Association for Information Systems
AIS Electronic Library (AISeL)

AMCIS 2005 Proceedings

Americas Conference on Information Systems (AMCIS)

2005

Building Discerning Knowledge Bases from Multiple Source Documents, with Novel Fact Filtering

Jason Hale
University of Mississippi, jghale@olemiss.edu

Sumali Conlon
University of Mississippi, sconlon@bus.olemiss.edu

Tim McCready
University of Mississippi, tmccread@olemiss.edu

Anil Vinjamur
University of Mississippi, vinjamur@olemiss.edu

Building Discerning Knowledge Bases from Multiple Source Documents, with Novel Fact Filtering

Jason Hale
University of Mississippi
jghale@olemiss.edu

Sumali Conlon
University of Mississippi
sconlon@bus.olemiss.edu

Tim McCready
University of Mississippi
tmccread@olemiss.edu

Susan Lukose
University of Mississippi
svlukose@olemiss.edu

Anil Vinjamur
University of Mississippi
vinjamur@olemiss.edu

ABSTRACT

Information extraction systems that remember only novel information (facts that differ semantically from those previously extracted) can be used to build lean knowledge bases fed from multiple, possibly overlapping sources. In previous research by the authors, natural language processing techniques were used to build a system to extract financial facts from international corporate reports of the *Wall Street Journal*. We will enhance that system to extract the same types of financial facts from a second source of corporate financial reports: *Reuters*. The improved system will provide more generality through its ability to extract from multiple sources rather than just one. In addition, it will provide novelty filtering of extracted information, admitting only novel facts into the database, while remembering all sources that a redundant fact came from.

Keywords: Information Extraction, Novelty and Redundancy Filtering, Text Mining, Natural Language Processing

INTRODUCTION

The explosion of the WWW has created opportunities for using and sharing enormous amounts of information. Unfortunately, most of this information is in textual form (Chen 2001). Converting existing stockpiles of web text to manageable format is laborious, so computer systems are needed that can process information expressed in natural languages. Web-based knowledge systems can help meet the demands of decision makers for fast, reliable, actionable business information. However, knowledge bases must be fed information that is more structured and manageable than free text.

In this research, we discuss our on-going research in the area of information extraction from online documents. Our past research was on extracting financial information from a single source (the *Wall Street Journal*). To build discerning knowledge bases, it would be useful to extract information from a variety of online sources. While online sources often complement one another, many will contain redundant or conflicting information. Thus, our current research is on extracting information from more than one source. The system compares and contrasts extracted information from these sources, synthesizes complementing information across sentence and document boundaries, filters out conflicts and duplicates, stores unique facts in the database, and remembers each source that corroborates each fact.

In this paper, we describe the motivation and related work for this research, discuss the data we use for building this system, explain the system architecture and techniques for extracting, comparing, and extraction information from multiple sources, and present preliminary results of the system.

Motivation and Related Work

Research in data mining has been powerful for analyzing static data such as items in data warehouses or relational databases. These technologies mostly focus on text clustering and text categorization, and recent techniques have included neural networks and genetic algorithms. Text clustering and categorization help business to classify data and texts but do not try to understand their contents. Like data mining, the goal of text mining is discovering knowledge, but by making inferences from information extracted from multiple text documents, rather than from databases. (Hearst, 2003.)

Research in artificial intelligence and natural language processing makes it possible to process text data. Current text related applications include spell and grammar checking, machine translation, natural language interface, information extraction, and text generation. In many business applications, information extraction (IE) allows us to interpret and convert implicit information from text to an explicit format. These techniques will be very helpful in dealing with the huge amounts of textual data.

Information extraction is the process of extracting specific data from text and presenting the data in a summarized, semantically structured format. IE research has been going on since the 1960s (Cardie, 1997). In the late 1960s, Naomi Sager at New York University developed an IE system that extracted hospital discharge information from patient records. Jacob and Rau developed SCISOR, an information extraction system for finding and analyzing corporate merger stories. The extracted data were stored in a template (Jacob and Rau, 1990). More recently, Gerdes (2003) built the EDGAR analyzer to analyze text from SEC filing data.

