Extracting, Linking and Integrating Data from Public Sources: A Financial Case Study

Doug Burdick[1], Mauricio Hernández[1], Howard Ho[1], Georgia Koutrika[1], Rajasekar Krishnamurthy[1]
Lucian Popa[1], Ioana R. Stanoi[1], Shivakumar Vaithyanathan[1], Sanjiv Das[2]

[1] IBM Research – Almaden [2] Finance Department, Santa Clara University

Abstract

We present Midas, a system that uses complex data processing to extract and aggregate facts from a large collection of structured and unstructured documents into a set of unified, clean entities and relationships. Midas focuses on data for financial companies and is based on periodic filings with the U.S. Securities and Exchange Commission (SEC) and Federal Deposit Insurance Corporation (FDIC). We show that, by using data aggregated by Midas, we can provide valuable insights about financial institutions either at the whole system level or at the individual company level.

The key technology components that we implemented in Midas and that enable the various financial applications are: information extraction, entity resolution, mapping and fusion, all on top of a scalable infrastructure based on Hadoop. We describe our experience in building the Midas system and also outline the key research questions that remain to be addressed towards building a generic, high-level infrastructure for large-scale data integration from public sources.

1 Introduction

During the last few years, we have observed an explosion in the number and variety of public data sources that are available on the web: research papers and citations data (e.g., Cora, Citeseer, DBLP), online movie databases (e.g., IMDB), etc. Accurate extraction and integration of key concepts from these sources is challenging since their contents can be distributed over multiple web sites and vary from unstructured (or text) to semi-structured (html, XML, csv) and structured (e.g., tables). Even highly regulated sources, such as government regulatory data from the SEC and FDIC, still pose challenges since a large number of filings are in text.

In this paper, we present our experience with building and applying Midas, a system that unleashes the value of information archived by SEC and FDIC, by extracting, integrating, and aggregating data from semi-structured or text filings. We show that, by focusing on high-quality financial data sources and by combining three complementary technology components – information extraction, information integration, and scalable infrastructure – we can provide valuable insights about financial institutions either at the system level (i.e., systemic analysis) or at the individual company level. Midas is a system that starts from a document-centric archive, as provided by SEC and FDIC, and builds an entity-centric repository (similar to the "web of concepts" [12]) where the main entities are companies, their key people, loans, securities, together with their relationships.

There is a plethora of research in information extraction [11, 14], entity resolution [15], schema mapping [16, 19] and, in general, information integration [17]. While Midas relies on technologies and ideas from these

Bulletin of the IEEE Computer Society Technical Committee on Data Engineering

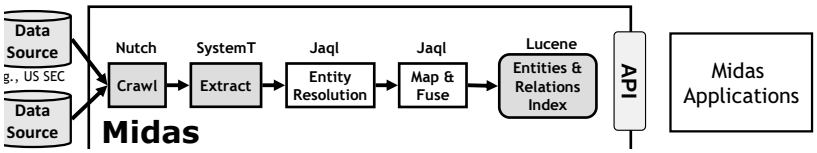

Figure 1: The Midas Data Flow

areas, our main contribution can be seen in the synergistic use of both unstructured and structured information integration to build a comprehensive solution for the financial domain that brings out the value of the data in public sources. This paper is organized as follows. Section 2 gives a high-level view of the Midas system. We describe in Section 3 two important types of applications that Midas enables. Section 4 gives further details on the integration stages implemented in Midas. We conclude in Section 5 with a discussion of remaining challenges.

2 Midas Overview

Figure 1 gives a high-level view of Midas, our system for extracting and integrating information from public data. Midas takes its input from multiple types of documents (ranging from text to HTML and XML) that are archived by SEC and FDIC. As output, Midas produces a set of integrated and cleansed entities and relationships, which are used by applications like the ones described in Section 3. The input data can be large (Peta-bytes of information, overall) with new incremental updates arriving daily. Hence, all components in the Midas data flow must be able to process large amounts of data efficiently and should scale well with increasing data sizes. To address these challenges, Midas is designed to run on Hadoop, and many of its operations are expressed in Jaql [5], a high-level language that compiles data transformations as map/reduce jobs.

