2005

Automatic Extraction and Generation of XML Documents from Financial Reports

Anil Vinjamur
University of Mississippi, vinjamur@olemiss.edu

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

Susan Lukose
University of Mississippi, svlukose@olemiss.edu

Tim McCready
University of Mississippi, tmccread@olemiss.edu

Jason Hale
University of Mississippi, jghale@olemiss.edu

Automatic Extraction and Generation of XML Documents from Financial Reports

Anil Vinjamur
University of Mississippi
vinjamur@olemiss.edu

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

Susan Lukose
University of Mississippi
svlukose@olemiss.edu

Tim McCready
University of Mississippi
tmccread@olemiss.edu

Jason Hale
University of Mississippi
jghale@olemiss.edu

ABSTRACT

Web services require XML formatted data. Human translation of business information from the rapidly expanding volume of documents to XML is labor-intensive and impractical. Computer programs can be built to extract domain-specific facts from web documents and convert them into an XML format. With a continual feed of web articles, such a system could be used to maintain an up-to-date XML knowledge base that could power web services for businesses. In this research, we build a system to automatically extract information from electronic international corporate financial reports, and translate this information into XML or XBRL (a well-known XML extension for accounting and financial data).

Keywords

Web services, Natural Language Processing, Information Extraction, XML, and XBRL.

INTRODUCTION

Web services have played a major role in business. As business environments change very quickly, information systems need up-to-date information. In order for web service systems to communicate with each other, the data must be tagged using some standard format such as XML or XBRL. If the data are well structured, such as data items in a database, the conversion process will be easy. However, at each particular period of time, there is a lot of new information produced and much of it is in the form of written documents, including news articles published on the web. For information systems to use information from these sources, people have to extract information from them, then put the information into an explicit format, such as a database or tagged XML document.

This research is built on our on-going research in the area of information extraction. Our system (FIRST), extracts corporate financial reports from the Wall Street Journal with the help of natural language processing (NLP) techniques. FIRST automatically tags the extracted financial information into a customized XML format and possibly to standard XBRL format.

We discuss web services and their needs in the second section. The third section discusses the related work about XML tagging in the literature. The fourth section describes financial data extraction and XML formatting while the fifth section shows the current results our system produces. Section six discusses the potential use of our system in business. Finally, the seventh section concludes the paper.

WEB SERVICES AND THEIR NEEDS

Business transactions on the web are growing rapidly and there is a need for information systems that support up-to-date information. Information on the web, which is comprised of news articles and business reports, is used both by investors and

business decision makers to help them make more informed decisions. To get information from news stories, people have to read and interpret these stories. If they want to use computers to further analyze this information, they must input this information in a database or some financial data analyzer. In web-service systems, the data must be converted into an XML format so that it can be used by business systems (B2B, B2C, C2C, etc.). W3C defines web services as follows:

> [Definition: A Web service is a software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine-processable format (specifically WSDL). Other systems interact with the Web service in a manner prescribed by its description using SOAP messages, typically conveyed using HTTP with an XML serialization in conjunction with other Web-related standards.
>
> http://www.w3.org/TR/ws-arch/#whatis

From this definition, we can see that web service systems interact with each other through machine-to-machine interaction over the network. Such systems must have the most current data to process their tasks efficiently, and must therefore be able to update their own information. They should be able to get new information from the web (such as news stories) and convert it into an XML format. These processes require advanced technologies to select the appropriate web pages and extract the right sets of information from these pages. Finally, these systems should be able to tag the data for use by web service systems. Research in the areas of information retrieval and natural language processing has been advanced in part to handle such tasks. We therefore apply these techniques to build a system that can analyze online business documents and convert these documents into XML formatted files for web service systems to use.

