DATA EXTRACTION FROM UNSTRUCTURED FINANCIAL DOCUMENTS – THE DATA FACTORY

BOSTON, OCTOBER 1ST, 2013

GEORGE ROTH – RECOGNOS INC, SAN FRANCISCO, CA, USA
DREW WARREN – RECOGNOS FINANCIAL, NEW YORK, USA
DIA TRAMBITAS – RECOGNOS, CLUJ, ROMANIA – SEMANTIC TECHNOLOGY
IOANA BARBANTAN – RECOGNOS, CLUJ, ROMANIA – MACHINE LEARNING
IULIAN MARGARINTESCU – RECOGNOS CLUJ, ROMANIA – SYSTEM ARCHITECT
WHAT IS THE DATA FACTORY

- The “data factory” is a complex hybrid system
- Software systems together with some manual exception processing
- Used to extract data from a collection of un-formatted or semi-formatted documents
- Data Factory is designed in such a way that is improved continuously based on optimizations that are done using (the number and role of the human participants decreases in time.)
- The product of the data factory consist of Data Feeds, e.g. Mutual Fund Data, Vendor Contract Data, Security Master Data, Issuer Data, etc.
- The users of the data feeds are receiving the data based on subscriptions, using the chosen delivery method.
THE COMPONENTS OF THE DATA FACTORY

- **Data Sources** that provides access to the documents used in the extraction process (“raw materials”)
- **Content Management Collaboration Environment** - that initiates the workflows associated with the extraction process, also providing work queues where the designated human workforce performs tasks related to: exception handling, Q&A process, manual overrides, approvals, etc.
- **Databases** where the extracted data together with the source documents are stored (we are using Raven DB – non SQL database – hosted on Amazon)
- **Extraction software** – combination of multiple technologies that is used in the extraction process: NLP software, machine learning software, regular expression processing
- **API** – use to deliver the extracted data – can be a REST based, batch delivery (XLS, XML, CSV, etc.)
- **UI** – used to browse and query the document collection together with the extracted data
THE TEAM: DATA FACTORY WORKERS

- **Data Feed Manager** – each product of the Data Factory (i.e. a Data Feed) has a Data feed Manager

- **Data Analyst – Subject Expert** – a business analyst that has expertise in the documents and associated business processes (e.g. Mutual Fund Data Analyst, SEC Documents Expert, etc.).

- **Data Analysts** – business people that are trained to write NLP extraction scripts and also understand the data. They contribute to the setup phase - *write the NLP extraction scripts* - and to the maintenance phase - *fulfill computer assisted tasks related to exception processing, issue resolution, etc.*

- **Software Engineers** – work on setting up the production line for a Data Feed, being in charge to assemble/customize the software components used by the Data Factory. The role of the software engineers decreases over time after the data factory is in production.
HOW TO SETUP A NEW “PRODUCTION LINE”

Setup a new Data Production line - Analysis

- Document Samples
  - Highlights for fields
- Data Analyst – Subject Expert
- Fields to be extracted – data feed taxonomy
- Data Analysts – Write NLP, regular expressions, Scripts
- Meta Data for the CMS – Work Flow (Work Queues, Scripts)
- Hire people
  - Train people – transfer business knowledge
- Data Feed Manager
- New Data Analysts – work with the CMS
  - QA Tools
How does the production line work

The documents are received from the source (e.g., EDGAR) if needed OCR is used.

Data Normalization (Canonical Form)

Normalized Documents

Data Extraction

Extraction Exceptions (not found)

QA 1

Offshore QA

Offshore Data Analysts (their number is reduced in time)

QA 1

Final US QA

New Data Analysts – work with the CMS

QA Tools

Data Delivery (API – REST – JSON – files)

Data Analysts – Write NLP, regular expressions, Scripts
CASE STUDY


- 20,000 Mutual Funds – 2 Million Data Points

- Changing every year – backlog – one year

- Validated the existing “Profile Data” – created through Crowd Sourcing of the funds

- Found errors in filings – refilling

- Duration – 6 months

- 10 people involved in the factory
### Sample Taxonomy (Fragment)