Crawl is in charge of retrieving documents from the archives and storing them in our local file system. Instances of **Crawl** are implemented using Nutch, a widely used open-source crawler (http://nutch.apache.org/). We run Nutch as Hadoop jobs to parallelize the fetching of documents. We have currently crawled close to 1,000,000 SEC documents related to financial companies and 77,000 FDIC reports for active banks.

Extract is in charge of extracting facts from the unstructured documents. Here, we leverage a large library of previously existing information extraction modules (annotators) implemented on top of SystemT, a rule-based information extraction system developed at IBM Research and built around a declarative rule language (AQL) [9]. SystemT can deliver an order of magnitude higher annotation throughput compared to a state-of-the-art grammar-based IE system [9]. Furthermore, high-quality annotators can be built in SystemT with accuracy that matches or outperforms the best published results [10]. AQL rules are applied to each input document and produce a stream of annotated objects, making this operation trivially parallelizable with Hadoop. After applying extraction rules to each input document, we obtain structured records containing the extracted attributes with their values (e.g., a person name, a job title, a company name), as well as associated meta-data (e.g., the file id, date, text location of each extracted attribute, etc).

Entity Resolution identifies and links extracted records that correspond to the same real-world entity. The data required to build an entity (e.g., a person) is spread across many documents that mention various aspects of that entity, at various times. Recognizing that these separate mentions refer to the same entity requires a complex and domain-dependent analysis in which the exact match of values may not work. For instance, person names are not always spelled the same; moreover, the documents might not explicitly contain a key to identify the person. Entity Resolution, which appears in the literature under other names (Record Linkage, Record Matching, Merge/Purge, De-duplication) [15], is often solved with methods that score fuzzy matches between two or more candidate records and use statistical weights to determine when these records indeed represent the same entity. Other methods explicitly express when two or more candidate records match using *rules*. Our implementation of Midas uses the latter approach, where we implemented the matching rules as Jaql scripts.

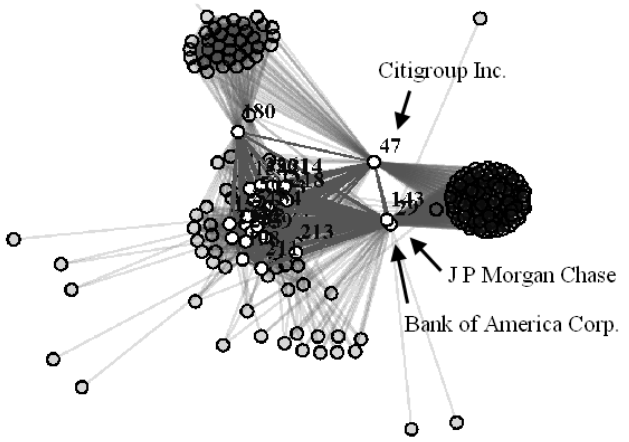

Figure 2: Co-lending network for 2005.

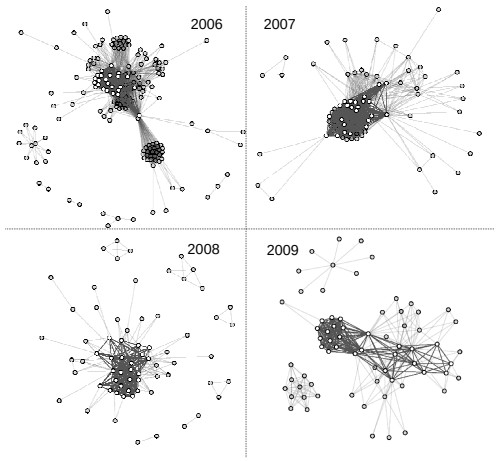

Figure 3: Co-lending networks for 2006–2009.

Map & Fuse transform the extracted (and possibly linked) records into aggregated entities. All necessary queries to join and map the source data into the expected target schema(s) are implemented as part of this operator. Since data is collected from multiple sources, duplicate values for certain fields are inevitable and must be solved to determine which value survives. A more complex type of fusion that takes place as part of Map & Fuse is the temporal aggregation for certain complex attributes of an entity that have a temporal aspect (e.g., the employment history of key executives over the years).