The other technologies that support web services include SOAP (Simple Object Access Protocol), UDDI (Universal Description, Discovery and Integration), and WSDL (Web Services Description Language) (http://www.acm.org/ubiquity/views/f_coyle_1.html). SOAP has helped fashion the client-server model over the Web. SOAP is simply a set of XML tags for moving XML data around the Web using standard Web protocols. The data is passed from a server over the web in the form of SOAP messages, which can be received and further processed or broadcasted by a layer of software services known as Message Oriented Middleware (MOM). SOAP and other kind of web services are helping in making diverse services interoperate to create complete business processes. XML, SOAP and Web Services define a new landscape for distributed computing that includes XML as the data, SOAP and HTTP as the protocols for moving data across the Web, and Web Service protocols such as UDDI and WSDL for the discovery and connection to those services.

With these emerging data transfer technologies and the increasing amount of financial data available on the web, business will benefit tremendously if there are systems that can extract information from various online sources and produce XML formatted files automatically.

RELATED WORK – XML and XBRL

Extensible Markup Language (XML) was introduced by W3C in 1996 as a simple dialect of SGML (Standard Generalized Markup Language). One of the primary goals of XML was to make documents easily available over the Web (http://www.w3.org/TR/WD-xml-961114.html). XML has been used extensively in web-oriented research areas, and has often been termed as the lingua franca of e-commerce. XML has also been the basis of many Markup languages such as MathML (Mathematical Markup Language), CML (Chemical Markup Language), XBRL (eXtensible Business Reporting Language) etc. XML provides a cost-efficient, platform-independent way to represent persistent data. Different possibilities of data storage in XML format have been explored by Emerick (2002).

There are relatively few research activities aimed at the transformation of texts into semantically annotated XML documents. Akhtar et al. (2003) proposed a system that automates the XML mark up of electronic documents using Self Organizing Maps and inductive learning algorithms. Iyengar et al. (2002) identified the possible uses of automating XML markup of domain-specific text documents and proposed an MIML-based (Maritime Information Markup Language) system that marks up, re-arranges and extracts useful marine information.

XBRL is one of the XML based languages that is fast becoming a standard for communicating business information on the internet. XBRL is a powerful and flexible version of XML that has been defined specifically to meet the requirements of business and financial information. It enables unique identifying tags to be applied to items of financial data, such as 'net profit', 'sales' etc. Many companies are now reporting their financial data according to the XBRL specifications. (http://www.xbrl.org)

In this paper, we concentrate on extracting a limited set of financial data that can be represented in a custom-XML format, and in particular research the possibility of converting the data into standardized XBRL format. We hope to extract all the relevant financial information from electronic documents and transform it into XML data with the help of our system.

AUTOMATIC DATA EXTRACTION AND XML TAGGING

Hand tagging of textual data to XML permits a relatively small amount of input data since it is a labor-intensive process. Hand tagging is, however, a very good tool to train adaptive systems to identify the common text processing techniques that help automate the XML tagging of text data. Considerable research has been done to automatically add XML tags to data to help make the text more meaningful and easier for further processing. Winkler et al. (2002) proposed a system that uses knowledge discovery in textual databases (KDT) and processes results into a final set of clusters whose labels serve as XML tags and DTD elements. In our paper, since we propose to create customized XML documents, the need to create domain-specific DTD is eliminated, leaving the system to extract all the relevant financial information and tag in standard format.

The authors have previously developed a text extraction system (FIRST) that extracts corporate financial reports from the Wall Street Journal with the help of NLP techniques. This paper is an extension of the FIRST system to adjust it to provide satisfactory results from alternate web sources such as Reuters, and to automatically tag the extracted financial information into a customized XML format (possibly XBRL).

To create XML formatted documents, our system first analyzes the content in the document. This content-analysis process requires the system to identify the key concepts. We use natural language processing techniques as a foundation. The key concepts are identified by looking at how news articles were written in the past. We use data from sources such as the Wall Street Journal (issues published from 1987 to 1989) and Reuters (Reuters Corpus - Volume 1, Lewis et al, 2004). These previous articles allow us to find patterns in the texts. For example, the term “increase” often appears after the term “sales” and before percentage amounts. This allows us to find the percentages by which “sales” “increase” in a story.