In IE, source texts can be structured or free-form. Some free-text extractors attempt to synthesize facts spanning multiple sentences in a document. IE systems tend to be very domain-specific, and require humans to build the extraction rules targeted to the domain and expected syntactic style of source text. Broadening their scope requires manually expanding the extraction rules, limiting general usability. Soderland (1999) and others have employed machine learning algorithms to create systems like WHISK that can automatically learn extraction rules in a variety of domains. Craven et al. (2000) employed machine learning algorithms to build WEBKB, a prototype information extraction system that constructed a computer-understandable knowledge base mirroring relevant information from domain-specific web pages it visited. These researchers coined the phrase "the multiple Elvis problem" for teaching an information extraction engine to recognize when independently discovered instances of a fact are equivalent (i.e., when two facts extracted from two different web sources are one and the same fact reported by two different sources, not two separate, novel facts.) Zhang, Callan, and Minka (2002) addressed the issue of novelty at the source-document level, as opposed to the extracted-fact level.

For extracted information to be useful to businesses, it must be well managed in a knowledge base that presents an abstraction of semantic facts that succinctly represent the best information discernable across all available sources. Three important aspects of the abstraction are condensing complementing facts, filtering out redundant facts, and identifying and resolving conflicting facts extracted from different sources. In this research, we continue our ongoing development of the FIRST information extraction system by adding fact-level redundancy detection, novelty filtering, and a knowledge base.

DATA AND METHODOLOGY

The FIRST system (2004) was originally developed by the authors to extract information from short, narrowly focused, free-text corporate financial reports from a single-source, the *Wall Street Journal*. Written in Perl, it relies on a standard Perl module for part-of-speech tagging, WordNet (Miller et al. 1990, Miller 1995) synonym expansion to enhance recall, and a KWIC (Luhn, 1960) index to enhance precision. The human-coded extraction rule base uses regular expressions to extract information from recognized patterns of text.

The enhanced FIRST QUARTER system offers the following improvements over the original: 1) it extracts facts from an additional data source, *Reuters*; 2) it adds a novelty filtering layer; 3) it makes inferences across sentence and document boundaries; 4) it stores unique facts in a RDBMS-based knowledge base, and 5) it makes each knowledge-based fact reference all of its substantiating sources.

We need to find overlapping news stories from our two sources in order to train and evaluate the extractor. We download from the *Reuters* web page (http://today.reuters.com/news/newsChannel.aspx?type=businessNews) business articles that contain corporate earnings reports and forecasts. For each *Reuters* article, we search www.wsj.com for the company name, and download each article for that company that posted within the same 24 hours. Each article is mined by hand and a list of

discrete facts identified from each. The articles are then fed into the FIRST QUARTER system, where they are synthesized into unique facts and put in the database. We then compare the manually extracted facts to the database to evaluate the precision, recall, and novelty detection of the system.

SYSTEM ARCHITECTURE

FIRST

In FIRST, the patterns were found using the data from KWIC index file and the output from the CMU toolkit. The patterns found helped us to build a rule base. FIRST used pattern recognition, natural language processing, and a WordNet lexical database to extract facts and predictions. About seven types of financial elements were extracted from corporate reports of the *Wall Street Journal*. Specifically, FIRST extracted reports and predictions of corporate earnings, income, profit, sales, revenue, exports, and production into tabular facts and put them in a flat file. An example of an extracted fact is shown in Figure 1. FIRST did not attempt to make inferences across sentence boundaries, and used a crude mechanism to detect company names based on capitalization.