We note that in the Midas implementation, there is no fixed ordering between Entity Resolution and Map & Fuse. Instead, there are multiple instantiations of these components that apply in a flow. As we see in Section 4, Midas first creates an initial set of integrated entities via an application of Map & Fuse on a set of input records, which happen to have global keys for the main entities. Thus, there is no need for entity resolution in this first phase. However, subsequently, Entity Resolution needs to be applied so that other types of extracted records (from text, with no keys) are linked to the initial set of integrated entities. Based on the generated links, the extracted records are then integrated into the appropriate target entities by applications of Map & Fuse.

3 Midas: The Applications

In this section, we discuss the types of financial applications that the data aggregated by Midas enables. We group these applications into two types (one systemic, and one at the individual company level).

3.1 Systemic Risk Analysis

Systemic analysis is defined as the measurement and analysis of relationships across entities towards understanding their impact on the system as a whole. The failure of a major player in a market that causes the failure/weakness of other players is an example of a systemic effect, such as the one experienced with the bankruptcy of Lehman Brothers in 2008. A major challenge to performing quality systemic analysis is the paucity of data for the *entire* system. Current approaches rely on a few proprietary data sets of limited scope [1, 2, 6, 20]. Midas addresses this challenge in a major way by leveraging unstructured or semi-structured public data archived by SEC and FDIC to provide much richer information across the entire system of financial institutions. In turn, this enables finance researchers to develop new and powerful risk analysis techniques. As an example, we will use co-lending relationships to construct networks of relationships between banks, and then use network analysis to determine which banks pose the greatest risk to the financial system.

Co-lending Systemic Risk. Using loan-related data extracted from SEC/FDIC filings from 2005 to 2009, we construct a network of connections between financial firms based on their co-investment in loans made to other corporations or financial institutions. Concretely, if five banks made a joint loan, we add undirected edges (each

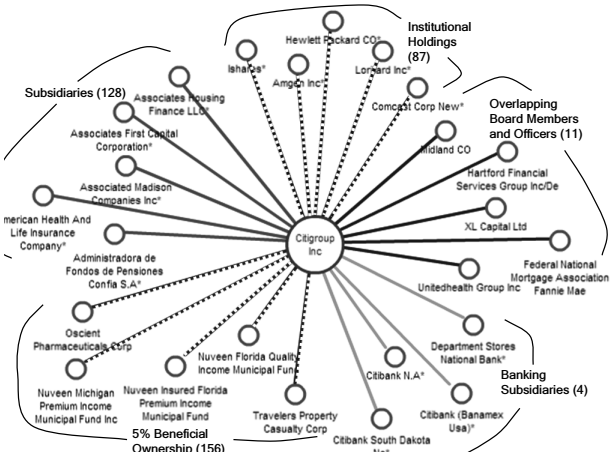

Figure 4: Companies related to Citigroup.

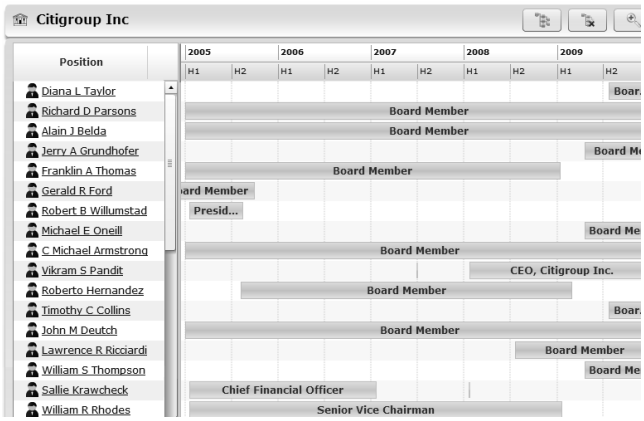

Figure 5: Key people for Citigroup.

with a score of 1, initially) for all pairs of the five banks. Overall, we create an undirected network with the banks as nodes, and where the edges have the total count of pairwise co-lending, aggregated across all loans. A bank failure will directly impact the co-lending activity of all banks it is connected with, and will indirectly impact the banks that are connected to the ones it is directly connected with. Therefore, even if a bank has very few co-lending relationships itself, it may impact the entire system if it is connected to a few major lenders.