We use KWIC index (Luhn 1960) and the CMU Statistical Language Modeling (SLM) Toolkit (http://www.speech.cs.cmu.edu/SLM_info.html) to analyze patterns that appear in the documents. The following example shows some patterns we found from the WSJ in the KWIC index file:

sales	declined	42%,	to	$53.4
sales	declined	between	1%	and
sales	declined	by	1%,	
sales	declined	slightly,	because	of
sales	declined	to	$475.6	million,
sales	declined	to	9.14	billion
sales	fell	3.7%	to	3.02
sales	fell	31%	during	the
sales	fell	33%	to	about
sales	fell	7.5%	to	99,107
sales	increased	3%	to	$477
sales	increased	3.1%	from	December,
sales	increased	3.7%	over	July

Our system is also able to identify synonyms of the terms we are interested in. We use the lexical semantic relations from WordNet as a major source of this semantic information (Miller et al. 1990, Miller 1995). WordNet helps our system to figure out, for example, that the term “increase” will have the same meaning as the term “gain,” or “grow.” After analyzing an article and converting it into information to be input into the database, the XML tagger tags the items in the database. Figure 1 shows the process of extracting financial information from online documents and generating XML formatted files.

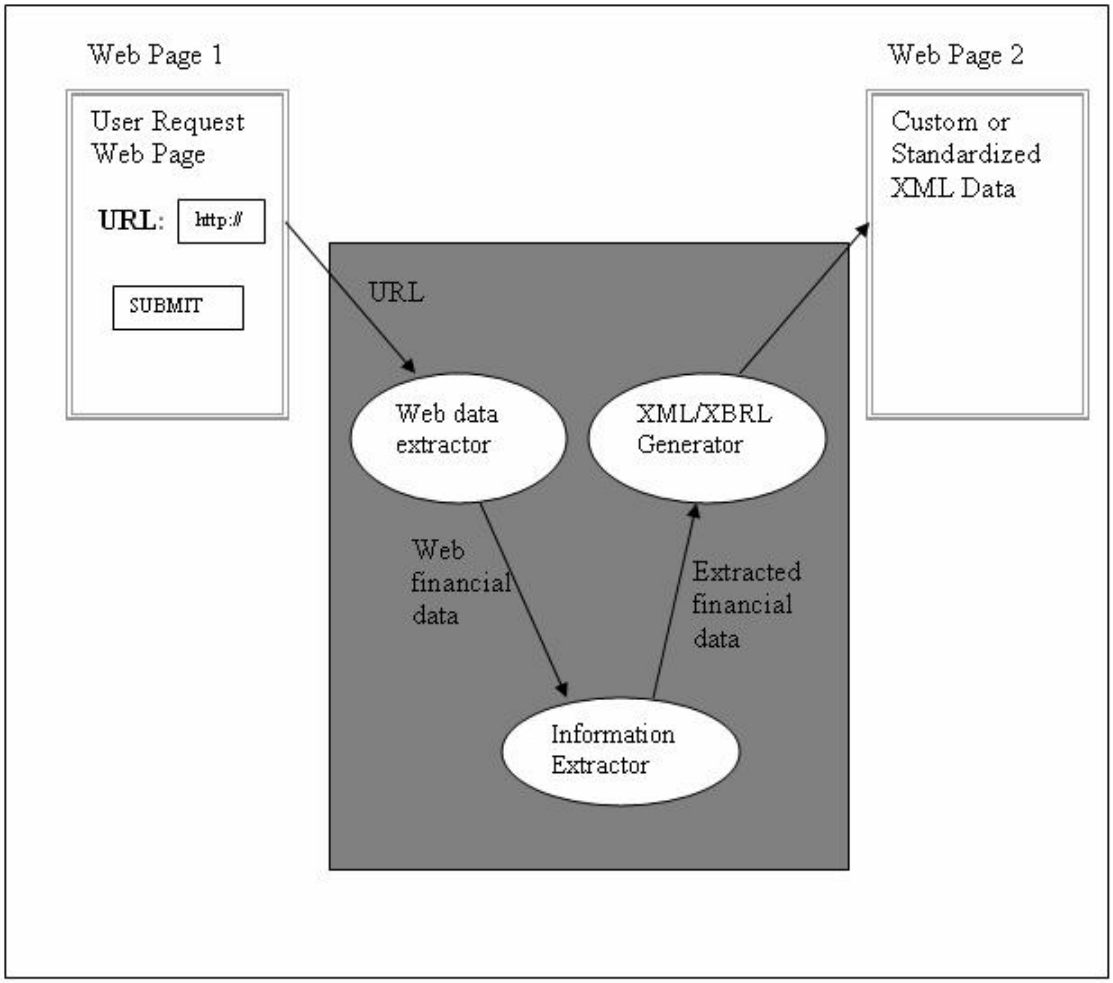

Figure 1: XML/XBRL Tagging Process

CURRENT RESULTS

The results of this project largely depend on the efficiency of the Web data extractor and the information extractor. Since this is a working paper, we list below the results obtained during the system data-training phase. Figure 2 shows an article we used from the Wall Street Journal while the extracted output is shown in Figure 3.