Organization Name:	SANYO ELECTRIC CO
Organization Description:	One of Japan's biggest makers of electrical and electronic products
Fact / Prediction	Fact (Has Happened)
Financial Item:	Earnings
Financial Item Status:	Fell
Financial Item % Change:	94%
Financial Item Change Description:	From $235.4 million to $13.7 million
Sales Status:	Fell
Sales % Change:	21%
Sales Change Description:	from $9.76 billion to $7.68 billion

Figure 1. Sample Output, FIRST System

FIRST QUARTER Design

FIRST QUARTER includes an Oracle database to store the novel facts and link them to their source articles. (See figure 3.) It is the job of the information extraction layer to scrub facts clean of their original language syntax. In FIRST QUARTER, "profit" and "earnings" are scrubbed to "income", and organization names are unambiguously represented at the knowledge layer, with the help of the ORGANIZATIONS reference table. Once the facts have been scrubbed, two facts "match" if they have the same organization (company name), financial element (income, sales, etc.), type (fact or prediction), and time interval (quarter). Matched facts are "related" as duplicates, complements, obviates, or conflicts, as described below. The novelty filter then ingests these facts into the knowledge base based on their relationships. Novelty filtering logic is depicted in Figure 4. The fact flow is depicted in Figure 2.

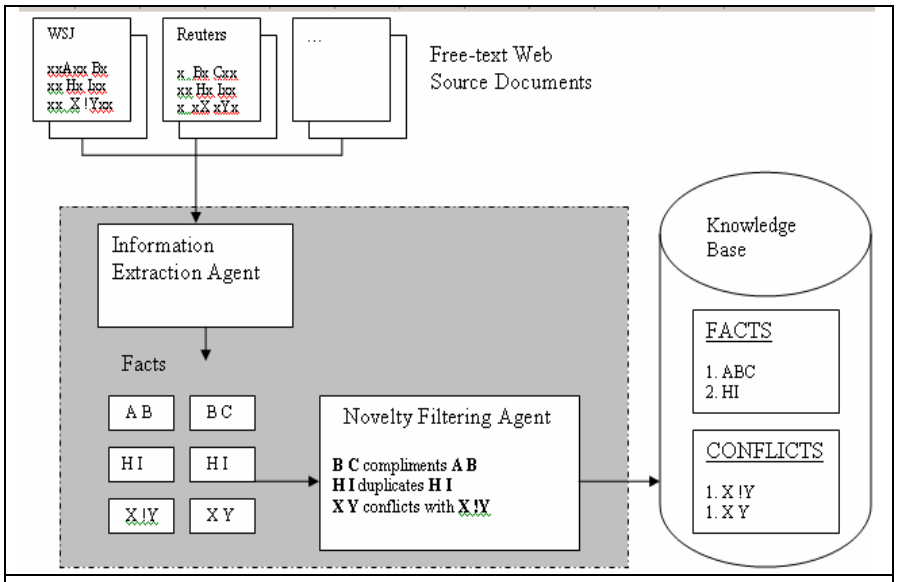


Figure 2. FIRST QUARTER Fact Flow

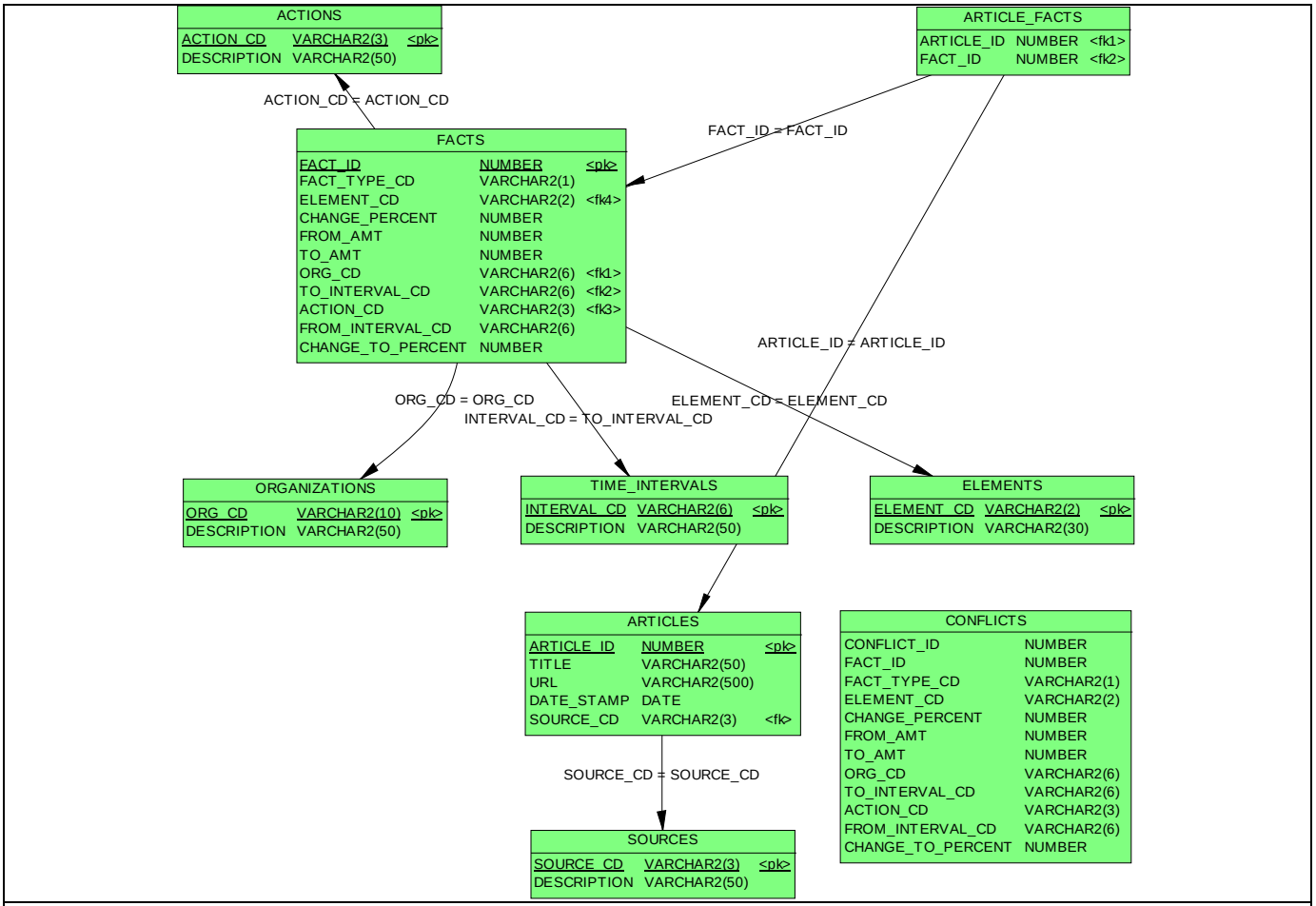
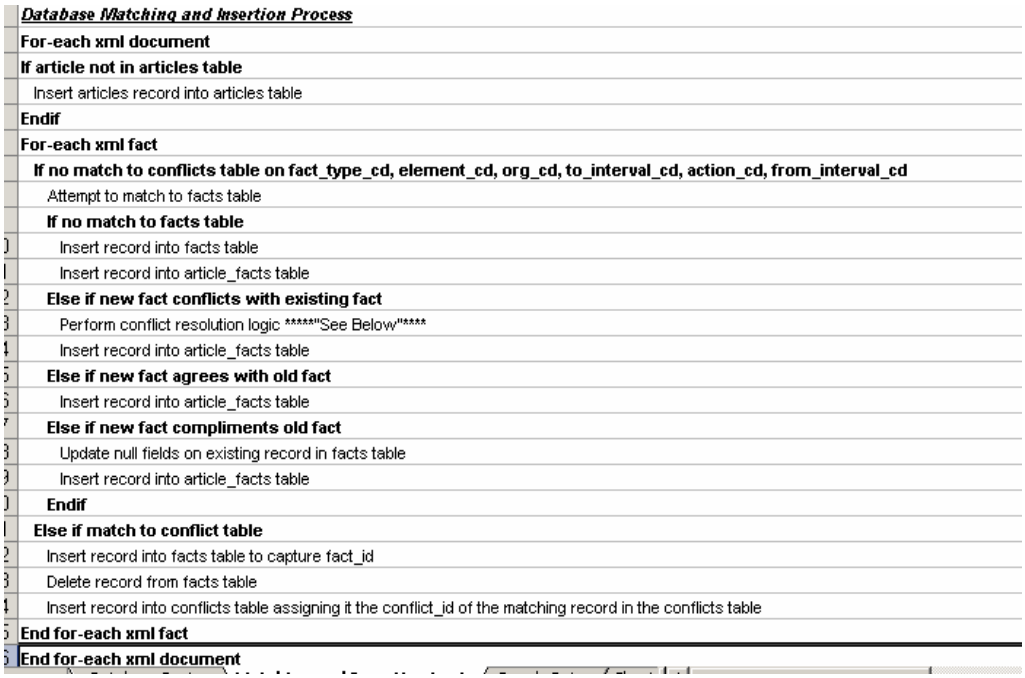


