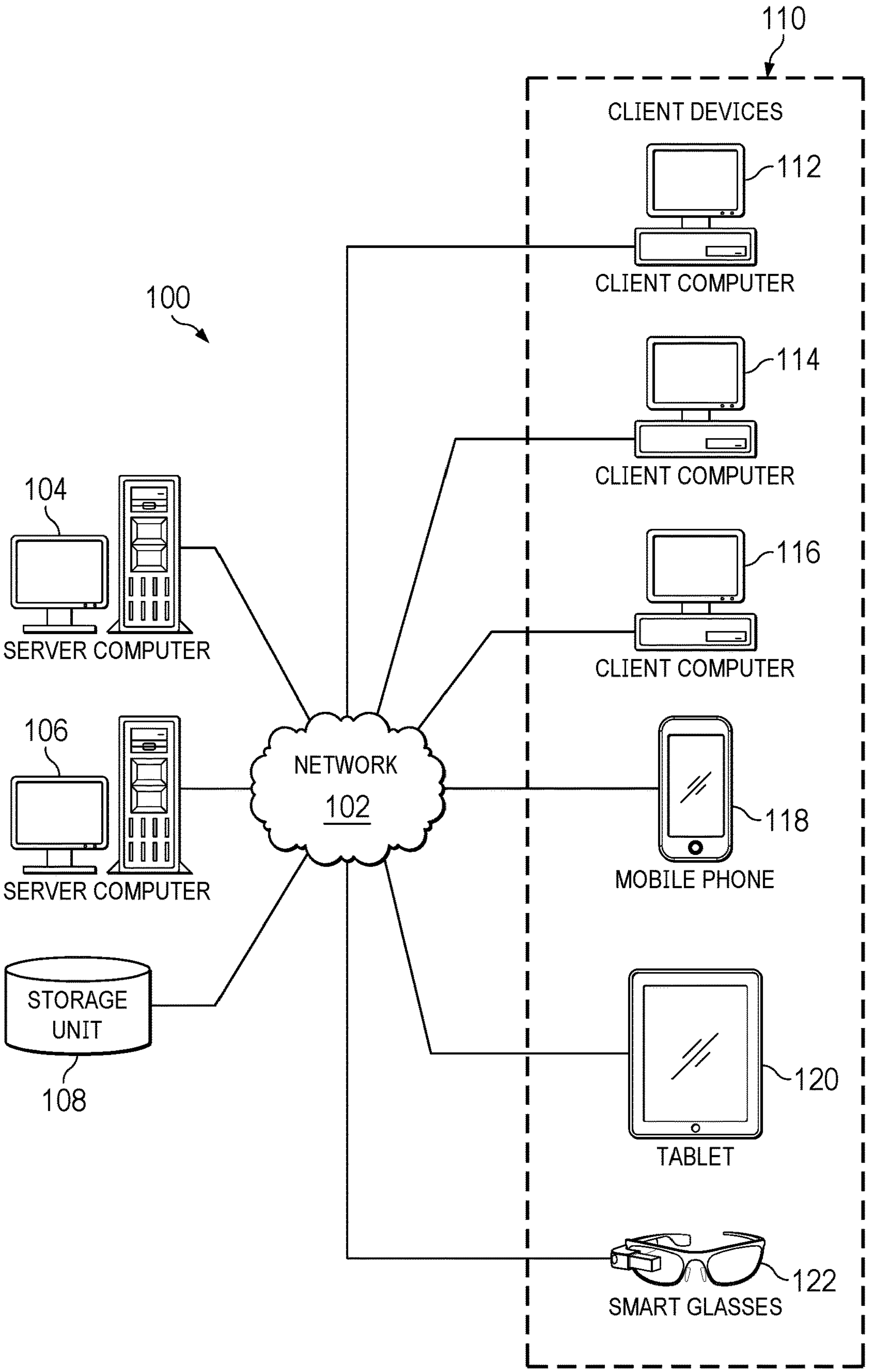

110
CLIENT DEVICES
112
CLIENT COMPUTER
100
114
CLIENT COMPUTER
104
SERVER COMPUTER
116
CLIENT COMPUTER
106
NETWORK
102
118
SERVER COMPUTER
MOBILE PHONE
STORAGE UNIT
108
120
TABLET
122
SMART GLASSES

FIG. 1

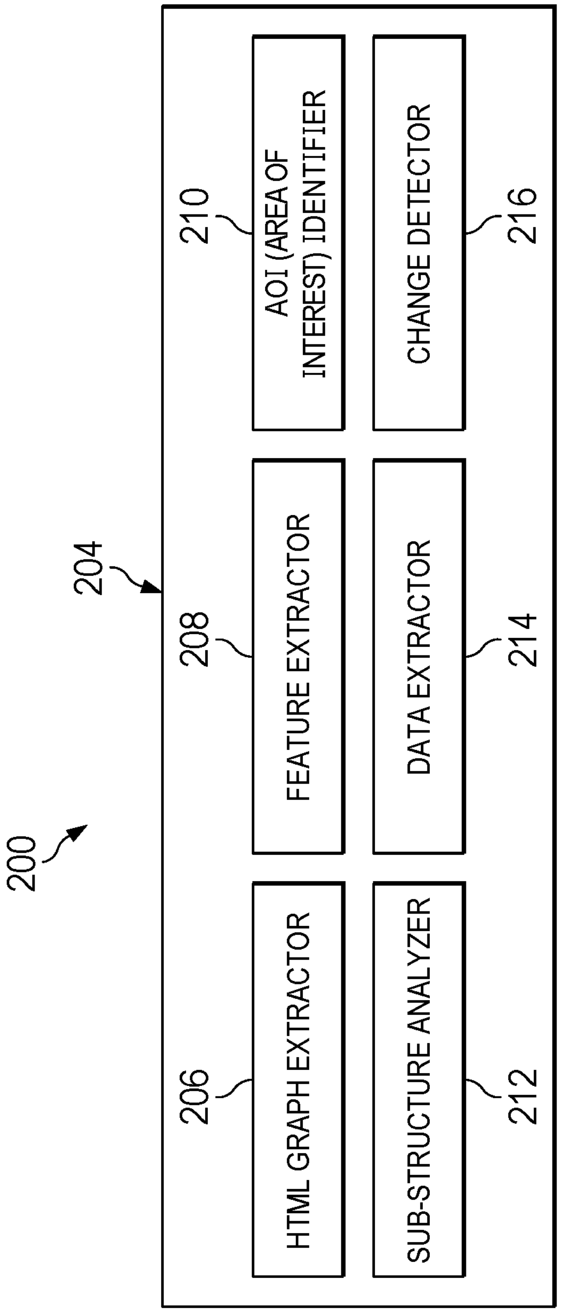

200
204
206
HTML GRAPH EXTRACTOR
208
FEATURE EXTRACTOR
210
AOI (AREA OF INTEREST) IDENTIFIER
212
SUB-STRUCTURE ANALYZER
214
DATA EXTRACTOR
216
CHANGE DETECTOR

FIG. 2

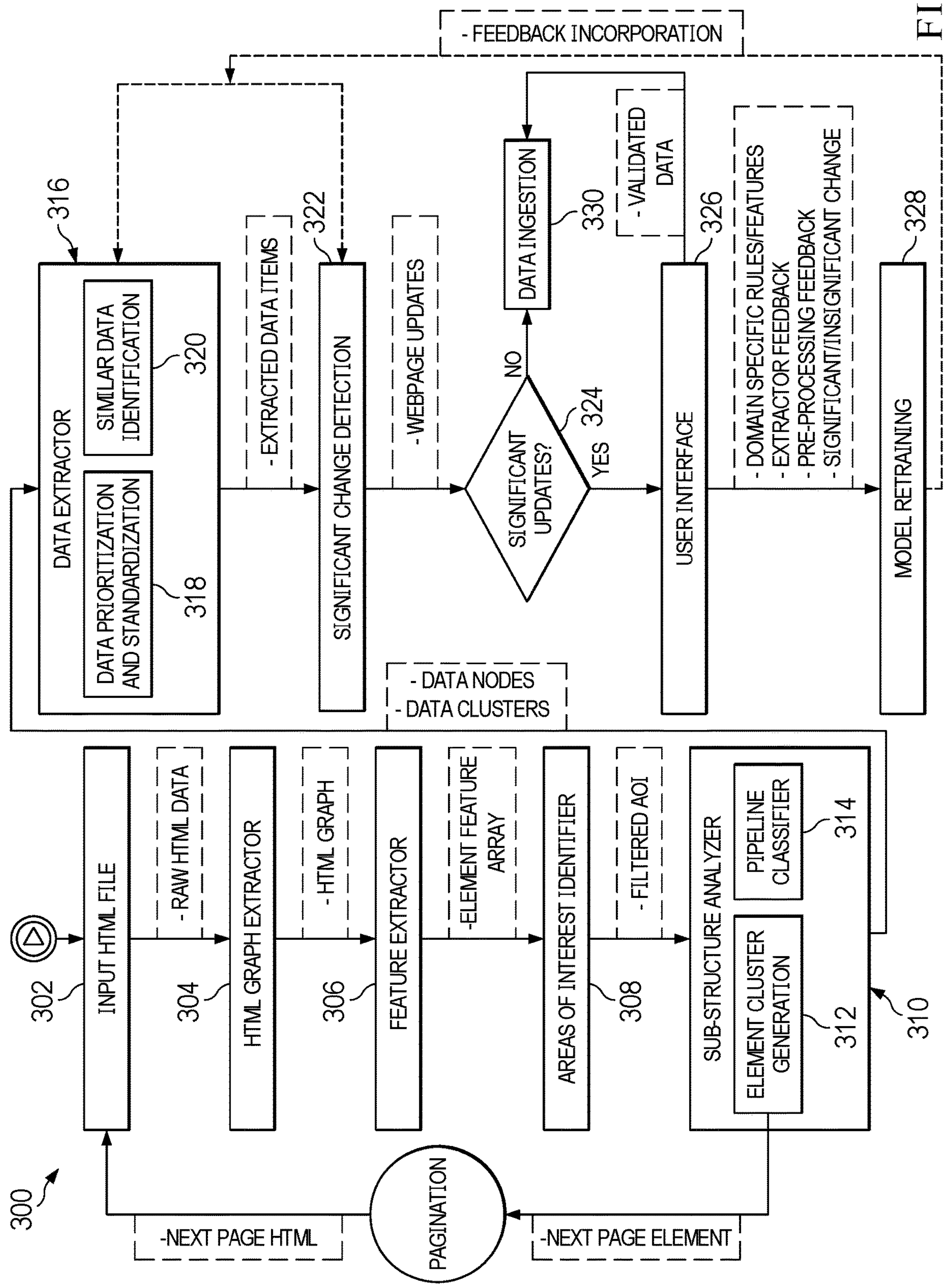

300
302
INPUT HTML FILE
-RAW HTML DATA
304
HTML GRAPH EXTRACTOR
-HTML GRAPH
306
FEATURE EXTRACTOR
-ELEMENT FEATURE ARRAY
308
AREAS OF INTEREST IDENTIFIER
-FILTERED AOI
310
SUB-STRUCTURE ANALYZER
312
ELEMENT CLUSTER GENERATION
314
PIPELINE CLASSIFIER
-NEXT PAGE ELEMENT
PAGINATION
-NEXT PAGE HTML
- DATA NODES
- DATA CLUSTERS
316
DATA EXTRACTOR
318
DATA PRIORITIZATION AND STANDARDIZATION
320
SIMILAR DATA IDENTIFICATION
- EXTRACTED DATA ITEMS
322
SIGNIFICANT CHANGE DETECTION
- WEBPAGE UPDATES
324
SIGNIFICANT UPDATES?
NO
YES
330
DATA INGESTION
- VALIDATED DATA
326
USER INTERFACE
- DOMAIN SPECIFIC RULES/FEATURES
- EXTRACTOR FEEDBACK
- PRE-PROCESSING FEEDBACK
- SIGNIFICANT/INSIGNIFICANT CHANGE
328
MODEL RETRAINING
- FEEDBACK INCORPORATION