```
<DOC>

NEXTEL COMMUNICATIONS INC.'s fourth-quarter earnings fell 26%.

The company also forecast lower net income for 2005, due to a higher
tax rate.

Net income for the Reston, Va., wireless provider, which is in an
agreement to be acquired by Sprint Corp., declined to $471 million, or
41 cents a share, from $634 million, or 55 cents a share, in the year-
earlier quarter.

Revenue for the quarter rose 19% to $3.58 billion from $3.01 billion.

Nextel gained 955,000 new subscribers during the fourth quarter.
For the year, the company added 2.9 million customers, closing out 2004
with 16.2 million users.


Growing demand for Boost, the company's youth-oriented, prepaid
wireless service helped lift revenue.
Nextel added 755,000 Boost customers during the year and expects that
figure to increase to one million more subscribers in 2005.
Boost, which is now available in about 75% of Nextel's network, will be
available to 100% of customers soon, company executives said.

Boost's average revenue per user, in the high $30 range, is higher than
that of most other prepaid services, which are generally in the $25
range, Chief Financial Officer Paul Saleh said.
Overall revenue per user was $68 in the fourth quarter, and $69 for the
full year.

Churn, or the percentage of customers that disconnect every month, was
1.5% during the fourth quarter and 1.6% for the full year, unchanged
from 2003 levels.

For the coming year, Nextel anticipates 2.9 million new subscribers,
flat with 2004.

Shares of Nextel were down 59 cents to $28.59 in 4 p.m. composite
trading on the Nasdaq Stock Market.

</DOC>
```