Figure 2 shows the resulting co-lending network for 2005. We see that there are three large components of co-lenders that are clustered together, and three hub banks, with connections to the large components. To determine which banks in the network are most likely to contribute to systemic failure, we compute for each bank the normalized eigenvalue centrality score that is described in [7]. The three nodes with the highest centrality are seen to be critical hubs in the network—these are J.P. Morgan (node 143), Bank of America (node 29), and Citigroup (node 47). They are bridges between all banks, and contribute highly to systemic risk. Figure 3 shows how the network evolves in the four years after 2005. Comparing 2006 with 2005, we see that there are still large components connected by a few central nodes. From 2007 onwards, as the financial crisis begins to take hold, co-lending activity diminished markedly. The diameter of the co-lending graph becomes marginally smaller as the network shrinks. Also, all high centrality banks tend to cluster into a single large component in the latter years (see Figure 3, especially years 2007 and 2008).

This type of analysis is just one illustration of many possible techniques that can be developed on top of extracted and integrated data from SEC and FDIC. The particular framework we used, based on co-lending and centrality, can also be extended into a full risk management system for regulators.

3.2 Drill-Down into Individual Entities

While the previous section has given a view of the financial companies at the whole system level, in this section we describe additional views that are centered around the individual entities. For example, once a company such as Citigroup Inc. has been identified as a critical hub for the financial system, a regulator may want to dive deeper into various aspects of Citigroup: its relationships with other companies (subsidiaries, competitors, borrowers, etc.), its key executives (officers and directors), its aggregated financial data (loans, investments, etc.).

Company Relationships. Figure 4 shows how Citigroup is related to other companies through investment, lending, ownership, as well as shared insiders. For each relationship type, along with a count of the number of related companies in that category, we show up to five representative companies. **Banking subsidiaries** lists the four banks that Citigroup has registered with the FDIC. This information was obtained by integrating data from SEC and FDIC. **Subsidiaries** is an exhaustive list of Citigroup's global subsidiaries, as reported in the latest annual report (typically in text or html format). **5% Beneficial Ownership** enumerates the securities in which Citigroup has more than 5% ownership based on analysis of SC-13D and SC-13G text filings made

by Citigroup and its subsidiaries. **Overlapping board members/officers** represents key insiders (directors or officers) that are shared between Citigroup and other companies. **Institutional Holdings** represents securities in which Citigroup has invested more than $10 million based on analysis of 13F text filings. While the relationship graph provides a birds-eye view on Citigroup, one can further drill down into any of the individual relationships. We explore next the key insider aspect of a company.

Insider Analysis and Employment Histories. Understanding management structure of companies and relationships across companies through common officers and board of directors is relevant in firm dynamics and corporate governance. Connected firms appear to end up merging more [8], and understanding post-merger management structures based on earlier connections between the managers of the merged firms is also being studied [18]. To enable such analysis, Midas exposes detailed employment history and trading information for insiders (i.e., key officers and directors) of individual companies. Figure 5 shows some of the key officers and directors associated with Citigroup over the last several years. For each key person, the various positions held in Citigroup along with the time periods are also displayed. This profile is built by aggregating data from individual employment records present in annual reports, proxy statements, current reports and insider reports.

Additional views for insider holdings, insider transactions, and lending exposure are described in [4].

4 Midas Integration Flow: Further Details

We now give concrete details of the Midas flow that integrates information related to financial companies. We discuss in Section 4.1 the initial construction of a reference or core set of company and people entities from insider reports (Forms 3/4/5). Since these forms are in XML and contain structured and relatively clean data, the resulting core set of entities forms the backbone of the rest of the integration flow. In Section 4.2, we detail how further information from unstructured forms is extracted, linked and fused to the core set of entities. The final result is a set of entities with rich relationships, including detailed employment histories of key people, lending/co-lending relationships among companies, and all the other relationships we discussed in Section 3.2.

4.1 Constructing Core Entities

We now discuss the initial construction and aggregation of company and key people entities from the XML files that correspond to insider reports (Forms 3/4/5).

Extraction of records from XML forms. We use Jaql to extract (and convert to JSON) the relevant facts from XML Forms 3/4/5. Each fact states the relationship, as of a given reporting date, between a company and a key officer or director. The relevant attributes for the company are: the SEC-assigned key (or cik) of the company, the company name and address, the company stock symbol. The relevant attributes for the person are: the SEC key or cik, name, an attribute identifying whether the person is an officer or a director, and the title of the person (i.e., "CEO", "Executive VP", "CFO", etc) if an officer. Other important attributes include the reporting date, a document id, a list of transactions (e.g., stock buys or sells, exercise of options) that the person has executed in the reporting period, and a list of current holdings that the person has with the company.