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>str_exchange_buy_schedule_code</td>
<td>yes</td>
<td>Send</td>
</tr>
<tr>
<td>str_forfeitures_distribution_waiver_code</td>
<td>yes</td>
<td>Send</td>
</tr>
<tr>
<td>str_hardship_distribution_waiver_code</td>
<td>yes</td>
<td>Send</td>
</tr>
<tr>
<td>str_involuntary_distribution_waiver_code</td>
<td>yes</td>
<td>Send</td>
</tr>
<tr>
<td>str_loan_distribution_waiver_code</td>
<td>yes</td>
<td>Send</td>
</tr>
<tr>
<td>str_minimum_amt</td>
<td>yes</td>
<td>Send</td>
</tr>
<tr>
<td>str_payout_calculation_amt_code</td>
<td>no</td>
<td>NID</td>
</tr>
<tr>
<td>str_plan_fee_waiver_code</td>
<td>yes</td>
<td>Send</td>
</tr>
<tr>
<td>str_qualified_distribution_waiver_age</td>
<td>yes</td>
<td>Send</td>
</tr>
<tr>
<td>str_qualified_distribution_waiver_code</td>
<td>yes</td>
<td>Send</td>
</tr>
<tr>
<td>str_rate_[0]</td>
<td>yes</td>
<td>Send</td>
</tr>
</tbody>
</table>
GOALS:

- Structure SEC filings in order to allow an easier and customized navigation within documents;
  - Interactive table of contents;
  - Links between funds, share classes and relevant chapters;

- Extract financial data points from Prospectus/SAI/(Sticker?) documents;
  - Better and faster exploitation of information;
  - Comparison of extracted data points with data from Profile;
  - Structured data export to different formats → support for user defined analysis on top of the extracted data;
  - Maintain provenance link
THE ANALYSIS PROCESS

1. Daily import - new filings are imported from the SEC platform;

2. Each filing (package containing several documents) is split into documents;

3. XBRL, Prospectus (485BPOS), SAI(497) and Sticker (497) documents are further analyzed:
   a) XBRL is automatically parsed and extracted data can be visualized using a dedicated user interface (www.mfip.info)
   b) Working on a query system using JSONiq
   c) Prospectus, SAI and Sticker documents go through an iterative semi-automatic processing chain that allows their 2 level structuring:
      - Division into chapters for an easy navigation;
      - Data extraction for an easy data exploitation.
DOCUMENT STRUCTURING – PHASE 1

- **Parsing**
  - The filing is divided into Document Parts, Sections and Chapters;
  - Funds described in each Document Part are identified together with their list of share classes.

- **Linking**
  - Links between sections and share classes are created

- **Validation**
  - The two previous steps are checked by a human expert and corrected if necessary.
The number and types of Document Parts contained within a filing depends on the filing class. There are currently 10 document types available: N-CSR, N-CSRS, SAI Sticker, SAI, Summary Sticker, Summary Prospectus, Statutory Prospectus, Statutory Sticker, Statutory Prospectus, N-Q, 24F-2NT.

**Original SEC Filing** - has a filing class such as 485APOS, 485BPOS, 497, 497K, etc.

**Document Part 1**

- e.g. All the Statutory Prospectuses identified within the current filing

**Section 1.1**

- e.g One Statutory Prospectus identified within the current document part

**Fund 1.1**

**Chapter list 1.1**

**Share class list 1.1**

**Document Part k**

- e.g. All the SAI identified within the current filing

- e.g One SAI identified within the current document part

**Section k.1**

- e.g All the Summary Prospectuses identified within the current filing

**Section k.r**

**Chapter list k.m**

**Fund k.1**

**Share class list k.1**

**Fund k.2**

**Share class list k.2**
Table of contents automatically detected using available templates

Navigable links are automatically created between the chapter name and the chapter start index in the document.
**DOCUMENT STRUCTURING – PHASE 1**

- **Parsing**
  - The filing is divided into Document Parts, Sections and Chapters;
  - Funds described in each Document Part are identified together with their list of share classes.

- **Linking**
  - Links between sections and share classes are created.