Figure 2: A *Wall Street Journal* article

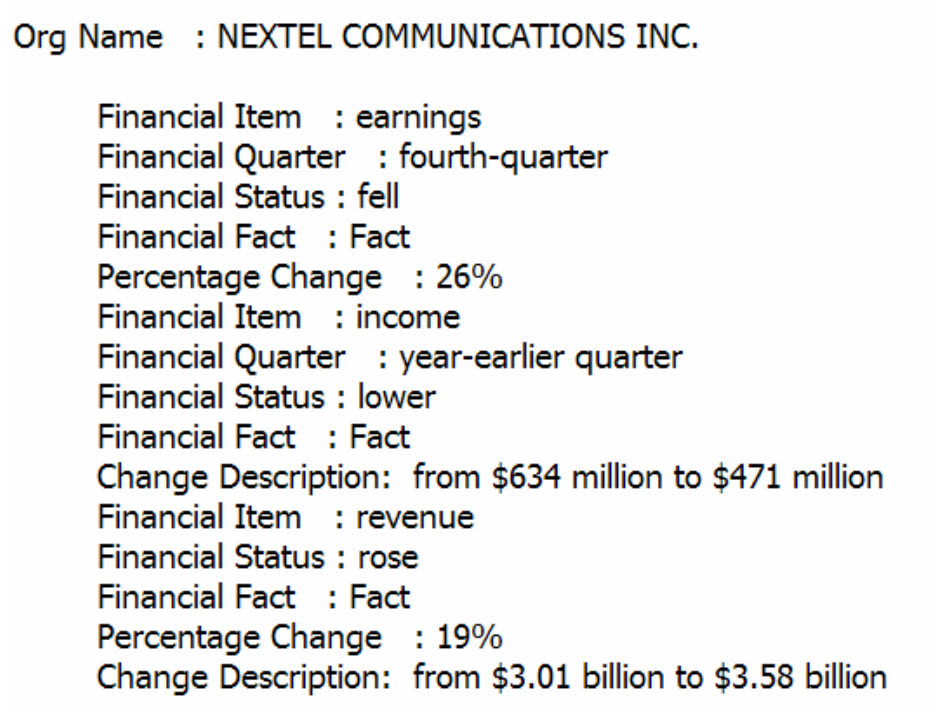

Figure 3: Extracted data from a *Wall Street Journal* article

After the system extracts information, the XML/XBRL tagger produces an XML file as shown in figure 4 and the XBRL tagged file is shown in figure 5.

```
- <Companies>
  - <Organization>
      <OrgName>NEXTEL COMMUNICATIONS INC.</OrgName>
      <OrgDescription />
    - <FinItem type="earnings">
        <Quarter>fourth-quarter</Quarter>
        <Status>fell</Status>
        <Fact>Fact</Fact>
        <ChangePercentage />
      </FinItem>
    - <FinItem type="income">
        <Quarter>year-earlier quarter</Quarter>
        <Status>lower</Status>
        <Fact>Fact</Fact>
        <ChangeFrom>from $634 million</ChangeFrom>
        <ChangeTo>to $471 million</ChangeTo>
      </FinItem>
    - <FinItem type="revenue">
        <Quarter />
        <Status>rose</Status>
        <Fact>Fact</Fact>
        <ChangePercentage />
        <ChangeFrom>from $3.01 billion</ChangeFrom>
        <ChangeTo>to $3.58 billion</ChangeTo>
      </FinItem>
    </Organization>
  </Companies>
```

Figure 4: Custom XML data

```
<?xml version="1.0" encoding="UTF-8" ?>
- <xbrl xmlns="http://www.xbrl.org/2003/instance" xmlns:link="http://www.xbrl.org/2003/linkbase"
    xmlns:xlink="http://www.w3.org/1999/xlink" xmlns:usfr-pt="http://www.xbrl.org/us/fr/common/pt/2004-08-15"
    xmlns:iso4217="http://www.xbrl.org/2003/iso4217" xmlns:xbrli="http://www.xbrl.org/2003/instance">
    <!--  Context Section    -->
  - <context id="P3MQ2FY2005">
    - <entity>
        <identifier scheme="http://www.sec.gov/CIK">789019</identifier>
      </entity>
    - <period>
        <startDate>2003-10-01</startDate>
        <endDate>2003-12-31</endDate>
      </period>
    </context>
    <!--  Units   -->
  - <xbrli:unit id="EPS">
    - <xbrli:divide>
      - <xbrli:unitNumerator>
          <xbrli:measure>iso4217:USD</xbrli:measure>
        </xbrli:unitNumerator>
      - <xbrli:unitDenominator>
          <xbrli:measure>xbrli:shares</xbrli:measure>
        </xbrli:unitDenominator>
      </xbrli:divide>
    </xbrli:unit>
    <usfr-pt:OperatingRevenue contextRef="P3MQ2FY2005" unitRef="USD" decimals="-6">3580000000</usfr-pt:OperatingRevenue>
    <usfr-pt:NetIncome contextRef="P3MQ2FY2005" unitRef="USD" decimals="-6">471000000</usfr-pt:NetIncome>
    <usfr-pt:SellingMarketingExpenses contextRef="P3MQ2FY2005" unitRef="USD" decimals="-6" />
  </xbrl>
```

Figure 5: XBRL-formatted data

On inspecting both the data formats, we can observe that the XBRL format helps us to specify the extracted data in a standard, concise fashion. The extracted financial and the source information variations stored in a relational database to allow further analysis and for study of the output variation based on differing financial resource inputs.