FIG. 3

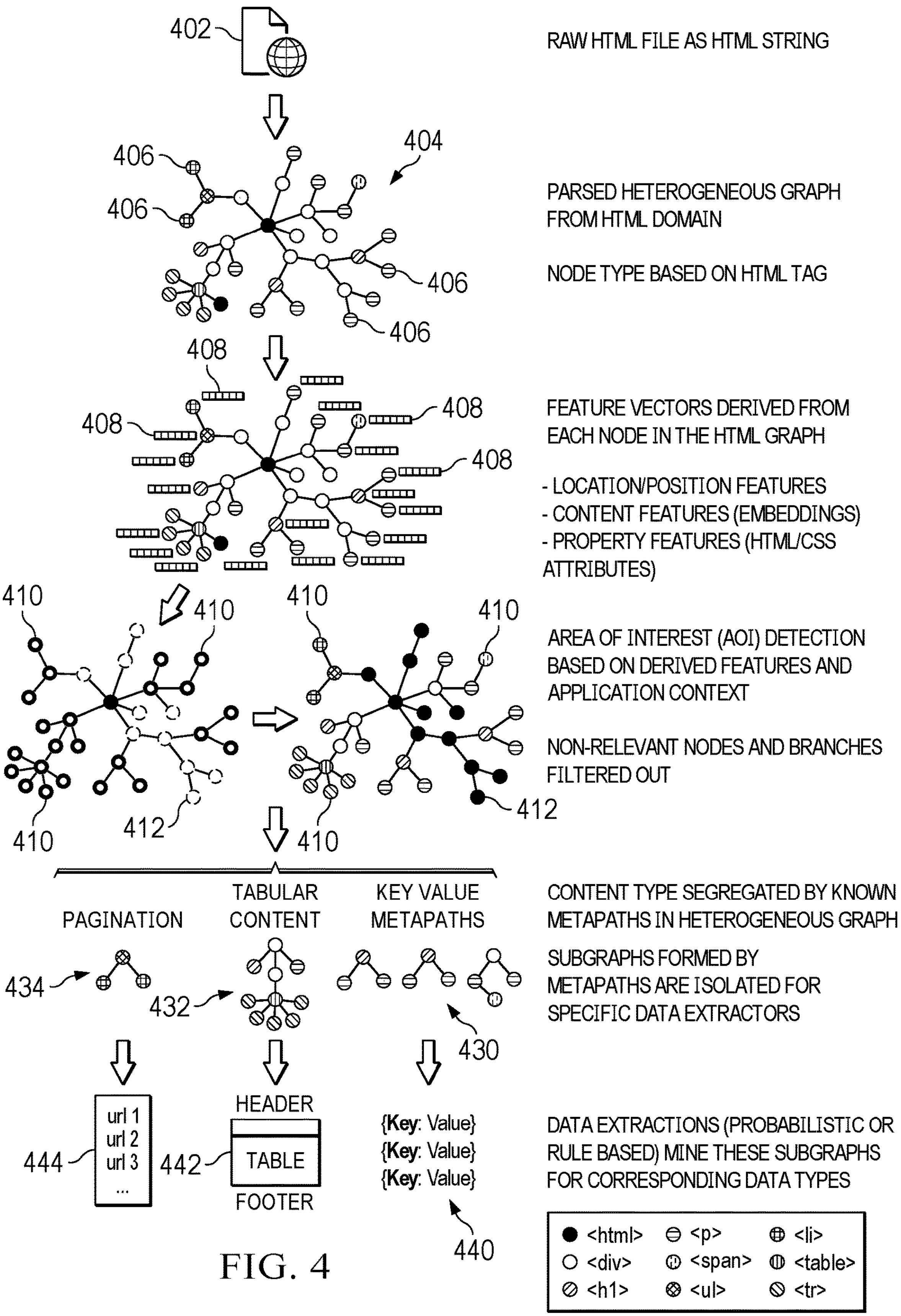
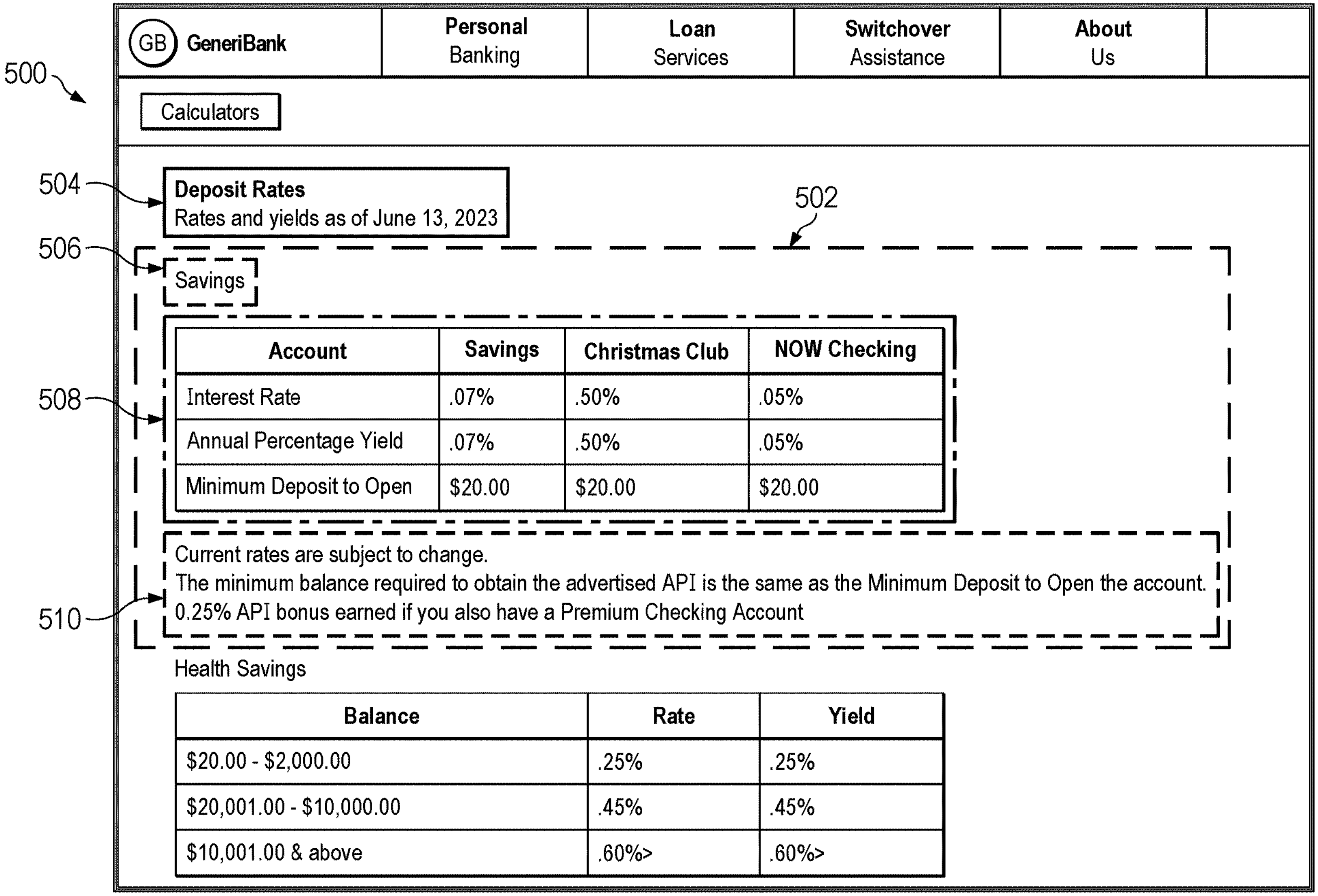

402
RAW HTML FILE AS HTML STRING
404
406
406
406
406
PARSED HETEROGENEOUS GRAPH FROM HTML DOMAIN
NODE TYPE BASED ON HTML TAG
408
408
408
408
FEATURE VECTORS DERIVED FROM EACH NODE IN THE HTML GRAPH
- LOCATION/POSITION FEATURES
- CONTENT FEATURES (EMBEDDINGS)
- PROPERTY FEATURES (HTML/CSS ATTRIBUTES)
410
410
410
410
410
410
412
412
AREA OF INTEREST (AOI) DETECTION BASED ON DERIVED FEATURES AND APPLICATION CONTEXT
NON-RELEVANT NODES AND BRANCHES FILTERED OUT
PAGINATION
TABULAR CONTENT
KEY VALUE METAPATHS
CONTENT TYPE SEGREGATED BY KNOWN METAPATHS IN HETEROGENEOUS GRAPH
434
432
430
SUBGRAPHS FORMED BY METAPATHS ARE ISOLATED FOR SPECIFIC DATA EXTRACTORS
444
url 1
url 2
url 3
...
442
HEADER
TABLE
FOOTER
{Key: Value}
{Key: Value}
{Key: Value}
440
DATA EXTRACTIONS (PROBABILISTIC OR RULE BASED) MINE THESE SUBGRAPHS FOR CORRESPONDING DATA TYPES
● <html>
○ <div>
⊘ <h1>
⊖ <p>
<span>
⊗ <ul>
⊕ <li>
<table>
⊘ <tr>

FIG. 4



Deposit Rates
Rates and yields as of June 13, 2023

Account	Savings	Christmas Club	NOW Checking
Interest Rate	.07%	.50%	.05%
Annual Percentage Yield	.07%	.50%	.05%
Minimum Deposit to Open	$20.00	$20.00	$20.00

Current rates are subject to change.
The minimum balance required to obtain the advertised API is the same as the Minimum Deposit to Open the account.
0.25% API bonus earned if you also have a Premium Checking Account

Health Savings

Balance	Rate	Yield
$20.00 - $2,000.00	.25%	.25%
$20,001.00 - $10,000.00	.45%	.45%
$10,001.00 & above	.60%>	.60%>

FIG. 5A

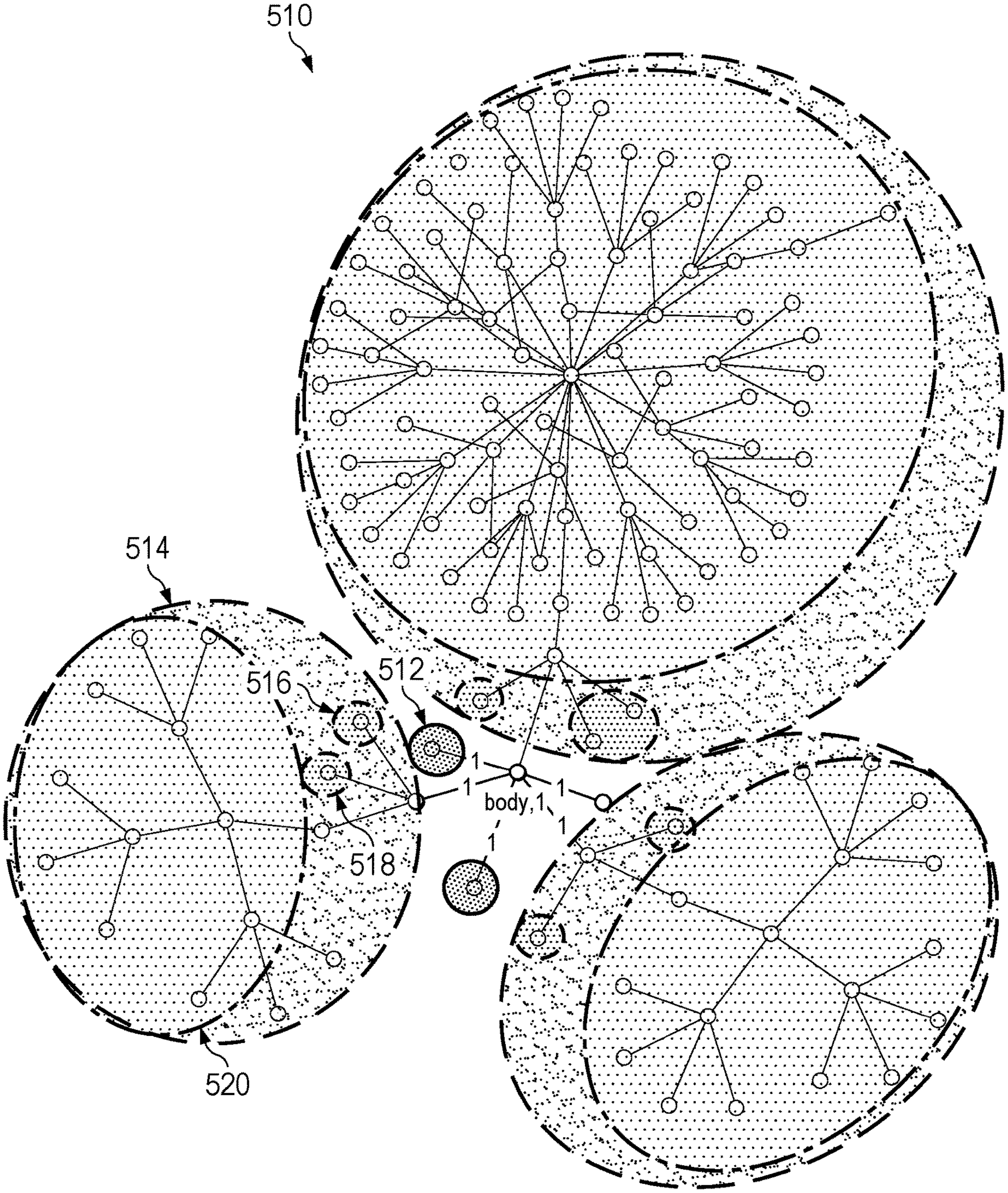
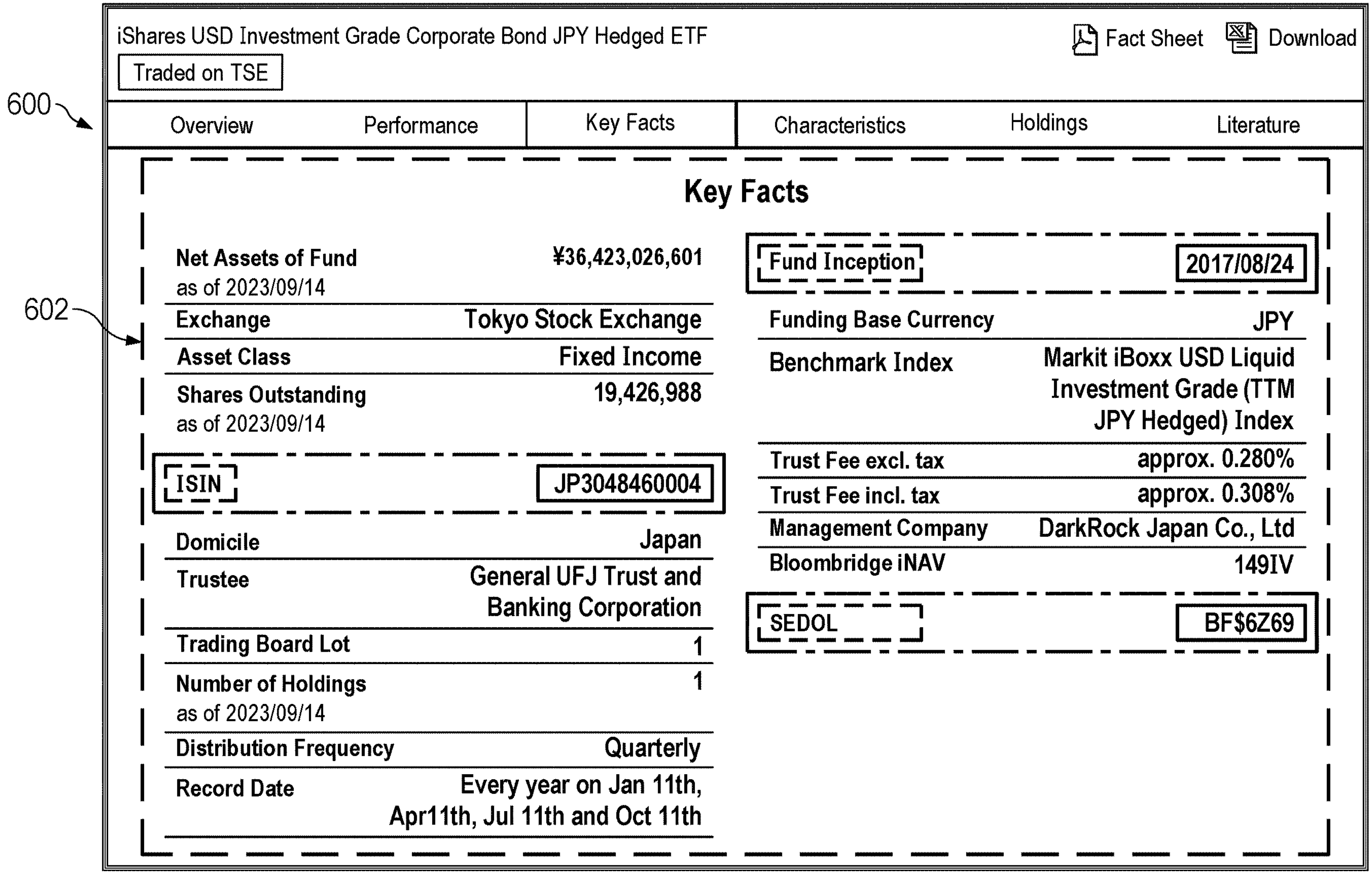


FIG. 5B



iShares USD Investment Grade Corporate Bond JPY Hedged ETF

Traded on TSE

Key Facts

Net Assets of Fund as of 2023/09/14	¥36,423,026,601
Exchange	Tokyo Stock Exchange
Asset Class	Fixed Income
Shares Outstanding as of 2023/09/14	19,426,988
ISIN	JP3048460004
Domicile	Japan
Trustee	General UFJ Trust and Banking Corporation
Trading Board Lot	1
Number of Holdings as of 2023/09/14	1
Distribution Frequency	Quarterly
Record Date	Every year on Jan 11th, Apr11th, Jul 11th and Oct 11th

Fund Inception	2017/08/24
Funding Base Currency	JPY
Benchmark Index	Markit iBoxx USD Liquid Investment Grade (TTM JPY Hedged) Index
Trust Fee excl. tax	approx. 0.280%
Trust Fee incl. tax	approx. 0.308%
Management Company	DarkRock Japan Co., Ltd
Bloombridge iNAV	149IV
SEDOL	BF$6Z69