- **Validation**
  - The two previous steps are checked by a human expert and corrected if necessary.
PHASE 1 – THE LINKING PROCESS

From a flatter structure towards...
PHASE 1 – THE LINKING PROCESS

A hierarchical structure

Document Part 1

- All the Statutory Prospectuses identified within the current filing

Section 1.1

- One Statutory Prospectus identified within the current document part

Section 1.n

- One Summary Prospectus identified within the current document part

Section k.1

- One SAI identified within the current document part

Section k.r

- All the SAI s identified within the current filing

Fund 1.1.1

Share class list 1.1.1

... Fund 1.1.k

Share class list 1.1.k

Chapter list 1.n

... Chapter list k.1

Fund k.1.1

Share class list k.1.1

... Fund k.1.2

Share class list k.1.2

Chapter list k.1

... Chapter list k.m

Fund k.m.1

Share class list k.m.1

Fund 1.n.k

Share class list 1.n.k

Fund 1.n.1

Share class list 1.n.1

... Fund 1.n.k

Share class list 1.n.k

... Fund k.m.1

Share class list k.m.1

...
Each section has associated a list of share classes.
DOCUMENT STRUCTURING – PHASE 1

- **Parsing**
  - The filing is divided into Document Parts, Sections and Chapters;
  - Funds described in each Document Part are identified together with their list of share classes.

- **Linking**
  - Links between sections and share classes are created

- **Validation**
  - The two previous steps are checked by a human expert and corrected if necessary.
Phase 2 – Data Extraction

2. Data extraction from Prospectus, SAI and Sticker documents:

- Analysis of complex documents (200-300 pages of unstructured English text) and extraction of relevant features of the mutual funds
  - The extraction concentrates on a taxonomy of 260 data points representing features or characteristics of mutual funds

- A hybrid data extraction approach
  - Rule based linguistic analysis
  - Custom extracted - data propagation algorithm
    - Check data point context from historic document and propagate value if contexts match
  - Custom table parsers based on
    - Regular expressions
    - Machine learning
Step 1 - Filings are parsed → Prospectus and SAI doc linked with the Share Classes they describe

HTML Filing

Acc nb: 0000950123-12-011157

CLASS Y; IVLCX / Class C; IENAX / Class A; CLASS C; IENIX / Class B; CLASS B; CLASS Y; IAUYX / Class Y; MGAVX / CLASS B; ITHCX / Class C; CLASS C; ITYAX / Class A; FLISX / Investor Class; CLASS B; CLASS R5; IEFCX / Class C; CLASS A; MSVCX / CLASS C; CLASS C; ILSBX / Class B; ILSYX / Class Y; CLASS A; MSAJX / CLASS R5; CLASS B; IENIX / CLASS R5; CLASS B; MSAVX / CLASS A; MSARX / CLASS R; FSTUX / Investor Class; ITYYX / Class Y; CLASS A; FGLDX / Investor Class; IGDAX / Class A; CLASS Y; IENYX / Class Y; ITYBX / Class B; IENBX / Class B; IUTCX / Class C; Class R; CLASS R5; CLASS Y; IGDYX / Class Y; FTPIX / CLASS R5; IGDCX / Class C; FSTEX / Investor Class; ILSRX / Class R; MSAIX / CLASS Y; CLASS B; CLASS R5; IGBDX / Class B; IAUTX / Class A; CLASS R; ILSAX / Class A; FSIUX / CLASS R5; CLASS R; CLASS Y; CLASS C; CLASS A; FTCHX / Investor Class; (61 Share Class IDs)

Prospectus 1

IENAX / Class A; IENBX / Class B; FSTEX / Investor Class; IENYX / Class Y; IEFCX / Class C; CLASS C; CLASS A; CLASS B; CLASS Y;

IENAX / Class A; IENBX / Class B; FSTEX / Investor Class; IENYX / Class Y; IEFCX / Class C; CLASS C; CLASS A; CLASS B; CLASS Y;

IENAX / Class A; IENBX / Class B; FSTEX / Investor Class; IENYX / Class Y; IEFCX / Class C; CLASS C; CLASS A; CLASS B; CLASS Y;

IENAX / Class A; IENBX / Class B; FSTEX / Investor Class; IENYX / Class Y; IEFCX / Class C; CLASS C; CLASS A; CLASS B; CLASS Y;

IENAX / Class A; IENBX / Class B; FSTEX / Investor Class; IENYX / Class Y; IEFCX / Class C; CLASS C; CLASS A; CLASS B; CLASS Y;

IENAX / Class A; IENBX / Class B; FSTEX / Investor Class; IENYX / Class Y; IEFCX / Class C; CLASS C; CLASS A; CLASS B; CLASS Y;

IENAX / Class A; IENBX / Class B; FSTEX / Investor Class; IENYX / Class Y; IEFCX / Class C; CLASS C; CLASS A; CLASS B; CLASS Y;