Aggregation of company and people entities. In this step, we process all the facts that were extracted from XML forms and group them by company cik. Each group forms the skeleton for a company entity. The important attributes and relationships for a company are aggregated from the group of records with the given company cik. As an example of important attribute of a company, we aggregate the set of all officers of a company such as Citigroup Inc. This aggregation is with respect to all the forms 3/4/5 that Citigroup Inc. has filed over the five years. Additional fusion must be done so that each officer appears only once in the list. Furthermore, for each officer, we aggregate all the positions that the respective person has held with the company. As an example, a person such as Sallie Krawcheck will result in one occurrence within the list of officers of Citigroup, where this occurrence contains the list of all the positions held by Sallie Krawcheck with Citigroup (e.g., CFO, CEO of Global Wealth Management). Since positions are strings that vary across forms, normalization code is used to identify and fuse the "same" position. Finally, each position is associated with a set of dates, corresponding to

all the filings that report that position. The earliest and the latest date in this set of dates is used to define the time span of the position (assuming continuous employment). The result of this analysis is illustrated in Figure 5.

To give a quantitative feel about the above processing, there are about $400,000$ facts extracted from the 3/4/5 forms. These $400,000$ facts result in about $2,500$ company entities, each with a rich structure containing officers with their position timelines (within the company), directors (with similar timelines), and also containing an aggregation of transactions and holdings (to be discussed shortly). A separate but similar processing generates, from the same $400,000$ facts, an inverted view where people are the top-level entities. We generate about $32,000$ people entities, corresponding to the officers or directors that have worked for the $2,500$ financial companies. Each person entity is also a complex object with nested attributes such as employment history, which spans, in general, multiple companies. For example, the person entity for Sallie Krawcheck has an employment history spanning both Citigroup Inc. (where she served as CFO and then CEO of Global Wealth Management) and Bank of America (which she joined later as President of Global Wealth and Investment Banking).

Fusion of insider transactions and holdings. The aggregation of transaction and holding data over the collection of forms 3/4/5 requires a detailed temporal and numerical analysis. First, we need to ensure that we group together securities of the same type. In general, there are multiple types of securities (derivatives or non derivatives), types of ownership (direct or indirect), and types of transactions (acquired, disposed, granted, open market purchase, etc.). The various values for such types are reported in text and have variations (e.g., "Common Stock" vs. "Class A stock" vs. "Common shares"). In order to avoid double counting of transactions and to report only the most recent holding amount for each type, we developed normalization code for types of securities and for types of ownership. Subsequent processing summarizes, for each company and for each year, the total amount of transactions of certain type (e.g., open market purchase) that company insiders executed in that year. Examples of views that can be constructed from results of such aggregation can be found in [4].

4.2 Incorporating Data from Unstructured Forms

We now discuss the processing involved in the extraction and fusion of new facts from unstructured data into the core entities. The new facts, which are extracted from either text or HTML tables, describe new attributes or relationships, and typically mention a company or a person by name without, necessarily, a key. Thus, before the new information can be fused into the existing data, entity resolution is needed to perform the linkage from the entity mentions to the actual entities in the core set. Consider the example of enriching a person's employment history. In addition to the insider reports, information about a person's association with a company occurs in a wide variety of less structured filings. This information ranges from point-in-time facts (when an officer/director signs a document) to complete biographies of a person. To extract and correctly fuse all the needed pieces of information, we must address several challenges.

Extract. Employment history records need to be extracted from various contexts such as biographies, signatures, job change announcements, and committee membership and compensation data. These records are typically of the form (person name, position, company name, start date, end date) for each position mentioned in the text. However, not all of the attribute values may be present or extracted successfully. For instance, extraction may result in records such as (James Dimon, Chairman, JP Morgan Chase, –, –), (James Dimon, Chief Executive Officer, JP Morgan Chase, –, –), (James Dimon, Director, JP Morgan Chase, 2000, –) and (Mr. Dimon, Chairman, unknown, "December 31, 2006", –). All of these records have to be linked and fused by the next stages.