FIG. 6A

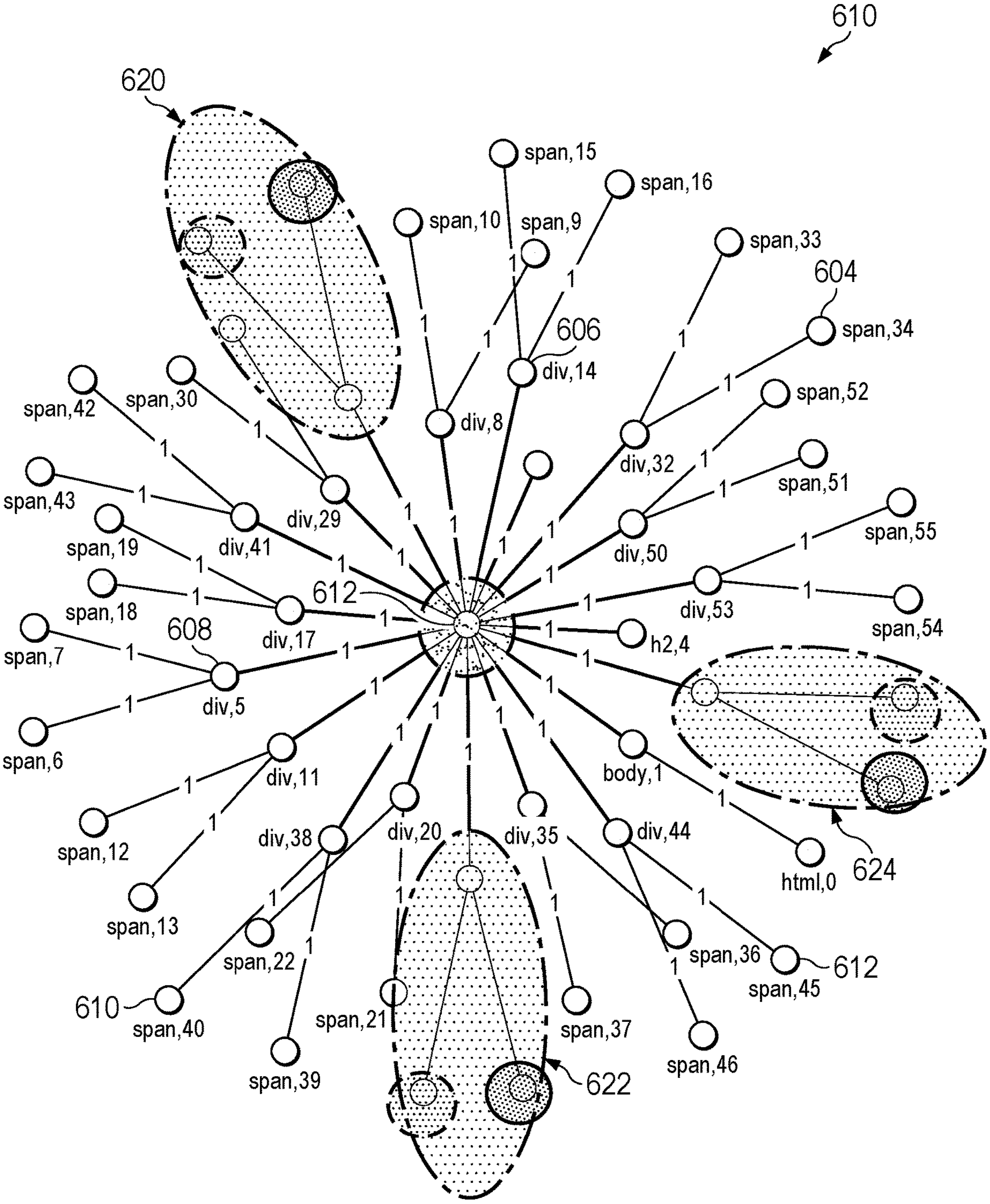
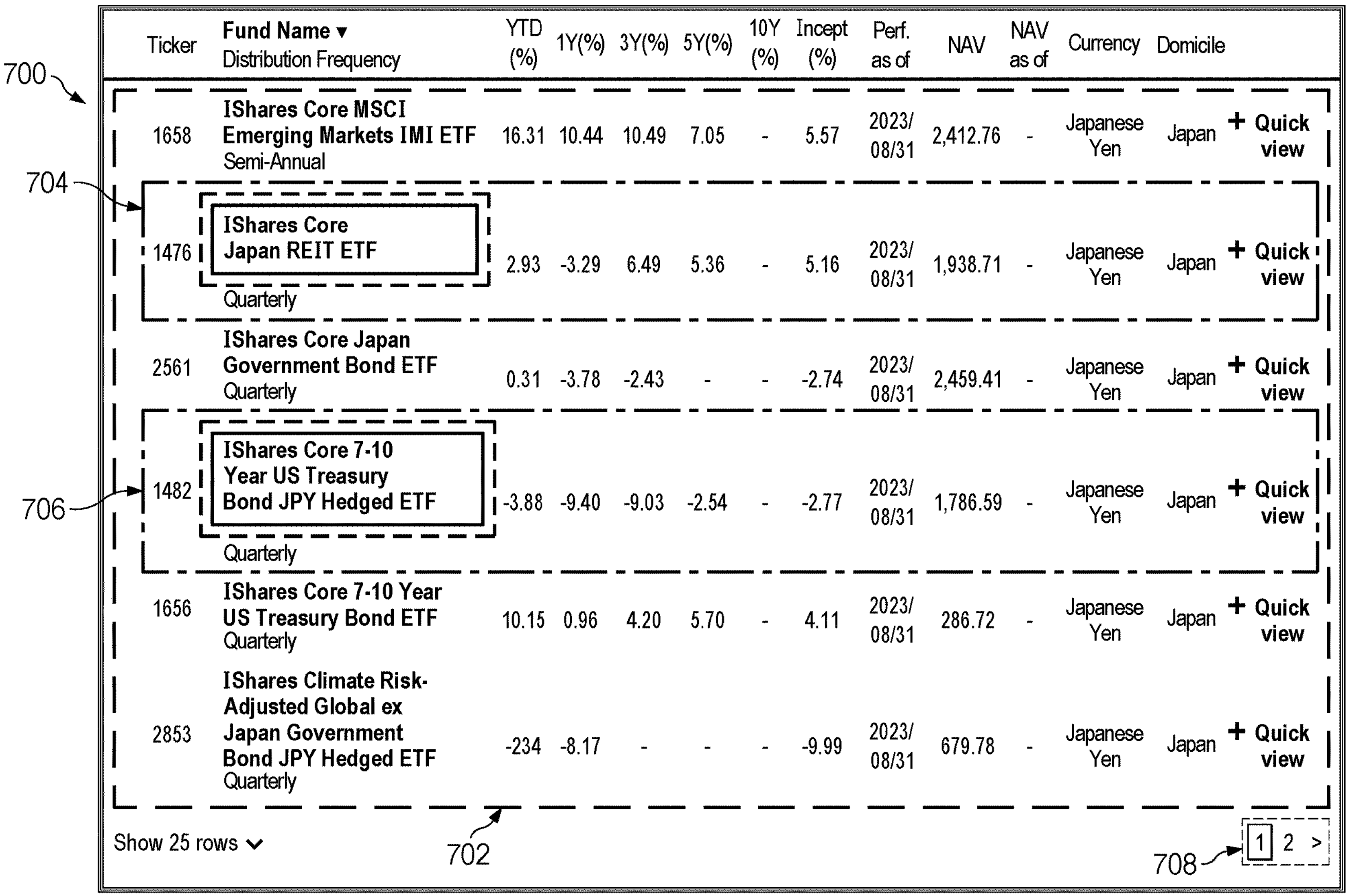

610
620
span,15
span,16
span,10
span,9
span,33
604
span,34
606
div,14
span,42
span,30
div,8
span,52
div,32
span,43
span,51
div,29
div,41
div,50
span,55
span,19
612
span,18
div,17
div,53
span,54
span,7
608
h2,4
div,5
span,6
div,11
body,1
span,12
div,38
div,20
div,35
div,44
html,0
624
span,13
span,22
span,36
span,45
612
610
span,40
span,21
span,37
span,39
span,46
622

FIG. 6B



Ticker	**Fund Name ▾** Distribution Frequency	YTD (%)	1Y(%)	3Y(%)	5Y(%)	10Y (%)	Incept (%)	Perf. as of	NAV	NAV as of	Currency	Domicile	
1658	**IShares Core MSCI Emerging Markets IMI ETF** Semi-Annual	16.31	10.44	10.49	7.05	-	5.57	2023/ 08/31	2,412.76	-	Japanese Yen	Japan	**+ Quick view**
1476	**IShares Core Japan REIT ETF** Quarterly	2.93	-3.29	6.49	5.36	-	5.16	2023/ 08/31	1,938.71	-	Japanese Yen	Japan	**+ Quick view**
2561	**IShares Core Japan Government Bond ETF** Quarterly	0.31	-3.78	-2.43	-	-	-2.74	2023/ 08/31	2,459.41	-	Japanese Yen	Japan	**+ Quick view**
1482	**IShares Core 7-10 Year US Treasury Bond JPY Hedged ETF** Quarterly	-3.88	-9.40	-9.03	-2.54	-	-2.77	2023/ 08/31	1,786.59	-	Japanese Yen	Japan	**+ Quick view**
1656	**IShares Core 7-10 Year US Treasury Bond ETF** Quarterly	10.15	0.96	4.20	5.70	-	4.11	2023/ 08/31	286.72	-	Japanese Yen	Japan	**+ Quick view**
2853	**IShares Climate Risk-Adjusted Global ex Japan Government Bond JPY Hedged ETF** Quarterly	-234	-8.17	-	-	-	-9.99	2023/ 08/31	679.78	-	Japanese Yen	Japan	**+ Quick view**

FIG. 7A

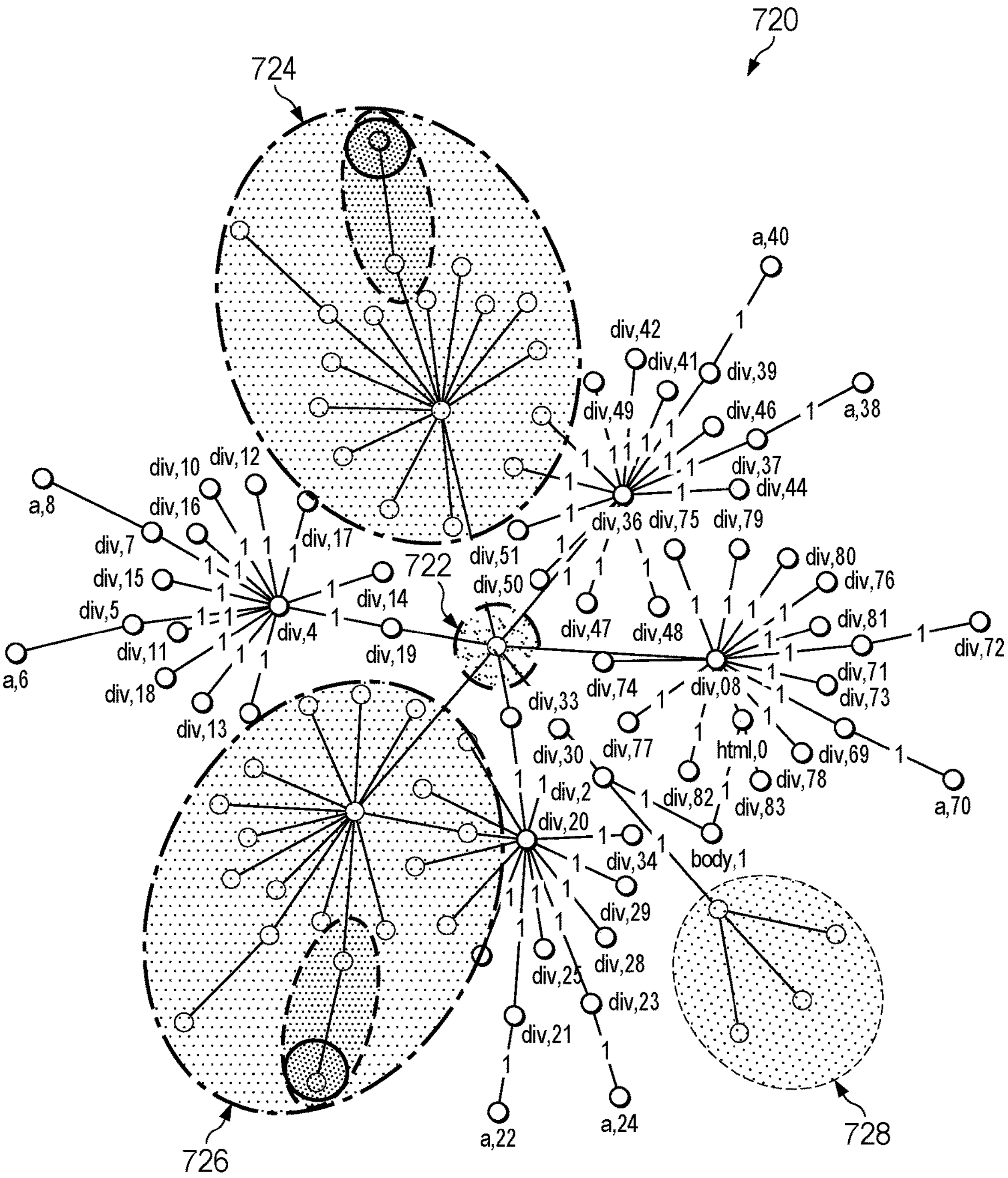
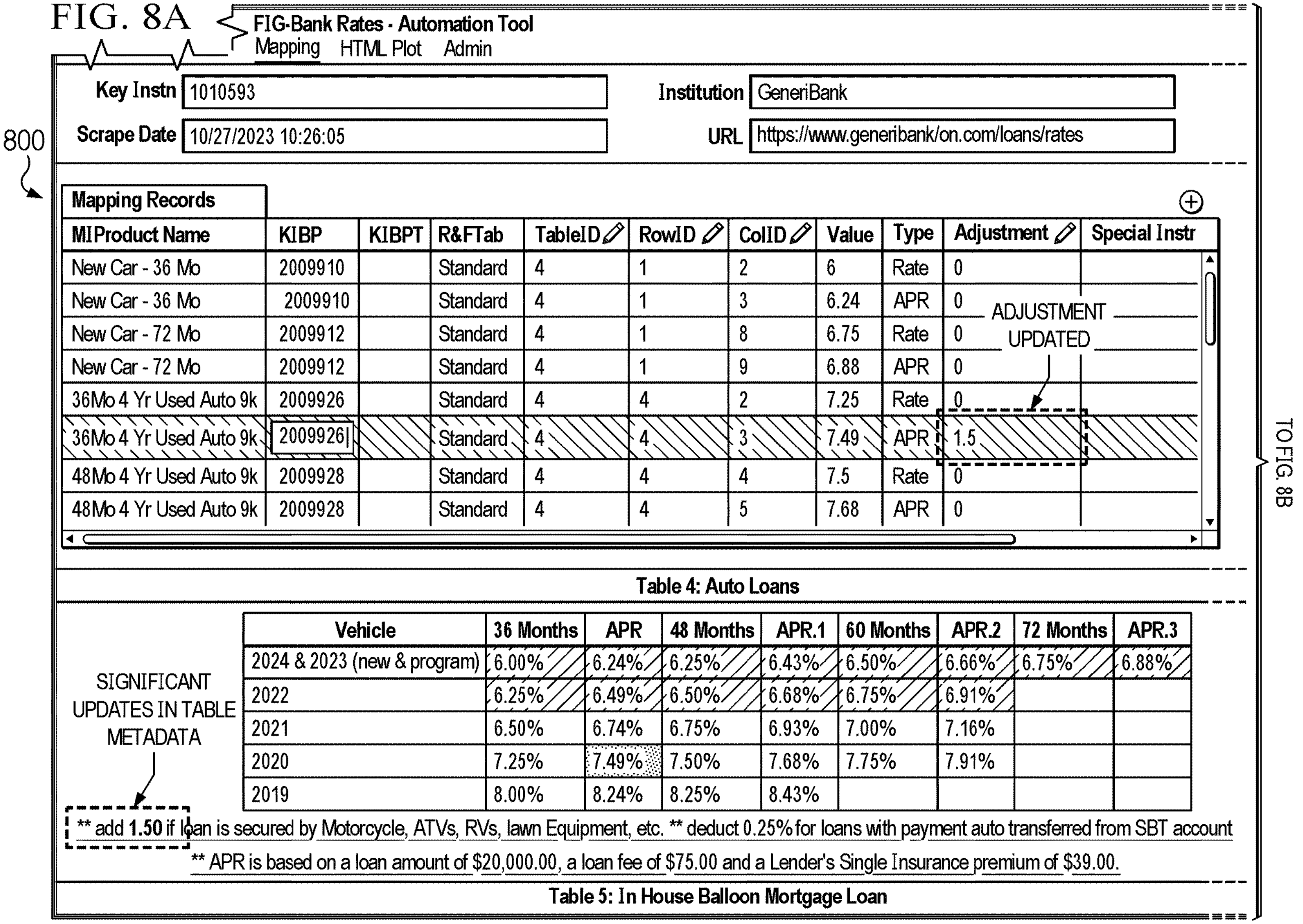