IENAX / Class A; IENBX / Class B; FSTEX / Investor Class; IENYX / Class Y; IEFCX / Class C; CLASS C; CLASS A; CLASS B; CLASS Y;

IENAX / Class A; IENBX / Class B; FSTEX / Investor Class; IENYX / Class Y; IEFCX / Class C; CLASS C; CLASS A; CLASS B; CLASS Y;

IENAX / Class A; IENBX / Class B; FSTEX / Investor Class; IENYX / Class Y; IEFCX / Class C; CLASS C; CLASS A; CLASS B; CLASS Y;

IENAX / Class A; IENBX / Class B; FSTEX / Investor Class; IENYX / Class Y; IEFCX / Class C; CLASS C; CLASS A; CLASS B; CLASS Y;

IENAX / Class A; IENBX / Class B; FSTEX / Investor Class; IENYX / Class Y; IEFCX / Class C; CLASS C; CLASS A; CLASS B; CLASS Y;

IENAX / Class A; IENBX / Class B; FSTEX / Investor Class; IENYX / Class Y; IEFCX / Class C; CLASS C; CLASS A; CLASS B; CLASS Y;

IENAX / Class A; IENBX / Class B; FSTEX / Investor Class; IENYX / Class Y; IEFCX / Class C; CLASS C; CLASS A; CLASS B; CLASS Y;

IENAX / Class A; IENBX / Class B; FSTEX / Investor Class; IENYX / Class Y; IEFCX / Class C; CLASS C; CLASS A; CLASS B; CLASS Y;

IENAX / Class A; IENBX / Class B; FSTEX / Investor Class; IENYX / Class Y; IEFCX / Class C; CLASS C; CLASS A; CLASS B; CLASS Y;

IENAX / Class A; IENBX / Class B; FSTEX / Investor Class; IENYX / Class Y; IEFCX / Class C; CLASS C; CLASS A; CLASS B; CLASS Y;

IENAX / Class A; IENBX / Class B; FSTEX / Investor Class; IENYX / Class Y; IEFCX / Class C; CLASS C; CLASS A; CLASS B; CLASS Y;

IENAX / Class A; IENBX / Class B; FSTEX / Investor Class; IENYX / Class Y; IEFCX / Class C; CLASS C; CLASS A; CLASS B; CLASS Y;

IENAX / Class A; IENBX / Class B; FSTEX / Investor Class; IENYX / Class Y; IEFCX / Class C; CLASS C; CLASS A; CLASS B; CLASS Y;

IENAX / Class A; IENBX / Class B; FSTEX / Investor Class; IENYX / Class Y; IEFCX / Class C; CLASS C; CLASS A; CLASS B; CLASS Y;

IENAX / Class A; IENBX / Class B; FSTEX / Investor Class; IENYX / Class Y; IEFCX / Class C; CLASS C; CLASS A; CLASS B; CLASS Y;

IENAX / Class A; IENBX / Class B; FSTEX / Investor Class; IENYX / Class Y; IEFCX / Class C; CLASS C; CLASS A; CLASS B; CLASS Y;

IENAX / Class A; IENBX / Class B; FSTEX / Investor Class; IENYX / Class Y; IEFCX / Class C; CLASS C; CLASS A; CLASS B; CLASS Y;

IENAX / Class A; IENBX / Class B; FSTEX / Investor Class; IENYX / Class Y; IEFCX / Class C; CLASS C; CLASS A; CLASS B; CLASS Y;

IENAX / Class A; IENBX / Class B; FSTEX / Investor Class; IENYX / Class Y; IEFCX / Class C; CLASS C; CLASS A; CLASS B; CLASS Y;

IENAX / Class A; IENBX / Class B; FSTEX / Investor Class; IENYX / Class Y; IEFCX / Class C; CLASS C; CLASS A; CLASS B; CLASS Y;

IENAX / Class A; IENBX / Class B; FSTEX / Investor Class; IENYX / Class Y; IEFCX / Class C; CLASS C; CLASS A; CLASS B; CLASS Y;

IENAX / Class A; IENBX / Class B; FSTEX / Investor Class; IENYX / Class Y; IEFCX / Class C; CLASS C; CLASS A; CLASS B; CLASS Y;