Figure 3. FIRST QUARTER Database Design

Fact Matching and Relationships

Duplicate Facts

An important feature of the knowledge base is that each unique fact should only be stored once. (If there is only one Elvis, it would be redundant to store 518 of them, even if there were 518 Elvis sightings.) However, multiple sources of each fact should be remembered because: 1) a fact corroborated by multiple sources is, intuitively, more reliable than a single-sourced fact, and 2) the consumer may want to refer to the source article(s) behind a fact. FIRST QUARTER accomplishes multiple sourcing with a separate table that traces each fact back to each of its source articles.



Database Matching and Insertion Process

For-each xml document
If article not in articles table
Insert articles record into articles table
Endif
For-each xml fact
If no match to conflicts table on fact_type_cd, element_cd, org_cd, to_interval_cd, action_cd, from_interval_cd
Attempt to match to facts table
If no match to facts table
Insert record into facts table
Insert record into article_facts table
Else if new fact conflicts with existing fact
Perform conflict resolution logic *****"See Below"*****
Insert record into article_facts table
Else if new fact agrees with old fact
Insert record into article_facts table
Else if new fact compliments old fact
Update null fields on existing record in facts table
Insert record into article_facts table
Endif
Else if match to conflict table
Insert record into facts table to capture fact_id
Delete record from facts table
Insert record into conflicts table assigning it the conflict_id of the matching record in the conflicts table
End for-each xml fact
End for-each xml document

Figure 4. Fact Novelty Matching Logic

Complementary Facts

In some cases, one source document will include part of a fact, but not the entire fact. For instance, from a *Reuters* article dated 02/18/05, FIRST QUARTER extracts the fact that J.M. Smucker Co.'s income increased to $36.1 million in the 3rd Quarter of 2004. From a *Wall Street Journal* article published the same day, FIRST extracts the fact that J.M. Smucker Co.'s income increased 15% from $31.3 million in the 3rd Quarter of 2003 to $31.6 million in the 4th Quarter of 2004. These are not identical facts; however, they are not conflicting either. So, the FIRST QUARTER system recognizes them as complementary, and stores (updates) the single fact with the best information from both articles (inferring across document boundaries.)

Facts of Differing Precision

From a *Wall Street Journal* article dated 05/18/05, FIRST QUARTER extracts the fact that JC Penney Co.'s 1st Quarter 2006 sales rose 4% from $4.03 billion to $4.19 billion; however, from a *Reuters* article published the same day, it extracts the fact that JC Penney Co.'s 1st Quarter 2006 sales increased 3.9% to $4.19 billion. Strictly speaking, the facts differ by 0.1% on the increase. However, if we assume that the increase was rounded to the nearest percent in the *Wall Street Journal* article, there is no conflict. FIRST QUARTER is designed to remember the most precisely reported attribute from among the various sources contributing to a fact.

Obviated Facts

Sometimes, web articles will be published with factual mistakes. Often, the publishing news source will publish a correction to the mistake in a new article. For instance, from a 5:35 PM 02/10/05 article of the *Wall Street Journal*, a fact is abstracted that Watson Pharmaceuticals Inc.'s profits for 4th Quarter 2004 increased 2% from 4th Quarter 2003. However, from a subsequent article from the *Wall Street Journal*, posted 30 minutes later, a fact is extracted that Watson Pharmaceuticals Inc.'s profits for 4th Quarter 2004 increased 4% from the 4th Quarter 2004. The 2nd article contains textual clues that it is correcting the article that previously posted. By enhancing the rules of FIRST QUARTER, we recognize these corrections, and do a fact update, rather than an insert.