720
724
a,40
div,42
div,41
div,39
div,49
div,46
a,38
div,37
div,44
div,10
div,12
a,8
div,16
div,17
div,36
div,75
div,79
div,80
div,7
div,51
div,76
div,15
722
div,50
div,14
div,5
div,4
div,47
div,48
div,81
div,72
div,11
div,19
a,6
div,18
div,74
div,08
div,71
div,73
div,33
div,13
div,77
html,0
div,69
div,30
div,78
div,2
div,82
div,83
a,70
div,20
div,34
body,1
div,29
div,28
div,25
div,23
div,21
a,24
a,22
726
728
1

FIG. 7B

FIG. 8A


800
FIG-Bank Rates - Automation Tool
Mapping
HTML Plot
Admin
ADJUSTMENT UPDATED
SIGNIFICANT UPDATES IN TABLE METADATA
TO FIG. 8B

Key Instn 1010593

Scrape Date 10/27/2023 10:26:05

Institution GeneriBank

URL https://www.generibank/on.com/loans/rates

Mapping Records

MIProduct Name	KIBP	KIBPT	R&FTab	TableID	RowID	ColID	Value	Type	Adjustment	Special Instr
New Car - 36 Mo	2009910		Standard	4	1	2	6	Rate	0	
New Car - 36 Mo	2009910		Standard	4	1	3	6.24	APR	0	
New Car - 72 Mo	2009912		Standard	4	1	8	6.75	Rate	0	
New Car - 72 Mo	2009912		Standard	4	1	9	6.88	APR	0	
36Mo 4 Yr Used Auto 9k	2009926		Standard	4	4	2	7.25	Rate	0	
36Mo 4 Yr Used Auto 9k	2009926		Standard	4	4	3	7.49	APR	1.5	
48Mo 4 Yr Used Auto 9k	2009928		Standard	4	4	4	7.5	Rate	0	
48Mo 4 Yr Used Auto 9k	2009928		Standard	4	4	5	7.68	APR	0	

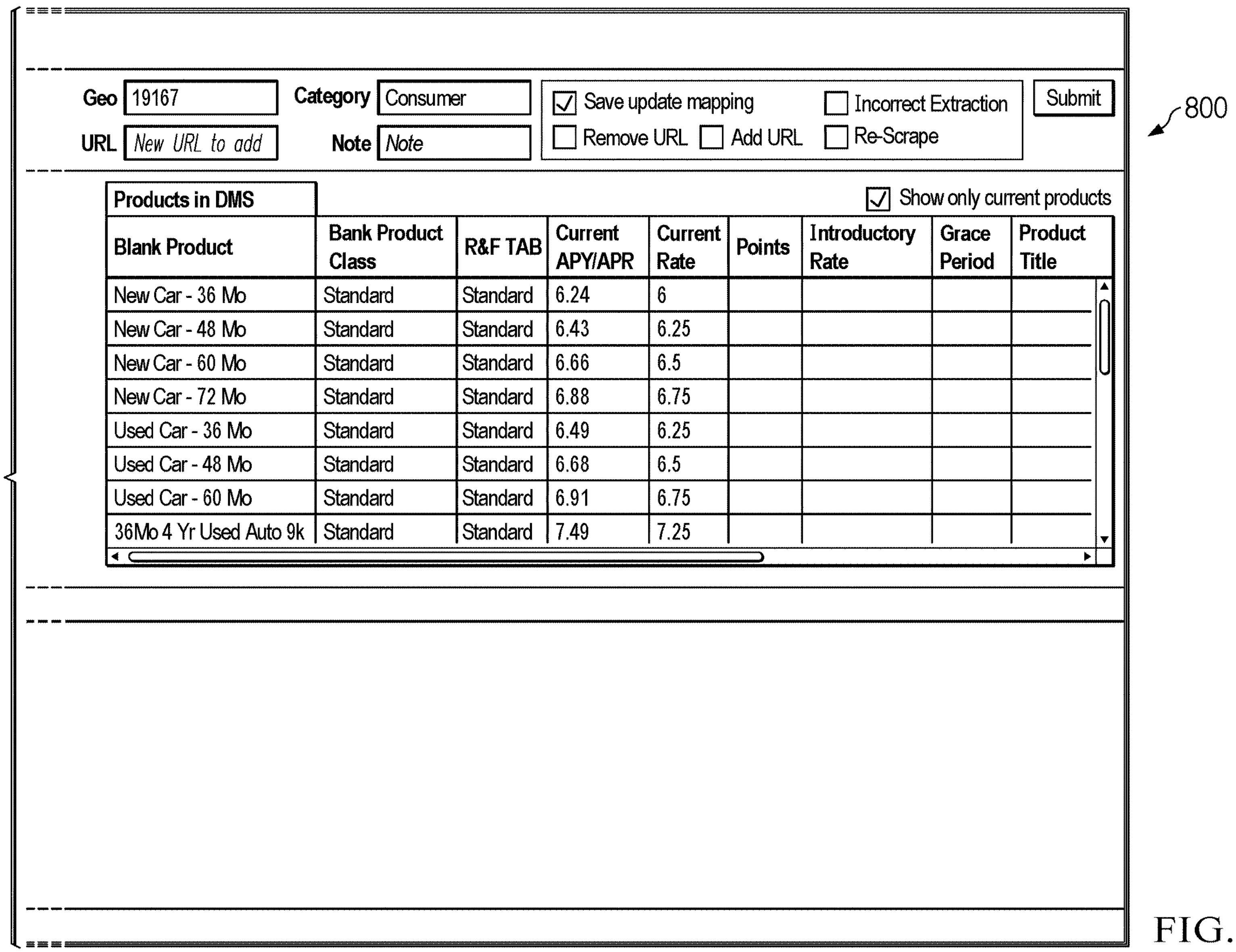
Table 4: Auto Loans

Vehicle	36 Months	APR	48 Months	APR.1	60 Months	APR.2	72 Months	APR.3
2024 & 2023 (new & program)	6.00%	6.24%	6.25%	6.43%	6.50%	6.66%	6.75%	6.88%
2022	6.25%	6.49%	6.50%	6.68%	6.75%	6.91%		
2021	6.50%	6.74%	6.75%	6.93%	7.00%	7.16%		
2020	7.25%	7.49%	7.50%	7.68%	7.75%	7.91%		
2019	8.00%	8.24%	8.25%	8.43%				

** add **1.50** if loan is secured by Motorcycle, ATVs, RVs, lawn Equipment, etc. ** deduct 0.25% for loans with payment auto transferred from SBT account

** APR is based on a loan amount of $20,000.00, a loan fee of $75.00 and a Lender's Single Insurance premium of $39.00.

Table 5: In House Balloon Mortgage Loan



Geo 19167 Category Consumer
URL *New URL to add* Note *Note*

☑ Save update mapping ☐ Incorrect Extraction
☐ Remove URL ☐ Add URL ☐ Re-Scrape

Products in DMS ☑ Show only current products

Blank Product	Bank Product Class	R&F TAB	Current APY/APR	Current Rate	Points	Introductory Rate	Grace Period	Product Title
New Car - 36 Mo	Standard	Standard	6.24	6				
New Car - 48 Mo	Standard	Standard	6.43	6.25				
New Car - 60 Mo	Standard	Standard	6.66	6.5				
New Car - 72 Mo	Standard	Standard	6.88	6.75				
Used Car - 36 Mo	Standard	Standard	6.49	6.25				
Used Car - 48 Mo	Standard	Standard	6.68	6.5				
Used Car - 60 Mo	Standard	Standard	6.91	6.75				
36Mo 4 Yr Used Auto 9k	Standard	Standard	7.49	7.25				

FROM FIG. 8A

FIG. 8B

FIG. 9A

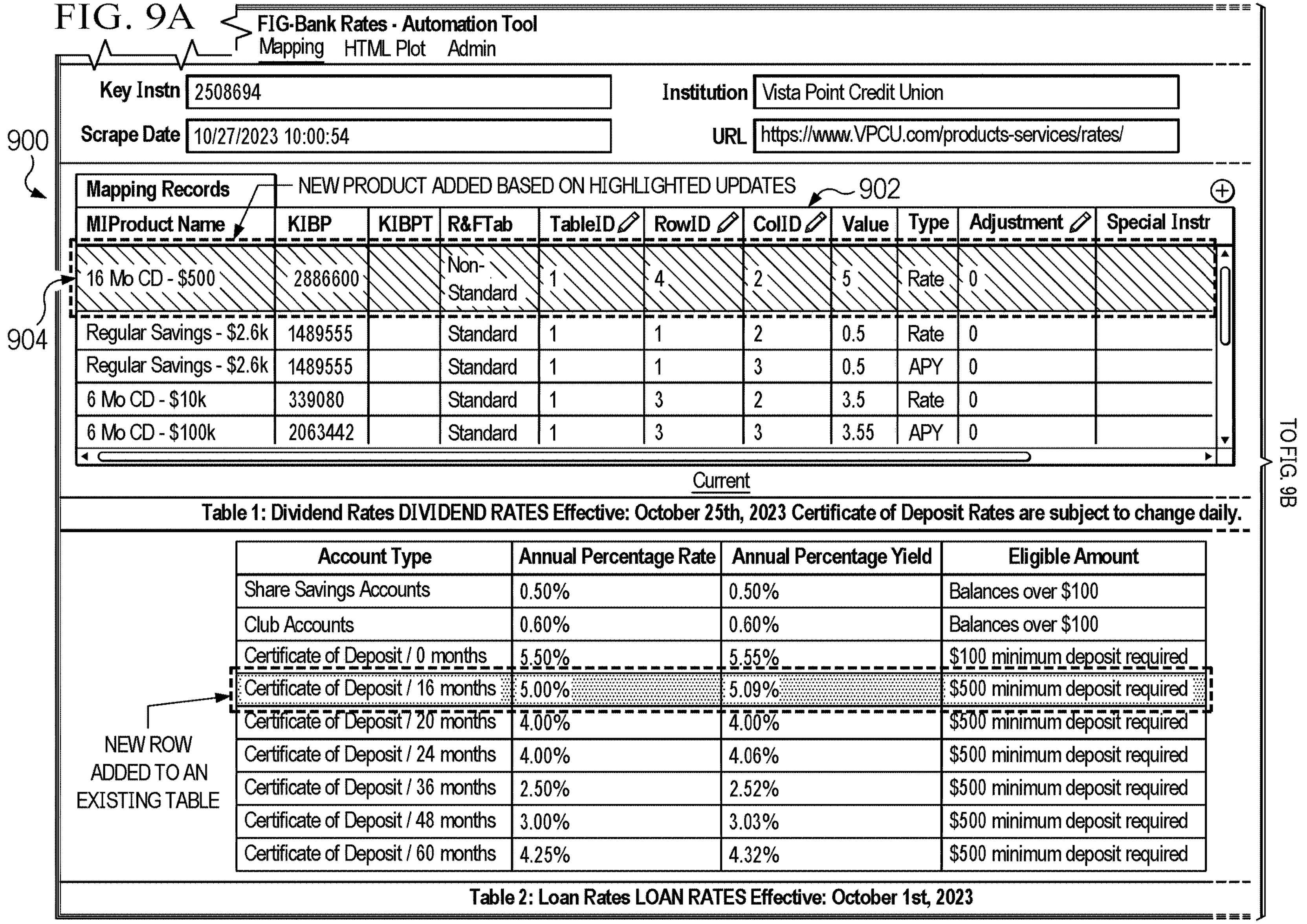

900
904
902
TO FIG. 9B
NEW PRODUCT ADDED BASED ON HIGHLIGHTED UPDATES
NEW ROW ADDED TO AN EXISTING TABLE
Mapping
HTML Plot
Admin
Current

FIG-Bank Rates - Automation Tool

Key Instn 2508694

Scrape Date 10/27/2023 10:00:54

Institution Vista Point Credit Union

URL https://www.VPCU.com/products-services/rates/

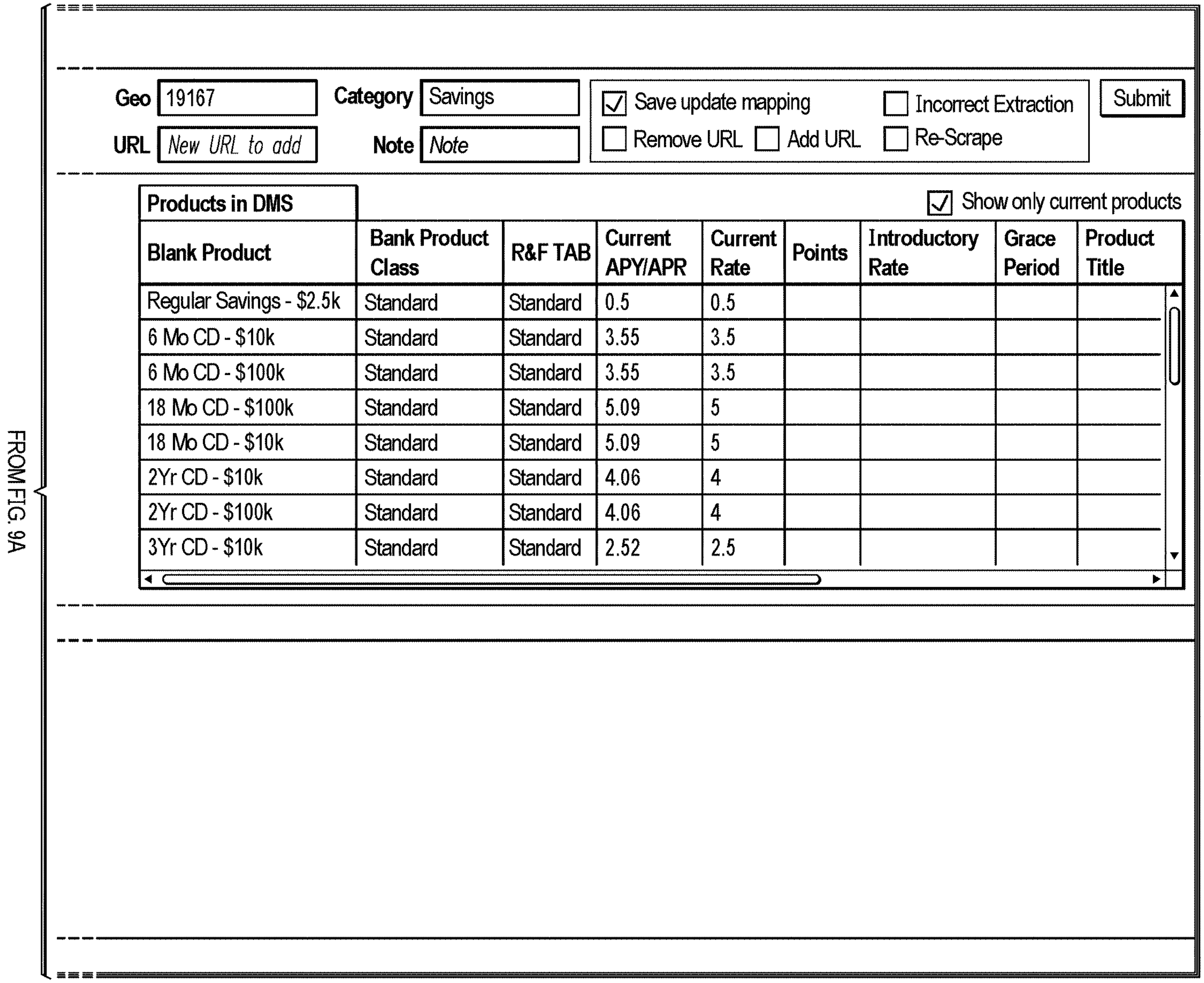
Mapping Records

MIProduct Name	KIBP	KIBPT	R&FTab	TableID	RowID	ColID	Value	Type	Adjustment	Special Instr
16 Mo CD - $500	2886600		Non-Standard	1	4	2	5	Rate	0	
Regular Savings - $2.6k	1489555		Standard	1	1	2	0.5	Rate	0	
Regular Savings - $2.6k	1489555		Standard	1	1	3	0.5	APY	0	
6 Mo CD - $10k	339080		Standard	1	3	2	3.5	Rate	0	
6 Mo CD - $100k	2063442		Standard	1	3	3	3.55	APY	0	

Table 1: Dividend Rates DIVIDEND RATES Effective: October 25th, 2023 Certificate of Deposit Rates are subject to change daily.