IENAX / Class A; IENBX / Class B; FSTEX / Investor Class; IENYX / Class Y; IEFCX / Class C; CLASS C; CLASS A; CLASS B; CLAS
Step 2 - Prospectuses and SAIs are grouped based on the common share classes

Prospectus 1

IENAX / Class A; IENBX / Class B; FSTEX / Investor Class; IENYX / Class Y; IEFCX / Class C; CLASS C; CLASS A; CLASS B; CLASS Y;

SAI 1

ITHCX / Class C; ITYAX / Class A; ITYYX / Class Y; ITYBX / Class B; FTPIX / CLASS R5; FTCHX / Investor Class; IBUTX / Class B; IAUTX / Class A; FSIUX / CLASS R5; IVLCX / Class C; FLISX / Investor Class; ILSBX / Class B; ILSYX / Class Y; ILSRX / Class R; ILSAX / Class A; FGLDX / Investor Class; IGDAX / Class A; IGDYX / Class Y; IGDCX / Class C; IGDBX / Class B; IENAX / Class A; IEFCX / Class C; IENIX / CLASS R5; IENYX / Class Y; IENBX / Class B; FSTEX / Investor Class;

Prospectus 1

IENAX / Class A; IENBX / Class B; FSTEX / Investor Class; IENYX / Class Y; IEFCX / Class C; CLASS C; CLASS A; CLASS B; CLASS Y;

SAI 2

CLASS C; CLASS A; CLASS B; CLASS Y; MGAVX / CLASS B; MSVCX / CLASS C; MSAJX / CLASS R5; MSAVX / CLASS A; MSARX / CLASS R; MSAIX / CLASS Y; CLASS Y; CLASS A; CLASS C; CLASS B; CLASS R5; CLASS R; CLASS Y; CLASS C; CLASS R5; CLASS B; CLASS R; CLASS A; CLASS A; CLASS B; CLASS Y; CLASS C; CLASS C; CLASS B; CLASS C; CLASS A; CLASS R; CLASS Y; CLASS R5;
Step 3 – Paired Prospectus and SAI documents are automatically analyzed and a set of data points are extracted

Document Pair 1

Prospectus 1

SAI 1

NLP/Table Parser Extraction ENGINE

No linking between Extracted Data Points and Share Class ID!!!
Step 4 – Extracted data is validated, missing values are searched and added if available. All values are linked to Share Classes.
System overview
DATA EXTRACTION APPROACH

- Linguistic analysis
  - Cogito Studio – Expert System Italy
  - Extraction rules written in 4GL

- Custom table parsers
  - Based on syntactic rules
  - Based on machine learning

- Data propagation algorithm
  - Historic data – automatically or manually extracted – can be used to guide/improve the extraction process
    - If value together with its context is the same as in old document it is automatically propagated to new document, automatically linked and validated;
    - If value changed but context is the same, the new value is extracted automatically, linked and validated.
Example of text from which a feature is extracted – underlined in yellow
The extraction in previous examples was done using the following rule:

The IDs are used to identify the semantics associated to a word (the syncon of a word).

Several primitives are available that can refer to sets of words senses (syncons) with very close meanings.
The extraction results are visualized in a CMS by financial experts that can validate or modify the value.
Validated extraction results are delivered to our customers.
DATA POINTS CLASSIFICATION AND EXTRACTION FROM TABLES. A MACHINE LEARNING APPROACH
EXTRACT DATA POINTS FROM TABLES USING MACHINE LEARNING

- **Dataset creation and Model building**
  - Tables identification
  - Dataset construction from tables
  - Create models based on the dataset
    - Assign label Yes/No to each of the instances

- **Data classification**
  - Identify data point values based on the learned models
  - Performance evaluation
**TABLE IDENTIFICATION**

- Given as input a Prospectus, identify all tables
- Classify each table in one of the categories of interest: IRA, Fees and Expenses, Distribution Reinvestment
### Extract Context from Tables

#### Context Extraction
- **Identify horizontal context**
  - Characters and words at the left of the position of the value
- **Identify vertical context**
  - Define window surrounding the value
  - Select characters and words in the window above the value

#### Example Context
- The context in the example:
  \{account, Individual, Initial, investment, retirement\}
- After applying preprocessing and stemming:
  \{account, individu, init, investm, retir\}
TEXT PREPROCESSING

- Stop words removal
  - Words that carry no relevant information to our problem including conjunctions and prepositions