Conflicting Facts

Some facts from different sources conflict with one another, and a knowledge base should account for these conflicts. FIRST QUARTER does so with a CONFLICTS table of facts, where groups of conflicting facts are stored until enough evidence can be gathered to judge between them. At that time, the substantiated facts are moved from the CONFLICTS table to the FACTS table, and the discredited facts are discarded. The conflict resolution process is depicted in Figure 5.

Conflict Resolution Logic****

If new fact match existing fact on fact_type_cd, element_cd, org_cd, to_interval_cd, action_cd, from_interval_c substantially on [change_percent or from_amt or to_amt or change_to_percent]

Insert new fact into facts table to capture fact_id

Delete new fact and existing fact from facts table and insert both records into conflicts table. "Both records would be assig conflict_id

Endif

Group conflicts table on conflict_id, change_percent, from_amt, to_amt, change_to_percent recording count of instances

If one conflict_id has higher group count than all other records in the conflict table with same conflict_id

Insert chosen record into facts table

Delete chosen record from conflicts table

Else if all instances of the same conflict_id occur an equal numer of times

Do not insert any record with the particular conflict_id into the facts table

All records with the particular conflict_id remain in the conflicts table

Endif

Figure 5. Conflict Resolution Management

RESULTS

To date, we have updated the rule base and successfully extracted time period information (e.g., 1Q2004, Full Year 2005, etc.), and abstracted financial information from overlapping corporate reports of both *Wall Street Journal* and *Reuters*. We have built the database in Oracle, and are building the novelty filter in Java that reads in XML fact extracts and updates the knowledge base. We are in the process of manually extracting and tagging a somewhat larger set of test documents from the Wall Street Journal and Reuters web sites. We will feed the raw source documents into FIRST QUARTER, and use the manually extracted information to validate and evaluate the performance of the FIRST QUARTER system, in recall, predictions, and novelty.

CONCLUSION

Business decision makers need fast, up-to-date information. The web is full of information buried in text documents. Automated tools are needed to search those documents, and extract and discern the relevant, novel facts, into knowledge bases that can inform business decision making. All sources corroborating each fact should be available on-demand. The ongoing FIRST research project features a domain-specific information extraction tool that relies on natural language

processing and a human-trained extraction rule base. The original tool was developed in Perl, and used WordNet and a KWIC Index to generate keywords. The current version, FIRST QUARTER, should be complete by summer 2005, adding novelty filtering, conflict resolution, an Oracle knowledge base, and the ability to generalize knowledge across multiple source documents.

REFERENCES

1. Cardie, C. (1997) Empirical methods in information extraction, *AI Magazine*, 18(4):65-80.
2. Chen (2001) *Knowledge Management Systems: A Text Mining Perspective*. The University of Arizona, Tucson, AZ.
3. Craven, M., DiPasquo, D., Freitag, D., McCallum, A., Mitchell, T., Nigam, K., Slattery, S. (2000) Learning to Construct Knowledge Bases from the World Wide Web. *Artificial Intelligence*. 118 (1-2): 66-113.
4. Hearst, Marti (2003) What Is Text Mining? (http://www.sims.berkeley.edu/~hearst/text-mining.html)
5. Jacobs, P. S. and Rau, L. (1990) SCISOR: Extracting information from On-Line News. *Communications of the ACM*, 33(11):88-97.
6. Luhn, H.P. (1960) Keyword-in-context index for technical literature (KWICindex), *American Documentation* 11:288-295.
7. Miller, G. A., Beckwith, R., Fellbaum, C., Gross, D. and Miller, K. J. (1990) Introduction to WordNet: an on-line lexical database, *International Journal of Lexicography*, Vol. 3, No. 4: 235 - 244.
8. Miller, G. A. (1995) WordNet: a Lexical Database for English, *Communication of the ACM*, Vol .38, No 11: 39-41.
9. Soderland, S. (1999) Learning Information Extraction Rules for Semi-structured and Free Text, *Machine Learning* 34(1-3), pp 233-272.
10. Zhang, Y., Callan, J., and Minka, T. (2002) Novelty and Redundancy Detection in Adaptive Filtering, *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval* pp 81-88.