We compare the results from our system with the results that human experts would produce. The precision and recall rates are used to measure the system performance. They are defined as:

$$\text{Precision} = \frac{\text{The number of items that are tagged correctly}}{\text{The number of items being tagged}}$$

$$\text{Recall} = \frac{\text{The number of items tagged by the system}}{\text{The number of possible items that experts would tag}}$$

Based on preliminary results, the performance of our information extraction system, the precision and recall rates are at 85% and 87% respectively.

POSSIBLE USES OF OUR SYSTEM

We believe that our system can evolve into a web financial data extraction tool that converts unstructured text data to a structured and standardized financial XML format (XBRL). Such XML formatted data can be used by business analysis tools (such as Semansys Business Analyzer, etc.) to give business users insight about the interested company's performance. The simplicity of XML data coupled with a combination of standard Web data transfer protocols and accurate financial information extraction agents can help create a web-based financial data extraction tool that can help businesses make more informed decisions.

Some companies are beginning to promise subscription services that feed customers with XML formatted information. Companies can readily post such standardized information into their own database systems, and

design and build business application systems against them to support decision making. However, without automated tools to convert business texts to XML format, such services will remain rare and expensive.

CONCLUSION

Currently, our system extracts a relatively small set of standard financial data. This is largely due to the nature of financial information available in the articles available on the Web. We hope to make our system a standardized data extraction engine by expanding the set of financial information that our system can extract, making it more useful for business users in the future. Our system is also aimed at serving as a web service that receives XML based requests and responds with the extracted custom or standard XML data, depending on the user request.

REFERENCES

1. Akhtar, S., Reilly, R. G., Dunnion J. (2003) Automating XML markup of text documents, Proceedings of HLT-NAACL 2003. (http://acl.ldc.upenn.edu/N/N03/).
2. Cardie, C. (1997) Empirical methods in information extraction, AI Magazine, 18(4):65--80.
3. Emerick, J. (2002) Managing XML data storage. In: Crossroads, Vol. 8, No 4:6-11.
4. Gao, S., Xu, D., Wang, Y. and Wang, H. (2004) "Development of a Web-service-agents-based Family Wealth Management System", Proceedings of the Tenth Americas Conference on Information Systems, (AMCIS 2004); New York, NY, 1841 - 1850.
5. Jacobs, P. S. and Rau, L. (1990) SCISOR: Extracting information from On-Line News. Communications of the ACM, 33(11):88-97.
6. Lewis, D. D.; Yang, Y.; Rose, T.; and Li, F. RCV1: A New Benchmark Collection for Text Categorization Research. Journal of Machine Learning Research, 5:361-397, 2004. http://www.jmlr.org/papers/volume5/lewis04a/lewis04a.pdf
7. Luhn, H.P. (1960) Keyword-in-context index for technical literature (KWIC index), American Documentation 11:288-295.
8. 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.
9. Miller, G. A. (1995) WordNet: a Lexical Database for English, Communication of the ACM, Vol .38, No 11: 39-41.
10. Iyengar, R. K. and R. M. Malyankar: A Method for Automating Text Markup. Conference on Digital Government (dg.o2002), Los Angeles, California, May 2002.
11. Winkler, K. and Spiliopoulou, M. (2002) Employing text Mining for Semantic day-went in DIAsDEM. In: AI artificial intelligence, catalog of themes text Mining, 16(2002)2:27-29.