- Punctuation and numeric values elimination

- Stemming
  - Algorithm used: Lovins Stemmer
  - Algorithm description:
    - 294 endings, 29 conditions, 35 transformation rules
  - Steps:
    - Step 1: find the longest ending, associate it with a condition and remove it
    - Step 2: apply the transformation whether or not the ending was removed
  - Why use stemming:
    - Normalization step: different forms of a word are set to the same stem
    - Improve classification accuracy
    - Reduce number of attributes in the dataset
STEMMING EXAMPLE

Problem: identify the field Minimum_IRA_Initial_Purchase_Amount in the table:

- The definition of the field contains the words \{account, individual, initial, investment, retirement\}
  - Field In Table = \{accounts, individual, initial, minimum, no, retirement\}
  - The similarity between the definition and the field is: 2.23
- After applying the stemmer:
  - Definition of field = \{account, individu, init, investm, retir\}
  - Field In Table = \{account, individu, init, minim, retir, no\}
  - The similarity between the definition and the field becomes: 1.71 \(\Rightarrow\) they are more similar
CONTEXT AS INSTANCE

- Context \{account, individu, init, investm, retir\}

Minimum IRA Initial Purchase Amount
EUCLIDEAN DISTANCE BETWEEN INSTANCES

- Instance1 (definition of field):
  0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,1
  ,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
  ,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0

- Instance2 (field in table):
  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,1
  ,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
  ,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0

- Euclidean Distance = \( \sqrt{\sum_{i=1}^{n}(\text{attribute}\downarrow i \text{ instance}\downarrow 1 - \text{attribute}\downarrow i \text{ instance}\downarrow 2)} \)
Create dataset from processed documents for each data point

- Examples representation as bag of words:
  each unique word in the examples becomes an attribute in the dataset
- Dataset content
  - Context \{account, individu, init, investm, retir\}
  - Class ➔ data point category
    \{Minimum_{IRA Initial Purchase Amount}\}

Build model

- Apply Random Forest
  - algorithm for classification of instances
  - consists in constructing a multitude of Decision Trees
- New instance classification:
  - Run down on all trees ➔ classification accuracy from each tree
  - Classification accuracy result is computed as the average of the previously obtained accuracies
Model snippet of a decision tree

```
investm <= 0
  | system <= 0
  |   | subseq <= 0
  |   |   | c <= 0
  |   |   |   | roth <= 0
  |   |   |   |   | typ <= 0
  |   |   |   |   |   | shar <= 0
  |   |   |   |   |   |   | regl <= 0
  |   |   |   |   |   |   |   | s <= 0
  |   |   |   |   |   |   |   |   | fund <= 0
  |   |   |   |   |   |   |   |   |   | account <= 0
  |   |   |   |   |   |   |   |   |   |   | program <= 0: No (44.0/12.0)
  |   |   |   |   |   |   |   |   |   |   | program > 0: Yes (3.0)
  |   |   |   |   |   |   |   |   |   | account > 0: Yes (109.0/17.0)
  |   |   |   |   |   |   |   |   |   |   | fund > 0: Yes (9.0/2.0)
  |   |   |   |   |   |   |   |   |   |   |   | s > 0: Yes (6.0)
  |   |   |   |   |   |   |   |   |   |   | regl > 0: No (31.0/5.0)
  |   |   |   |   |   |   |   |   |   | shar > 0: Yes (18.0/1.0)
  |   |   |   |   |   |   |   |   |   | typ > 0: Yes (31.0/1.0)
  |   |   |   |   |   |   |   |   |   | roth > 0: Yes (14.0)
  |   |   |   |   |   |   | c > 0: Yes (25.0)
  |   | subseq > 0: No (14.0)
  | system > 0: Yes (38.0)
investm > 0
```
DATA POINTS FROM TABLES USING MACHINE LEARNING - STEPS

- **Dataset creation and Model building**
  - Tables identification
  - Dataset construction from tables
  - Create models based on dataset
    - Assign label Yes/No to each of the instances

- **Data classification**
  - Identify data point values based on the learned models
  - Performance evaluation
HOW THE APPROACH IS USED FOR A NEW PROSPECTUS

- Identify tables and classify them with table types
- Load models according to the types
- For each possible value identified in the table, extract context (including stemming)
- Create a new instance with the context and unknown class \{context, ?\}
- Run instance through all models $\rightarrow$ values for accuracy are obtained for each model
- The label of the instance will be the same as the model for which the greatest membership was obtained
**How the approach is used for a new prospectus - Example**