Using biographies as an example, we give a flavor of the challenges one typically encounters in extracting employment records from unstructured documents in SEC. First, biographies typically appear as short paragraphs within very large HTML documents (100s KBs to 10s MBs) and within HTML tables, where individual employment facts may be formatted in different ways. For instance, a position with a long title may span multiple rows while the corresponding person's name may align with only one of these rows, depending on the visual layout. Moreover, a person may have multiple positions linked with a single organization (e.g., Chairman and CEO of JP Morgan Chase); these positions sometimes are associated with the same start date (e.g., CEO and

President since 12/31/2005) or may have different timelines (some of which may have to be inferred later in subsequent stages). Furthermore, individual sentences may refer to an individual via a partial name (e.g., "Mr. Dimon") or by using pronouns (e.g., "he"). Sometime the name of a related individual may be mentioned in the biography. Hence, extracting the person name itself may be a challenge. All of these challenges are addressed in Midas by carefully designed and customized AQL rules that are often complemented by Java functions.

Entity Resolution. As mentioned, the attributes extracted for biographies include the name of the person, the name of the filer company (also the cik, since this is associated with the filing entity) and the biography text itself. However, information in biographies does not contain the key (cik) for the person and we need entity resolution to link each extracted biography record to a person cik. Entity resolution is an iterative process that requires understanding the data, writing and tuning the matching rules, and evaluating the resulting precision (are all matches correct?) and recall (did we miss any matches and why?). We summarize next the main issues in matching people mentioned in biographies to the actual person entities.

No standardization in entity names. People names come in different formats (e.g. "John A. Thain" vs. "Thain John" vs. "Mr. Thain", or "Murphy David J" vs. "Murphy David James III"). Hence, exact name matching will only find some matches and we need approximate name matching functions to resolve more biographies. On the other hand, two people with similar names (even when working for the same company) may be in fact two different people. For example, "Murphy David J" and "Murphy David James III" are two different people. To tackle this challenge, we designed specialized person name normalization and matching functions that cater for variations in names, suffixes such as "Jr.", "II", and allow matching names at varying precision levels. We iterated through our data and entity resolution results several times to fine-tune our functions.

Achieving high precision. To improve precision beyond just the use of name matching, we observed that for a biography record, we typically know the cik of the company (since it is the filing entity). As a result, we were able to develop matching rules that exploit such contextual information. In particular, the rules narrow the scope of matching to only consider the people entities that are already known to be officers or directors of the filing company (as computed from Forms 3/4/5). The final resulting precision was close to 100%.

Improving recall. To improve recall, in general, one needs multiple matching rules. For example, there are cases where the filer company is not in the employment history of a person (based on Forms 3/4/5). Hence, we must include other, more relaxed rules that are based just on person name matching. Having multiple rules, we prioritized them so that weaker matches are kept only when there are no matches based on stronger evidence. For instance, if we matched a "Thain John A" mentioned in a biography to both a "John A. Thain" and a "Thain John" in key people, via two different rules, we will only keep the first match since it is based on a rule that matches first/last name *and* middle name initial. Our initial rules achieved a 82.29% recall, that is, 82.29% of $23,195$ biographies were matched to a person cik. At the end of the tuning process, we raised that to 97.38%.

After Entity Resolution, a subsequent instantiation of the Map & Fuse stage takes place to perform the actual mapping and fusion of the linked records. The process is similar to what was described in the earlier Section 4.1, except that fusion is now with respect to an already existing set of entities. We omit any further details.

5 Future Directions

Even though aimed at a specific (financial) domain, building the target integrated dataset for Midas was a highly non-trivial task, which required about a year of intensive research and development that used multiple languages (e.g., Java, AQL, and Jaql) and packages on top of the Hadoop platform. We also note that traditional ETL or SQL/XML processing cannot be directly applied since the data is highly heterogeneous, with large variation in the number of attributes and data values; moreover the scale of the data can be daunting.

The longer-term vision here is to take a Midas-type of system[1] to the next level where one can easily develop and customize a sophisticated data integration flow in any given domain. Ideally, a small team of data analysts should be able to identify the important concepts or entity types that they are interested in, and then specify at

[1]Other examples, such as DBLife [13], exist here.