Account Type	Annual Percentage Rate	Annual Percentage Yield	Eligible Amount
Share Savings Accounts	0.50%	0.50%	Balances over $100
Club Accounts	0.60%	0.60%	Balances over $100
Certificate of Deposit / 0 months	5.50%	5.55%	$100 minimum deposit required
Certificate of Deposit / 16 months	5.00%	5.09%	$500 minimum deposit required
Certificate of Deposit / 20 months	4.00%	4.00%	$500 minimum deposit required
Certificate of Deposit / 24 months	4.00%	4.06%	$500 minimum deposit required
Certificate of Deposit / 36 months	2.50%	2.52%	$500 minimum deposit required
Certificate of Deposit / 48 months	3.00%	3.03%	$500 minimum deposit required
Certificate of Deposit / 60 months	4.25%	4.32%	$500 minimum deposit required

Table 2: Loan Rates LOAN RATES Effective: October 1st, 2023



Geo 19167 Category Savings

URL *New URL to add* Note *Note*

☑ Save update mapping ☐ Incorrect Extraction

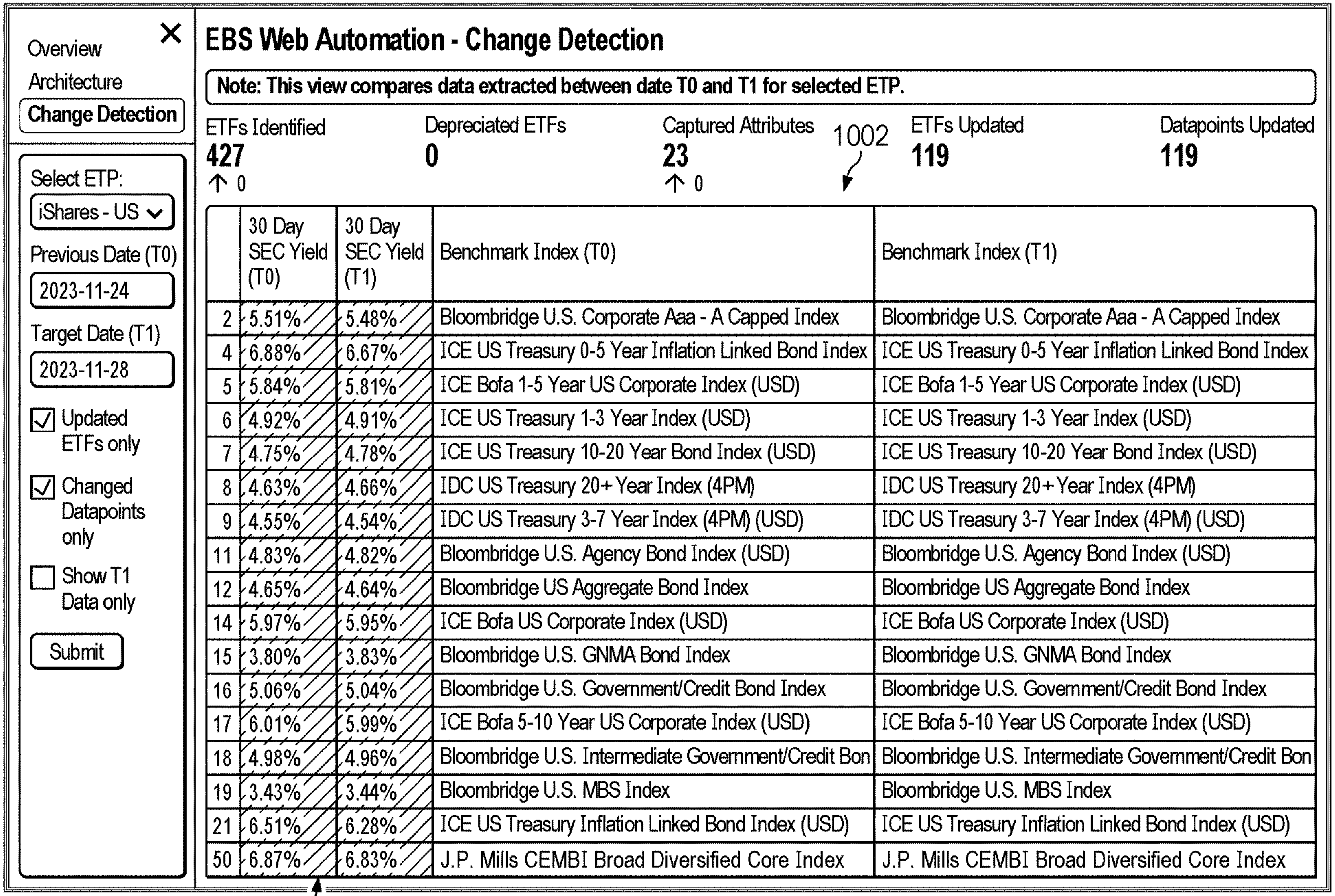
☐ Remove URL ☐ Add URL ☐ Re-Scrape

Products in DMS

☑ Show only current products

Blank Product	Bank Product Class	R&F TAB	Current APY/APR	Current Rate	Points	Introductory Rate	Grace Period	Product Title
Regular Savings - $2.5k	Standard	Standard	0.5	0.5				
6 Mo CD - $10k	Standard	Standard	3.55	3.5				
6 Mo CD - $100k	Standard	Standard	3.55	3.5				
18 Mo CD - $100k	Standard	Standard	5.09	5				
18 Mo CD - $10k	Standard	Standard	5.09	5				
2Yr CD - $10k	Standard	Standard	4.06	4				
2Yr CD - $100k	Standard	Standard	4.06	4				
3Yr CD - $10k	Standard	Standard	2.52	2.5				

FIG. 9B



EBS Web Automation - Change Detection

Note: This view compares data extracted between date T0 and T1 for selected ETP.

ETFs Identified	Depreciated ETFs	Captured Attributes	ETFs Updated	Datapoints Updated
427 ↑ 0	0	23 ↑ 0	119	119

Select ETP: iShares - US

Previous Date (T0): 2023-11-24

Target Date (T1): 2023-11-28

☑ Updated ETFs only

☑ Changed Datapoints only

☐ Show T1 Data only

Submit

	30 Day SEC Yield (T0)	30 Day SEC Yield (T1)	Benchmark Index (T0)	Benchmark Index (T1)
2	5.51%	5.48%	Bloombridge U.S. Corporate Aaa - A Capped Index	Bloombridge U.S. Corporate Aaa - A Capped Index
4	6.88%	6.67%	ICE US Treasury 0-5 Year Inflation Linked Bond Index	ICE US Treasury 0-5 Year Inflation Linked Bond Index
5	5.84%	5.81%	ICE Bofa 1-5 Year US Corporate Index (USD)	ICE Bofa 1-5 Year US Corporate Index (USD)
6	4.92%	4.91%	ICE US Treasury 1-3 Year Index (USD)	ICE US Treasury 1-3 Year Index (USD)
7	4.75%	4.78%	ICE US Treasury 10-20 Year Bond Index (USD)	ICE US Treasury 10-20 Year Bond Index (USD)
8	4.63%	4.66%	IDC US Treasury 20+ Year Index (4PM)	IDC US Treasury 20+ Year Index (4PM)
9	4.55%	4.54%	IDC US Treasury 3-7 Year Index (4PM) (USD)	IDC US Treasury 3-7 Year Index (4PM) (USD)
11	4.83%	4.82%	Bloombridge U.S. Agency Bond Index (USD)	Bloombridge U.S. Agency Bond Index (USD)
12	4.65%	4.64%	Bloombridge US Aggregate Bond Index	Bloombridge US Aggregate Bond Index
14	5.97%	5.95%	ICE Bofa US Corporate Index (USD)	ICE Bofa US Corporate Index (USD)
15	3.80%	3.83%	Bloombridge U.S. GNMA Bond Index	Bloombridge U.S. GNMA Bond Index
16	5.06%	5.04%	Bloombridge U.S. Government/Credit Bond Index	Bloombridge U.S. Government/Credit Bond Index
17	6.01%	5.99%	ICE Bofa 5-10 Year US Corporate Index (USD)	ICE Bofa 5-10 Year US Corporate Index (USD)
18	4.98%	4.96%	Bloombridge U.S. Intermediate Government/Credit Bon	Bloombridge U.S. Intermediate Government/Credit Bon
19	3.43%	3.44%	Bloombridge U.S. MBS Index	Bloombridge U.S. MBS Index
21	6.51%	6.28%	ICE US Treasury Inflation Linked Bond Index (USD)	ICE US Treasury Inflation Linked Bond Index (USD)
50	6.87%	6.83%	J.P. Mills CEMBI Broad Diversified Core Index	J.P. Mills CEMBI Broad Diversified Core Index

FIG. 10

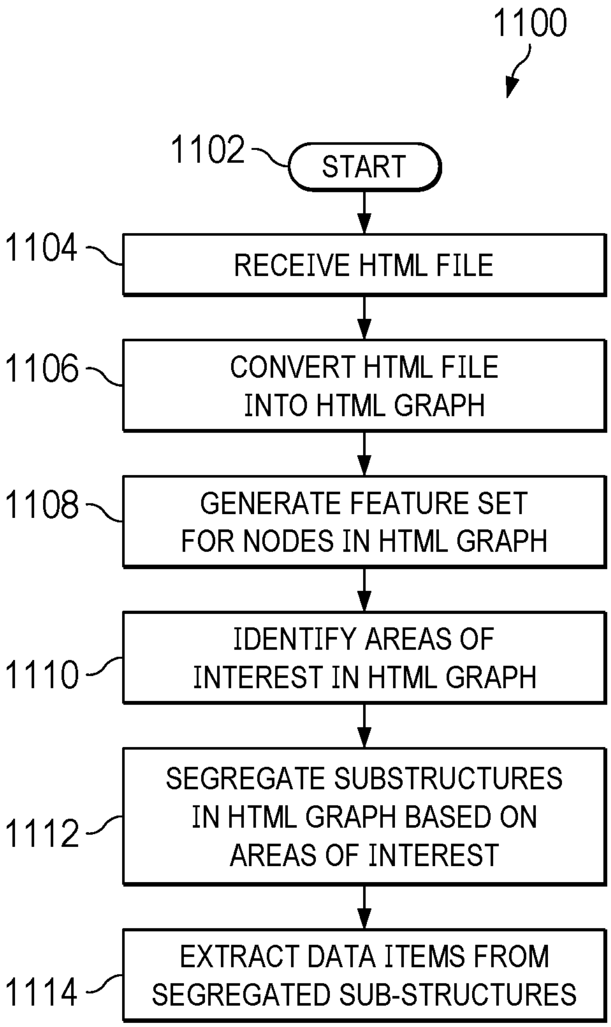

1100
1102
START
1104
RECEIVE HTML FILE
1106
CONVERT HTML FILE INTO HTML GRAPH
1108
GENERATE FEATURE SET FOR NODES IN HTML GRAPH
1110
IDENTIFY AREAS OF INTEREST IN HTML GRAPH
1112
SEGREGATE SUBSTRUCTURES IN HTML GRAPH BASED ON AREAS OF INTEREST
1114
EXTRACT DATA ITEMS FROM SEGREGATED SUB-STRUCTURES

FIG. 11

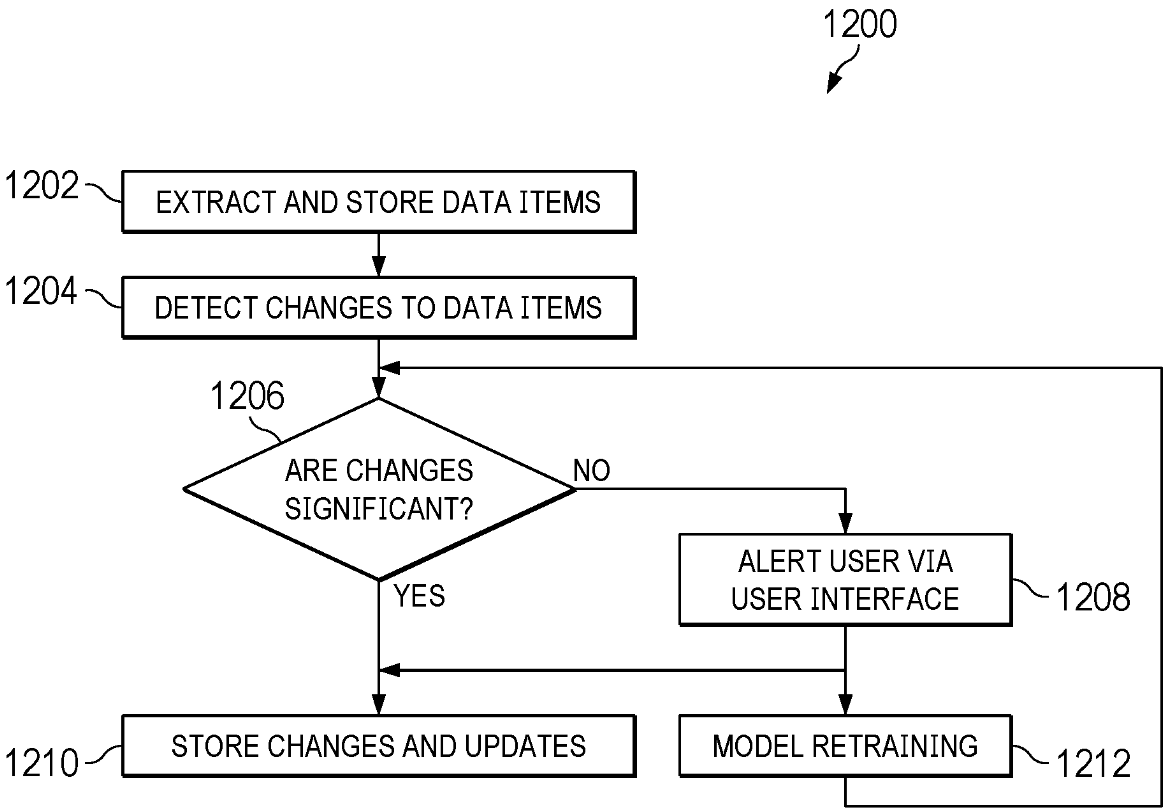

1200
1202
EXTRACT AND STORE DATA ITEMS
1204
DETECT CHANGES TO DATA ITEMS
1206
ARE CHANGES SIGNIFICANT?
NO
YES
ALERT USER VIA USER INTERFACE
1208
1210
STORE CHANGES AND UPDATES
MODEL RETRAINING
1212