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
<th>Date</th>
<th>Value</th>
<th>Date</th>
<th>Value</th>
<th>Date</th>
<th>Value</th>
<th>Date</th>
<th>Value</th>
<th>Date</th>
<th>Value</th>
<th>Date</th>
<th>Value</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minimum Coverdell IRA Purchase Amount</td>
<td>250.00</td>
<td>02/28/2023</td>
<td>250.00</td>
<td>07/01/2023</td>
<td>250.00</td>
<td>02/28/2023</td>
<td>250.00</td>
<td>07/01/2023</td>
<td>250.00</td>
<td>02/28/2023</td>
<td>250.00</td>
<td>07/01/2023</td>
<td>250.00</td>
<td>02/28/2023</td>
</tr>
<tr>
<td>Minimum Coverdell IRA Subsequent Purchase A</td>
<td>100.00</td>
<td>02/28/2023</td>
<td>100.00</td>
<td>07/01/2023</td>
<td>100.00</td>
<td>02/28/2023</td>
<td>100.00</td>
<td>07/01/2023</td>
<td>100.00</td>
<td>02/28/2023</td>
<td>100.00</td>
<td>07/01/2023</td>
<td>100.00</td>
<td>02/28/2023</td>
</tr>
<tr>
<td>Minimum Initial Purchase Amount</td>
<td>100.00</td>
<td>02/28/2023</td>
<td>100.00</td>
<td>07/01/2023</td>
<td>100.00</td>
<td>02/28/2023</td>
<td>100.00</td>
<td>07/01/2023</td>
<td>100.00</td>
<td>02/28/2023</td>
<td>100.00</td>
<td>07/01/2023</td>
<td>100.00</td>
<td>02/28/2023</td>
</tr>
<tr>
<td>Minimum IRA/Roth IRA Initial Purchase Amount</td>
<td>250.00</td>
<td>02/28/2023</td>
<td>250.00</td>
<td>07/01/2023</td>
<td>250.00</td>
<td>02/28/2023</td>
<td>250.00</td>
<td>07/01/2023</td>
<td>250.00</td>
<td>02/28/2023</td>
<td>250.00</td>
<td>07/01/2023</td>
<td>250.00</td>
<td>02/28/2023</td>
</tr>
<tr>
<td>Minimum IRA/Roth IRA Subsequent Purchase A</td>
<td>100.00</td>
<td>02/28/2023</td>
<td>100.00</td>
<td>07/01/2023</td>
<td>100.00</td>
<td>02/28/2023</td>
<td>100.00</td>
<td>07/01/2023</td>
<td>100.00</td>
<td>02/28/2023</td>
<td>100.00</td>
<td>07/01/2023</td>
<td>100.00</td>
<td>02/28/2023</td>
</tr>
<tr>
<td>Minimum Initial Purchase Amount via an Exchange</td>
<td>1000.00</td>
<td>02/28/2023</td>
<td>1000.00</td>
<td>07/01/2023</td>
<td>1000.00</td>
<td>02/28/2023</td>
<td>1000.00</td>
<td>07/01/2023</td>
<td>1000.00</td>
<td>02/28/2023</td>
<td>1000.00</td>
<td>07/01/2023</td>
<td>1000.00</td>
<td>02/28/2023</td>
</tr>
</tbody>
</table>
**PERFORMANCE EVALUATION**

- Evaluate performance of the system by computing the classification accuracy on unseen data.
- Classification accuracy refers to the ability of the model to correctly predict the class label of new unseen data.
- Reported as %.
- Model accuracy refers to the degree of fitting the data and the model.
- We computed the accuracy for each of the models we built.
The classification accuracy varies between 41% and 85% due to:

- the numerous ways in which a data point can be expressed in the tables (the various ways of expressing each data point)
- the number of examples used for training the model
- the context extraction correctness
Demo – www.mfip.info
CONTACT

- **Business Information:**
  Drew Warren – Recognos Financial New York
dwarren@recognosfinancial.com

- **Technical Information**
  George Roth – Recognos Inc. – San Francisco
groth@recognos.com

The system can be seen at: [www.mfip.info](http://www.mfip.info)
The presentation will be published on Slide Share by George Roth