FIG. 12

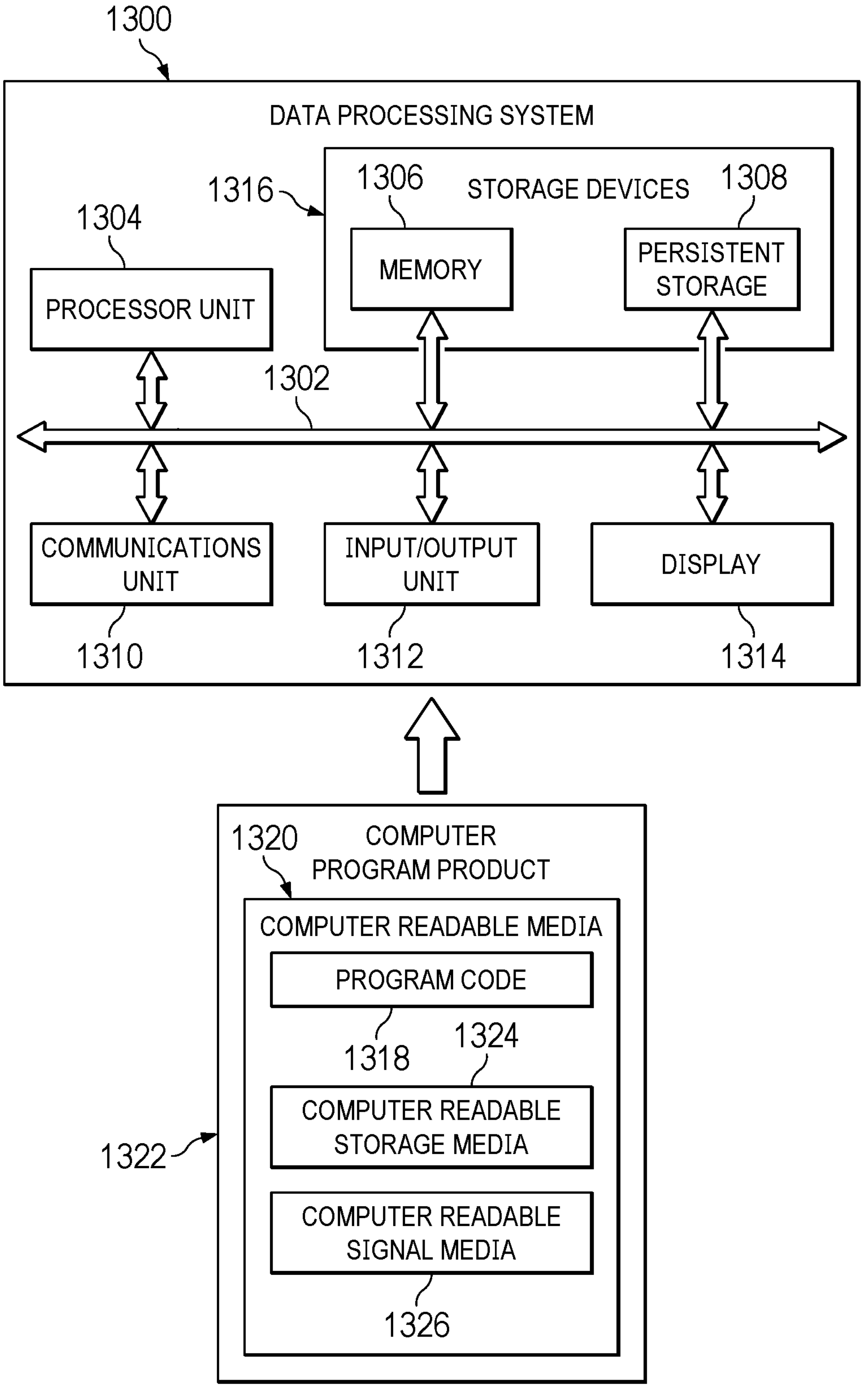

1300
DATA PROCESSING SYSTEM
1316
1306
STORAGE DEVICES
1308
1304
MEMORY
PERSISTENT STORAGE
PROCESSOR UNIT
1302
COMMUNICATIONS UNIT
INPUT/OUTPUT UNIT
DISPLAY
1310
1312
1314
1320
COMPUTER PROGRAM PRODUCT
COMPUTER READABLE MEDIA
PROGRAM CODE
1318
1324
COMPUTER READABLE STORAGE MEDIA
1322
COMPUTER READABLE SIGNAL MEDIA
1326

FIG. 13

METHOD AND SYSTEM FOR WEB DATA EXTRACTION USING META-PATH GRAPH

BACKGROUND INFORMATION

1. Field

[0001] The present disclosure relates generally to web data extraction, and more specifically to a method and system for web data extraction using meta-path graphs.

2. Background

[0002] Conventional web data extraction methods have limitations and challenges. Current methods of extracting web data from HTML documents predominantly rely on predefined templates, fixed pointers or rigid rules. These approaches suffer from scalability issues, requiring substantial maintenance overhead to set up and manage crawlers through manual feature engineering and logic definition.

[0003] Due to the forgiving nature of HTML syntax that allows variations and shortcuts while designing an HTML document, extracting data from HTML documents can pose additional challenges. These variations in HTML syntax can lead to several drawbacks when trying to extract data. Some of the drawbacks are ambiguity in structure, parsing errors, inconsistent data extraction and difficulty in identifying elements.

[0004] Furthermore, conventional web data extraction methods often fail to discriminate between relevant changes on web pages crucial for organizational use cases and inconsequential alterations resulting from page geometry modifications or noise.

SUMMARY

[0005] An illustrative embodiment provides a computer-implemented method for web data extraction. The method comprises receiving an HTML file containing HTML data and converting the HTML data into an HTML graph. Elements in the HTML file are represented by nodes in the HTML graph and relationships among the elements are represented by meta-paths. The method comprises generating feature sets for the nodes in the HTML graph. The method comprises identifying areas of interest in the HTML graph based on the feature sets of the nodes. The method comprises refining the identified areas of interest by segregating sub-structures having recurring patterns or sequences. The method comprises extracting data items from the segregated sub-structures and storing the extracted data items.

[0006] In an illustrative embodiment, the feature sets are represented by vectors. The vectors include one or more of: location or position vectors; content vectors; property features; and domain specific vectors.

[0007] In an illustrative embodiment, the method comprises detecting changes to web pages by detecting changes to the extracted data items from the segregated sub-structures, wherein the changes to the web pages are modifications or updates to the web pages. The extracted data items are standardized and stored as one or more of: key-value pairs; tabular content; and pagination.

[0008] In an illustrative embodiment, the method comprises determining if the changes to web pages are significant changes. If there are significant changes, users are notified via a user interface.

[0009] In an illustrative embodiment, the method comprises training a machine learning model using the significant changes to form a trained model object and segregating the sub-structures using the trained model object.

[0010] Another illustrative embodiment provides a system for web data extraction. The system comprises a storage device configured to store program instructions. The system comprises one or more processors operably connected to the storage device and configured to execute the program instructions to cause the system to: receive an HTML file containing HTML data; convert the HTML data into an HTML graph, wherein elements in the HTML file are represented by nodes in the HTML graph and relationships among the elements are represented by meta-paths; generate feature sets for the nodes in the HTML graph; identify areas of interest in the HTML graph based on the feature sets of the nodes; refine the identified areas of interest by segregating sub-structures having recurring patterns or sequences; and extract data items from the segregated sub-structures and store the extracted data items.

[0011] Another illustrative embodiment provides a computer program product for web data extraction. The computer program product comprises a computer-readable storage medium having program instructions embodied thereon to perform the steps of: receiving an HTML file containing HTML data; converting the HTML data into an HTML graph, wherein elements in the HTML file are represented by nodes in the HTML graph and relationships among the elements are represented by meta-paths; generating feature sets for the nodes in the HTML graph; identifying areas of interest in the HTML graph based on the feature sets of the nodes; refining the identified areas of interest by segregating sub-structures having recurring patterns or sequences; and extracting data items from the segregated sub-structures and storing the extracted data items.

BRIEF DESCRIPTION OF THE DRAWINGS

[0012] The novel features believed characteristic of the illustrative embodiments are set forth in the appended claims. The illustrative embodiments will best be understood by reference to the following detailed description of the illustrative embodiments of the present disclosure when read in conjunction with the accompanying drawings, wherein:

[0013] FIG. **1** illustrates a network of data processing system in accordance with an illustrative embodiment;

[0014] FIG. **2** is a block diagram of a system for web data extraction in accordance with an illustrative embodiment;

[0015] FIG. **3** illustrates a system for web data extraction in accordance with an illustrative embodiment;

[0016] FIG. **4** illustrates web data extraction by constructing an HTML graph in accordance with an illustrative embodiment;

[0017] FIGS. **5**A and **5**B illustrate an example web data extraction in accordance with illustrative embodiments;

[0018] FIGS. **6**A and **6**B illustrate another example web data extraction in accordance with illustrative embodiments;

[0019] FIGS. **7**A and **7**B illustrate another example web data extraction in accordance with illustrative embodiments;

[0020] FIGS. **8**A, **8**B, **9**A, **9**B and **10** illustrate user interfaces showing data extracted from web pages and detection of significant changes;

[0021] FIG. **11** illustrates a flowchart of a process for web data extraction in accordance with an illustrative embodiment;

[0022] FIG. **12** illustrates a flowchart of a process for detecting changes or updates to web pages in accordance with an illustrative embodiment; and

[0023] FIG. **13** illustrates a block diagram of a data processing system in accordance with an illustrative embodiment.

DETAILED DESCRIPTION

[0024] The illustrative embodiments address limitations of conventional web data extraction methods and systems. The illustrative embodiments provide a method and system for web data extraction using a meta-path graph data model.

[0025] In an illustrative embodiment, an HTML file is received. The HTML file contains raw HTML data. An HTML graph extractor converts the raw HTML data into an HTML graph. Each HTML element in the graph is represented by a node, and relationships among the elements are represented by meta-paths (e.g., edges or connections), thus illustrating parent-child relationships in the graph.

[0026] A feature extractor creates a feature set for each node in the graph. The feature set may be represented by vectors which may include: (1) location/position vectors (e.g., coordinates or relative positions); (2) content vectors (e.g., embeddings representing text content); and (3) property vectors (e.g., attributes from HTML/CSS).

[0027] Based on the feature sets, an area of interest (AOI) identifier detects or identifies specific regions, sub-graphs, sub-structures or nodes within the HTML graph (or DOM-Document Object Model) that are of interest to a specific organization or an entity. The specific regions, sub-graphs or sub-structures may include headers, footers, menus, tables, main content, sidebars, etc. The AOI identifier filters out non-relevant nodes. The non-relevant nodes are considered not of interest to the specific organization or entity.

[0028] A sub-structure analyzer refines the identified areas of interest using meta-path based models. The sub-structure analyzer identifies recurring patterns or sequences of nodes and relationships, helping to understand the structural patterns or sequences within the HTML graph. These patterns are recorded (for example, in a database) and correlated to data items for subsequent steps in the process. During inference, the sub-structure analyzer refers to this history to segregate or isolate reoccurring patterns.

[0029] A data extractor extracts data items from the refined areas of interest. The extracted data items are standardized and stored. The extracted values can be stored, for example, as key-value pairs, tabular content (e.g., headers, table, footers) and pagination (e.g., URLs). The extracted values may suggest additional data based on learned patterns or user-defined rules.

[0030] In some embodiments, changes to the web pages are detected. The changes to the web pages can be detected from the changes to segregated or isolated sub-structures in the HTML graph or can be detected from the changes in the extracted data items.

[0031] If changes to the web pages are determined to be significant, users are automatically notified via a user interface for feedback on the changes.

[0032] In some embodiments, using the changed sub-structures, a machine learning model is trained to form a new model object. The trained model object represents a mapping function which makes predictions or decisions on new or unseen sub-structures or data in the HTML graph.

[0033] With reference to FIG. **1**, a pictorial representation of a network of data processing system is depicted in which illustrative embodiments may be implemented. Network data processing system **100** is a network of computers in which the illustrative embodiments may be implemented. Network data processing system **100** contains network **102**, which is the medium used to provide communications links between various devices and computers connected within network data processing system **100**. Network **102** may include connections, such as wire, wireless communication links, or fiber optic cables.

[0034] In the depicted example, server computers **104** and **106** and storage unit **108** connect to network **102**. In addition, client devices **110** connect to network **102**. In the depicted example, server computer **104** provides information, such as boot files, operating system images, and applications to client devices **110**. Client devices **110** can be, for example, computers, workstations, or network computers. As depicted, client devices **110** include client computers **112**, **114**, and **116**. Client devices **110** can also include other types of client devices such as mobile phone **118**, tablet computer **120**, and smart glasses **122**.

[0035] In the illustrative example of FIG. **1**, server computers **104** and **106**, storage unit **108**, and client devices **110** are network devices that connect to network **102** in which network **102** is the communications media for these network devices. Some or all of client devices **110** may form an Internet of things (IoT) in which these physical devices can connect to network **102** and exchange information with each other over network **102**.

[0036] Program code located in network data processing system **100** can be stored on a computer-recordable storage medium and downloaded to a data processing system or other device for use. For example, the program code can be stored on a computer-recordable storage medium on server computers **104** and **106** and storage unit **108** and downloaded to client devices **110** over network **102** for use on client devices **110**.

[0037] In the illustrative example of FIG. **1**, network **102** can be the Internet representing a worldwide collection of networks and gateways that use the Transmission Control Protocol/Internet Protocol (TCP/IP) suite of protocols to communicate with one another. At the heart of the Internet is a backbone of high-speed data communication lines between major nodes or host computers consisting of thousands of commercial, governmental, educational, and other computer systems that route data and messages. Of course, network data processing system **100** also may be implemented using different types of networks. For example, network **102** can be comprised an intranet, a local area network (LAN), a metropolitan area network (MAN), or a wide area network (WAN). FIG. **1** is intended as an example, and not as an architectural limitation for the different illustrative embodiments.

[0038] FIG. **2** is a block diagram of system **200** for web data extraction in accordance with an illustrative embodiment. System **200** comprises computer system **204**. Computer system **204** is a physical hardware system which includes one or more data processing systems. When more than one data processing system is present in computer system **204**, those data processing systems are in communication with each other using a communications medium.

The communications medium can be a network. The data processing systems can be selected from at least one of a computer, a server computer, a tablet computer, or some other suitable data processing system. System **200** can be implemented, for example, in server computers **104** and **106** or may be implemented in client devices **110**.

[0039] Computer system **204** includes HTML graph extractor **206**. HTML graph extractor **206** converts raw HTML data into a graph representation. In an illustrative embodiment, the graph extractor converts raw HTML data into an HTML graph (e.g., heterogenous graph). Each HTML element (e.g., tag) in the graph is represented by a node, and relationships among the elements are represented by meta-paths (e.g., edges or connections).

[0040] Computer system **204** includes feature extractor **208**. Feature extractor **208** creates a feature set for each node in the HTML graph. The feature set may be represented by vectors which include: (1) location/position features (e.g., coordinates or relative positions); (2) content features (e.g., embeddings representing text content); and (3) property features (e.g., attributes from HTML/CSS).

[0041] Computer system **204** includes area of interest (AOI) identifier **210** configured to identify specific regions, sub-structures or nodes within the HTML graph that are of interest to a specific organization or entity. The specific regions or sub-structures may be identified from the feature sets created by feature extractor **208**. The specific regions or sub-structures may include headers, menus, tables, main content, sidebars, etc. AOI identifier **210** filters out non-relevant nodes. The non-relevant nodes are considered not of interest to the specific organization or entity.

[0042] Computer system **204** includes sub-structure analyzer **212** configured to refine the identified areas of interest further using meta-path based models. Sub-structure analyzer **212** identifies patterns or sequences of node types and relationships, helping to understand the recurring structural patterns within the HTML graph. Sub-structure analyzer **212** segregates or isolates sub-structures having patterns or sequences.

[0043] Computer system **204** includes data extractor **214** which prioritizes and extracts data items from the refined areas of interest. The extracted data items are standardized and stored. The extracted data items can be stored, for example, as key-value pairs, tabular content (e.g., headers, table, footers) and pagination (e.g., URLs). The extracted data items may suggest additional data based on learned patterns or user-defined rules.

[0044] In some embodiments, computer system **204** includes change detector **216** configured to detect changes to the web pages. The changes may be defined as any modifications or updates to the web pages. For example, any modifications or changes in contents, data items, text, tables, headers, footers, etc., may be considered as changes to the web pages. The changes to the web pages can be detected from changes to the segregated or isolated sub-structures or changes to the extracted data items.

[0045] If significant changes to the web pages are detected, users are alerted through a user interface for feedback on the changes. Significant changes may be defined as changes that are considered crucial for organizational use cases. For example, if a web page publishes current mortgage interest rates or asset values of investment funds, changes in the mortgage interest rates or changes in the asset values may be considered significant changes. In contrast, insignificant changes may be defined as inconsequential alterations resulting from page geometry modifications or noise.

[0046] Using the new sub-structures, a machine learning model can be trained to form a new trained model object. The trained model object represents a mapping function which makes predictions or decisions on new or unseen sub-structures or data in the HTML graph.

[0047] In some example embodiments, various components of computer system **204** can be implemented in software, hardware, firmware of one or more combinations of software, hardware or firmware.

[0048] FIG. **3** illustrates system **300** for web data extraction in accordance with an illustrative embodiment. Initially, HTML file **302** which contains raw data is received. Next, graph extractor **304** converts the raw data of HTML file **302** into a graph representation. Graph extractor **304** parses raw HTML data and generates a network graph of HTML nodes (e.g., heterogenous graph). In the HTML graph, each HTML element (e.g., tags, headers, footers, tables, other content) is represented by a node, and relationships among the elements are represented by meta-paths (e.g., edges or connections), thus illustrating parent-child relationships in the graph.

[0049] Next, feature extractor **306** extracts an array of features from each node in the graph. The feature set may be represented by vectors which may include: (1) location/position vectors which capture element hierarchy and relation to other elements (e.g., coordinates or relative positions); (2) content vectors which capture content of tags (e.g., embeddings representing text content); (3) property vectors which provide information about tags, tag metadata, CSS attributes, etc.; (4) domain specific vectors which provide project specific features.

[0050] Next, area of interest (AOI) identifier **308** identifies specific regions, sub-structures or nodes within the HTML graph that are of interest to a specific organization. Area of interest identifier **308** uses information from the feature vectors and input from users to infer most probable regions, sub-structures or nodes of interest in the graph. For example, an organization may be interested in mortgage rates or certificate of deposit (CD) rates which are published on a web page. AOI identifier **308** identifies regions, sub-structures or nodes within the HTML graph that contain tags, headers, footers, tables and other content related to the subject matter of interest. AOI identifier **308** filters out non-relevant regions, sub-structures or nodes which are not of interest to the organization. AOI identifier **308** also classifies meta-data associated with identified areas of interest.

[0051] Sub-structure analyzer **310** refines the identified areas of interest further using meta-path based models. Sub-structure analyzer **310** identifies recurring patterns or sequences of sub-structures or node-types and relationships, helping to understand the structural patterns or sequences within the HTML graph.

[0052] In an illustrative embodiment, sub-structure analyzer **310** includes meta-path cluster generator **312** that generates element clusters which are then segregated or isolated to identify different content types. This segregation is done using the feature vectors and can represent collections of different content types. For example, element clusters may include collections of key-value pairs, tabular contents and paginations. Sub-structure analyzer **310** includes pipeline classifier **314** which classifies isolated clusters by content types.

[0053] Next, data extractor **316** extracts data items from the segregated or isolated sub-structures or clusters. The extracted data is prioritized based on their confidence metrices. Data extractor **316** includes data prioritization and/or standardization module **318** that prioritizes extracted data items based on relevance or importance to the specific organization. The extracted data items are standardized and stored, for example, as key-value pairs, tabular content (e.g., headers, table, footers) and pagination (e.g., URLs). Data extractor **316** includes similar data identifier **320** which identifies patterns or sequences with sub-structures similar to the extracted data items based on the clusters generated by cluster generator **312**. The patterns or sequences may be used to detect additional data based on learned patterns or user-defined rules.

[0054] In some embodiments, change detector **322** identifies significant changes and updates to the web pages based on user defined logic and feedback. The changes or updates to the web pages may result in changes to one or more segregated or isolated sub-structures in the HTML graph.

[0055] In some embodiments, change detector **322** detects changes or updates to the web pages by detecting changes or updates to the segregated or isolated sub-structures. In some embodiments, change detector **322** detects changes or updates to the web pages by detecting changes of updates in the extracted and stored data items.

[0056] Because the segregated or isolated sub-structures are identified as areas of interest, by detecting changes or updates to the segregated sub-structures, relevant or important changes or updates to the web pages are identified. Similarly, because stored data items are extracted from feature vectors of the segregated or isolated sub-structures and stored in a standardized format (e.g., table), by detecting changes or updates in the stored data items, relevant or important changes or updates to the web pages are identified.

[0057] For example, by analyzing the same sub-structure at two different dates, relevant or important changes to the web pages can be detected. A sub-structure S**1** may be analyzed on Jun. 1, 2023 and thereafter on Jul. 1, 2023. By detecting changes or differences between the sub-structure S**1** as it existed on Jul. 1, 2023 and the sub-structure S**1** as it existed on Jun. 1, 2023, changes to the web pages can be detected.

[0058] Similarly, by detecting the changes or differences in a table TI containing extracted data items associated with the sub-structure S**1** on Jun. 1, 2023 and the table TI on Jul. 1, 2023, relevant and important changes to the web pages can be detected.

[0059] Thus, instead of monitoring entire web pages to detect changes, specific regions or sub-structures that are areas of interest are monitored. The effect of this is that reduced processing resource and maintenance overhead are required to detect changes to the web pages. Furthermore, because only specific regions or sub-structures of interest are monitored for changes, only those changes that are relevant or important to an organization or entity are identified. As such, processing resources are not wasted on identifying changes to the web pages that may not be important to an organization or an entity.

[0060] In contrast, existing change detection methods monitor entire web pages by setting up crawlers which check each element in the web pages. As such, existing change detection methods require significant processing resource and high maintenance overhead. Furthermore, existing change detection methods frequently fail to distinguish between changes that are relevant or important an organization or an entity from those changes that are not of significance.

[0061] Referring again to FIG. **3**, in decision block **324**, if changes or updates to the web pages are determined to be significant, users are alerted via user interface **326** for feedback on the changes, and the changes and updates are stored in database **330**. Based on user feedback, machine learning model **328** can be trained to form a new trained model object. The trained model object represents a mapping function which makes predictions or decisions on new or unseen sub-structures or data in the HTML graph. The trained model object is used by data extractor for similar data identification and extraction. If there are no significant changes or updates present, the data is stored in database **330**.

[0062] FIG. **4** illustrates web data extraction from an HTML file by constructing an HTML graph in accordance with an illustrative embodiment. Initially, HTML file **402** is received. HTML graph extractor **304** converts HTML file **402** into HTML graph **404**. In HTML graph **404**, nodes **406**, which are illustrated as circles, represent tags and contents of HTML file **402**. For example, nodes **406** may represent one of the following tags:

[0063] <html>: html;

[0064] <div>: division;

[0065] <h1>: header;

[0066] <p>: paragraph;

[0067] <span>: span;

[0068] <ul>: list;

[0069] <li>: list item;

[0070] <table>: table;

[0071] <tr>: table row.

[0072] Feature extractor **306** creates feature set **408** for each node **406** in the HTML graph. Feature set **408** may be represented by vectors which may include: (1) location/position vectors (e.g., coordinates or relative positions); (2) content vectors (e.g., embeddings representing text content); and (3) property vectors (e.g., attributes from HTML/CSS).

[0073] In some embodiments, feature extractor **306** creates expressive features from the nodes and tags of the HTML graph. Feature extractor **306** is configured to capture information from text and domain in the graph, which are crucial for subsequent processes. In some embodiments, the feature extraction process starts with the parsing of the heterogeneous graph. The location/position vectors capture information of a node's position in the context of its ancestors, child nodes and descendants. The content vectors are encoded representations of text created using embeddings to capture the semantics of content present in the nodes. The property vectors capture an array of f attributes related to tag information, CSS and text features from tag parameters. The domain specific vectors such as content type, content length, specific Booleans further augment the information present at each node.

[0074] Area of interest (AOI) identifier **308** analyzes feature set **408** which are represented by vectors. Based on the analysis, AOI identifier **308** identifies specific regions, sub-structures or nodes **410** within the HTML graph that are of interest to a specific organization or an entity. For example, an organization may be interested in mortgage rates, certificate of deposit (CD) interest rates or commodity prices which are published on a web page. The AOI identifier identifies regions, sub-structures or nodes **410** within the HTML graph that contain tags, headers, footers, tables and other content related to the subject matter of interest. The AOI identifier filters out non-relevant regions, sub-structures or nodes **412**. Non-relevant regions or nodes **412** are considered not of interest to the organization or entity. Non-relevant nodes **412** are illustrated as solid dark circles.

[0075] In some embodiments, AOI identifier **308** uses ensemble classification models to categorize and rank relevant and irrelevant nodes in the graph. AOI identifier **308** extracts a latent representation of the multi-fold features along with the input application context which are used to identify corresponding regions, sub-structures or nodes of interest for the data items.

[0076] Next, sub-structure analyzer **310** refines the identified areas of interest further using meta-path based models. The sub-structure analyzer identifies recurring patterns or sequences of node-types and relationships, helping to understand the structural patterns or sequences within the HTML graph. In an illustrative embodiment, the sub-structure analyzer generates element clusters which are segregated or isolated by content types. For example, element clusters may be segregated by key-value pairs (e.g., **430**), tabular contents (e.g., **432**) and paginations (e.g., **434**).

[0077] In some embodiments, identified areas of interests are further processed to identify and rank subgraphs within the areas of interest. Sub-structure analyzer **310** builds ensemble models on known meta-paths and uses these models to find and rank most similar subgraphs or sub-structures to be passed on to specific extraction modules for subsequent data extraction process.

[0078] Next, data extractor **316** prioritizes and extracts data items from the isolated sub-structures or clusters. The extracted data items are standardized and stored, for example, as key-value pairs **440**, tabular content **442** (e.g., headers, table, footers) and pagination **444** (e.g., URLs).

[0079] In some embodiments, change detector **322** identifies changes to the web pages by monitoring changes to the segregated or isolated sub-graphs **430**, **432** and **434**. Also, changes to the web pages are identified by monitoring changes or updates to the stored data items **440**, **442** and **444**. If changes to the web pages are determined to require significant updates to the stored data items, users are alerted via a user interface for feedback on the changes. Based on user feedback, a machine learning model is trained to form a new trained model object. The trained model object represents a mapping function which makes predictions or decisions on new or unseen sub-structures or data in the HTML graph. The trained model object is used by data extractor for similar data identification and extraction.

[0080] Thus, instead of monitoring entire web pages to detect changes, specific regions or sub-structures that are areas of interest are monitored for any changes. According to other embodiments, segregated sub-graphs are monitored for any changes to the web pages. The effect of this is that reduced processing resource and maintenance overhead are required to detect changes to the web pages. Furthermore, because only specific regions or sub-structures of interest and segregated data items are monitored for changes, only those changes that are relevant or important to an organization or an entity are identified. As such, processing resources are not wasted on identifying changes to the web pages that may not be important to an organization or an entity.

[0081] FIGS. **5**A and **5**B illustrate an example web data extraction in accordance with illustrative embodiments. In FIG. **5**A, web page **500** provides deposit rate information which is identified as areas of interest. Web page **500** displays table **502** which includes title **504**, header **506**, table contents **508** and footer **510**.

[0082] FIG. **5**B shows corresponding graph **510** of table **502**. Graph **510** is broken into sub-graphs and meta-paths showing relationships among sub-graphs and nodes. Sub-graph **514** represents table **502**. Inside sub-graph **514**, node **516** represents header **506**, node **518** represents footer **510**, and sub-graph **520** represents table contents **508**. These areas of interest are passed on for sub-structure analysis and data extraction.

[0083] FIGS. **6**A and **6**B illustrate an example web data extraction in accordance with illustrative embodiments. FIG. **6**A shows web page **600** which provides information about an investment fund ("iShares Investment Fund"). This information is indicated as "Key Facts" in box **602**.

[0084] FIG. **6**B shows corresponding graph **610** of the contents of box **602**. In graph **610**, contents of box **602** are shown by a plurality of nodes. Root node **612** (division tag) represents box **602**. Other nodes **604**, **606**, **608**, **612**, which are span tags or division tags, represent other elements inside box **602**. In some embodiments, data extraction models are used to extract key-value pairs in sub-graphs **620**, **622** and **624**. Sub-graphs **620**, **622** and **624** have similar structures because they represent rows in box **602**. As such, these sub-graphs can be used to extract similar data items from other areas of graph **610**.

[0085] FIGS. **7**A-**7**B illustrate another example web data extraction in accordance with illustrative embodiments. FIG. **7**A shows web page **700** and a corresponding HTML graph **720** is shown in FIG. **7**B. Web page **700** includes table **702** having rows **704**, **706** and pagination box **708**.

[0086] In graph **720**, root node **722** represents table **702**. Sub-graph **724** represents row **704**, and sub-graph **726** represents row **706**. Sub-graph **728** represents pagination box **708**. In sub-graph **728**, pagination box and page numbers are represented by list nodes <ul> and list elements <li>. Because sub-graphs **724** and **726** represent rows, they have similar structures. By analyzing sub-graph **724**, similar sub-structures such as sub-graph **726** can be identified.

[0087] Although, web page **700** shows table **702**, the corresponding graph **720** does not contain table tags <table> because the developer of web page **700** opted to use division tags <div> and anchor tags <a> instead of using table tags <table> to construct table **702**. Due to the flexible nature of HTML syntax, which allows variations, division and anchor tags can be used instead of table tags to construct a table on web page **700**.

[0088] FIGS. **8**A-**8**B illustrate user interface **800** showing data extracted from a web page and detection of significant changes. User interface **800** shares insight from significant change detection with users. Extracted items from an HTML graph are displayed in mapping records, and extracted tables and corresponding metadata are presented in a bottom pane. The location of extracted items is highlighted in the table pane. Any significant changes found in non-tabular metadata content are highlighted for user's analysis. A user interprets the changes and adds appropriate updates to the items.

Feedback to a machine learning model is submitted by selecting appropriate checkboxes and clicking a Submit button.

[0089] FIGS. **9**A-**9**B illustrate user interface **900** showing data extracted from a web page and detection of new data items in table **902**. In table **902**, mapped items are extracted from a web page and new row **904** is added. A user can interpret this change in table **902** and add a new product to mapping records. Thus, table **902** shows previously mapped items and the new row. Feedback to a machine learning model is submitted by selecting appropriate checkboxes and clicking a Submit button.

[0090] FIG. **10** illustrates user interface **1000** showing extracted data **1002** from a web page and change detection view **1002**. User interface **1000** allows custom date selection for the changes. Relevant statistics and changes in the data items are highlighted. A user can validate the changes and add appropriate updates to be stored in a database as feedback. A user can highlight any incorrect extraction submit feedback to the machine learning model by selecting appropriate checkboxes and clicking a Submit button.

[0091] With reference next to FIG. **11**, a flowchart of process **1100** for a computer-implemented method for web data extraction is provided. Process begins at step **1102**. An HTML file containing HTML data is received (step **1104**). The HTML data is converted into an HTML graph by an HTML graph extractor (step **1106**). In the HTML graph, elements in the HTML file are represented by nodes and relationships among the elements are represented by metapaths.

[0092] Next, a feature set for the nodes in the HTML graph are generated by a feature extractor (step **1108**). The feature set may be represented by vectors which may include: (1) location/position vectors (e.g., coordinates or relative positions); (2) content vectors (e.g., embeddings representing text content); and (3) property vectors (e.g., attributes from HTML/CSS). In some embodiments, the vectors are represented by N bits of binary values.

[0093] Next, areas of interest in the HTML graph are identified by an area of interest (AOI) identifier based on the feature set of the nodes (step **1110**). The identified areas of interest are then refined by a sub-structure analyzer by segregating or isolating sub-structures having recurring patterns or sequences (step **1112**). Next, data items from the segregated or isolated sub-structures are extracted by a data extractor and stored in a standardized format (step **1114**). The extracted data items are standardized and stored, for example, as key-value pairs, tabular content (e.g., headers, table, footers) and pagination (e.g., URLs).

[0094] With reference next to FIG. **12**, a flowchart of process **1200** for detecting changes or updates to web pages is provided. Data items from segregated and or isolated sub-structures are extracted and stored by a data extractor (step **1202**). Next, a change detector identifies changes and updates to the web pages based on user defined logic and feedback (step **1204**). In some embodiments, the change detector detects changes or updates to the web pages by detecting changes or updates to the segregated or isolated sub-structures. In some embodiments, the change detector detects changes or updates to the web pages by detecting changes of updates in the extracted and stored data items.

[0095] In decision block **1206**, if changes or updates to the web pages are determined to be significant, users are alerted via a user interface (step **1208**) for feedback on the changes, and the changes and updates are stored in a database (step **1210**). Based on user feedback, a machine learning model is trained to form a new trained model object (step **1212**). The trained model object represents a mapping function which makes predictions or decisions on new or unseen substructures or data in the HTML graph. If there are no significant changes or updates present, the data is stored in the database (step **1214**).

[0096] Turning now to FIG. **13**, an illustration of a block diagram of a data processing system is depicted in accordance with an illustrative embodiment. Data processing system **1300** may be used to implement server computers **104** and **106** and client devices **110** in FIG. **1**, as well as computer system **200** in FIG. **2**. In this illustrative example, data processing system **1300** includes communications framework **1302**, which provides communications between processor unit **1304**, memory **1306**, persistent storage **1308**, communications unit **1310**, input/output unit **1312**, and display **1314**. In this example, communications framework **1302** may take the form of a bus system.

[0097] Processor unit **1304** serves to execute instructions for software that may be loaded into memory **1106**. Processor unit **1304** may be a number of processors, a multiprocessor core, or some other type of processor, depending on the particular implementation. In an embodiment, processor unit **1304** comprises one or more conventional general-purpose central processing units (CPUs). In an alternate embodiment, processor unit **1304** comprises one or more graphical processing units (GPUS).

[0098] Memory **1306** and persistent storage **1308** are examples of storage devices **1316**. A storage device is any piece of hardware that is capable of storing information, such as, for example, without limitation, at least one of data, program code in functional form, or other suitable information either on a temporary basis, a permanent basis, or both on a temporary basis and a permanent basis. Storage devices **1316** may also be referred to as computer-readable storage devices in these illustrative examples. Memory **1306**, in these examples, may be, for example, a random access memory or any other suitable volatile or non-volatile storage device. Persistent storage **1308** may take various forms, depending on the particular implementation.

[0099] For example, persistent storage **1308** may contain one or more components or devices. For example, persistent storage **1308** may be a hard drive, a flash memory, a rewritable optical disk, a rewritable magnetic tape, or some combination of the above. The media used by persistent storage **1308** also may be removable. For example, a removable hard drive may be used for persistent storage **1308**. Communications unit **1310**, in these illustrative examples, provides for communications with other data processing systems or devices. In these illustrative examples, communications unit **1310** is a network interface card.

[0100] Input/output unit **1312** allows for input and output of data with other devices that may be connected to data processing system **1300**. For example, input/output unit **1312** may provide a connection for user input through at least one of a keyboard, a mouse, or some other suitable input device. Further, input/output unit **1312** may send output to a printer. Display **1314** provides a mechanism to display information to a user.

[0101] Instructions for at least one of the operating systems, applications, or programs may be located in storage devices **1316**, which are in communication with processor unit **1304** through communications framework **1302**. The processes of the different embodiments may be performed by processor unit **1304** using computer-implemented instructions, which may be located in a memory, such as memory **1306**.

[0102] These instructions are referred to as program code, computer-usable program code, or computer-readable program code that may be read and executed by a processor in processor unit **1304**. The program code in the different embodiments may be embodied on different physical or computer-readable storage media, such as memory **1306** or persistent storage **1308**.

[0103] Program code **1318** is located in a functional form on computer-readable media **1320** that is selectively removable and may be loaded onto or transferred to data processing system **1300** for execution by processor unit **1104**. Program code **1318** and computer-readable media **1320** form computer program product **1322** in these illustrative examples. In one example, computer-readable media **1320** may be computer-readable storage media **1324** or computer-readable signal media **1326**.

[0104] In these illustrative examples, computer-readable storage media **1324** is a physical or tangible storage device used to store program code **1318** rather than a medium that propagates or transmits program code **1318**. Computer readable storage media **1324**, as used herein, is not to be construed as being transitory signals per se, such as radio waves or other freely propagating electromagnetic waves, electromagnetic waves propagating through a waveguide or other transmission media (e.g., light pulses passing through a fiber-optic cable), or electrical signals transmitted through a wire, as used herein, is not to be construed as being transitory signals per se, such as radio waves or other freely propagating electromagnetic waves, electromagnetic waves propagating through a waveguide or other transmission media (e.g., light pulses passing through a fiber-optic cable), or electrical signals transmitted through a wire.

[0105] Alternatively, program code **1318** may be transferred to data processing system **1300** using computer-readable signal media **1326**. Computer-readable signal media **1326** may be, for example, a propagated data signal containing program code **1318**. For example, computer-readable signal media **1326** may be at least one of an electromagnetic signal, an optical signal, or any other suitable type of signal. These signals may be transmitted over at least one of communications links, such as wireless communications links, optical fiber cable, coaxial cable, a wire, or any other suitable type of communications link.

[0106] The different components illustrated for data processing system **1300** are not meant to provide architectural limitations to the manner in which different embodiments may be implemented. The different illustrative embodiments may be implemented in a data processing system including components in addition to or in place of those illustrated for data processing system **1100**. Other components shown in FIG. **13** can be varied from the illustrative examples shown. The different embodiments may be implemented using any hardware device or system capable of running program code **1318**.

[0107] As used herein, "a number of," when used with reference to items, means one or more items. For example, "a number of different types of networks" is one or more different types of networks.

[0108] Further, the phrase "at least one of," when used with a list of items, means different combinations of one or more of the listed items can be used, and only one of each item in the list may be needed. In other words, "at least one of" means any combination of items and number of items may be used from the list, but not all of the items in the list are required. The item can be a particular object, a thing, or a category.

[0109] For example, without limitation, "at least one of item A, item B, or item C" may include item A, item A and item B, or item B. This example also may include item A, item B, and item C or item B and item C. Of course, any combinations of these items can be present. In some illustrative examples, "at least one of" can be, for example, without limitation, two of item A; one of item B; and ten of item C; four of item B and seven of item C; or other suitable combinations.

[0110] The flowcharts and block diagrams in the different depicted embodiments illustrate the architecture, functionality, and operation of some possible implementations of apparatuses and methods in an illustrative embodiment. In this regard, each block in the flowcharts or block diagrams can represent at least one of a module, a segment, a function, or a portion of an operation or step. For example, one or more of the blocks can be implemented as program code, hardware, or a combination of the program code and hardware. When implemented in hardware, the hardware may, for example, take the form of integrated circuits that are manufactured or configured to perform one or more operations in the flowcharts or block diagrams. When implemented as a combination of program code and hardware, the implementation may take the form of firmware. Each block in the flowcharts or the block diagrams may be implemented using special purpose hardware systems that perform the different operations or combinations of special purpose hardware and program code run by the special purpose hardware.

[0111] In some alternative implementations of an illustrative embodiment, the function or functions noted in the blocks may occur out of the order noted in the figures. For example, in some cases, two blocks shown in succession may be performed substantially concurrently, or the blocks may sometimes be performed in the reverse order, depending upon the functionality involved. Also, other blocks may be added in addition to the illustrated blocks in a flowchart or block diagram.

[0112] The different illustrative examples describe components that perform actions or operations. In an illustrative embodiment, a component may be configured to perform the action or operation described. For example, the component may have a configuration or design for a structure that provides the component an ability to perform the action or operation that is described in the illustrative examples as being performed by the component.

[0113] Many modifications and variations will be apparent to those of ordinary skill in the art. Further, different illustrative embodiments may provide different features as compared to other illustrative embodiments. The embodiment or embodiments selected are chosen and described in order to best explain the principles of the embodiments, the practical application, and to enable others of ordinary skill in the art to understand the disclosure for various embodiments with various modifications as are suited to the particular use contemplated.

What is claimed is:

1. A computer-implemented method for web data extraction, comprising:

receiving an HTML file containing HTML data;

converting the HTML data into an HTML graph, wherein elements in the HTML file are represented by nodes in the HTML graph and relationships among the elements are represented by meta-paths;

generating feature sets for the nodes in the HTML graph;

identifying areas of interest in the HTML graph based on the feature sets of the nodes;

refining the identified areas of interest by segregating sub-structures having recurring patterns or sequences; and

extracting data items from the segregated sub-structures and storing the extracted data items.

2. The method of claim **1**, wherein the feature sets are represented by vectors, and wherein the vectors include one or more of:

location or position vectors;

content vectors;

property features; and

domain specific vectors.

3. The method of claim **1**, further comprising detecting changes to web pages by detecting changes to the extracted data items from the segregated sub-structures, wherein the changes to the web pages are modifications or updates to the web pages.

4. The method of claim **1**, wherein the extracted data items are standardized and stored as one or more of:

key-value pairs;

tabular content; and

pagination.

5. The method of claim **1**, further comprising identifying the areas of interest in the HTML graph using input from users to infer probable regions or nodes that are of interest.

6. The method of claim **1**, further comprising:

determining if the changes to web pages are significant changes; and

if there are significant changes, notifying users via a user interface.

7. The method of claim **6**, further comprising:

training a machine learning model using the significant changes to form a trained model object; and

segregating the sub-structures using the trained model object.

8. The method of claim **2**, wherein the location or position vectors comprise coordinates or relative positions of nodes in the HTML graph.

9. The method of claim **2**, wherein the content vectors comprise embeddings representing text content of nodes in the HTML graph.

10. The method of claim **2**, wherein the property vectors comprise attributes extracted from the nodes in the HTML graph.

11. A system for web data extraction, the system comprising:

a storage device configured to store program instructions; and

one or more processors operably connected to the storage device and configured to execute the program instructions to cause the system to:

receive an HTML file containing HTML data;

convert the HTML data into an HTML graph, wherein elements in the HTML file are represented by nodes in the HTML graph and relationships among the elements are represented by meta-paths;

generate feature sets for the nodes in the HTML graph;

identify areas of interest in the HTML graph based on the feature sets of the nodes;

refine the identified areas of interest by segregating sub-structures having recurring patterns or sequences; and

extract data items from the segregated sub-structures and store the extracted data items.

12. The system of claim **11**, wherein the processors further execute instructions to represent the feature sets by vectors, and wherein the vectors include one or more of:

location or position vectors;

content vectors;

property vectors; and

domain specific vectors.

13. The system of claim **11**, wherein the processors further execute instructions to detect changes to web pages by detecting changes to the extracted data items from the segregated sub-structures, wherein the changes to the web pages are modifications or updates in the web pages.

14. The system of claim **11**, wherein the processors further execute instructions to standardize and store the extracted data items as one or more of:

key-value pairs;

tabular content; and

pagination.

15. The system of claim **11**, wherein the processors further execute instructions to identify the areas of interest in the HTML graph using input from users to infer probable regions or nodes that are of interest to an organization.

16. The system of claim **11**, wherein the processors further execute instructions to:

determine if the changes to web pages are significant changes; and

if there are significant changes, notify users via a user interface.

17. The system of claim **16**, wherein the processors further execute instructions to:

train a machine learning model using the significant changes to form a trained model object; and

segregate the sub-structures using the trained model object.

18. A computer program product for web data extraction, the computer program product comprising:

a computer-readable storage medium having program instructions embodied thereon to perform the steps of:

receiving an HTML file containing HTML data;

converting the HTML data into an HTML graph, wherein elements in the HTML file are represented by nodes in the HTML graph and relationships among the elements are represented by meta-paths;

generating feature sets for the nodes in the HTML graph;

identifying areas of interest in the HTML graph based on the feature sets of the nodes;

refining the identified areas of interest by segregating sub-structures having recurring patterns or sequences; and

extracting data items from the segregated sub-structures and storing the extracted data items.

19. The computer program product of claim **18**, further comprising instructions for detecting changes to web pages by detecting changes to the extracted data items, wherein the changes to the web pages are modifications or updates in the web pages.

20. The computer program product of claim **18**, wherein the feature sets are represented by vectors, and wherein the vectors include one or more of:

location or position vectors;
content vectors;
property vectors; and
domain specific vectors.

21. The computer program product of claim **18**, wherein the extracted data items are standardized and stored as one or more of:

key-value pairs;
tabular content; and
pagination.

22. The computer program product of claim **18**, further comprising instructions for identifying the areas of interest in the HTML graph using input from users to infer probable regions or nodes that are of interest to an organization.

23. The computer program product of claim **19**, further comprising instructions for determining:

if the changes to web pages are significant changes; and
if there are significant changes, notifying users via a user interface.

24. The computer program product of claim **18**, further comprising instructions for:

training a machine learning model using the significant changes to form a trained model object; and
segregating the sub-structures using the trained model object.

* * * * *