Automated Understanding of Financial Statements Using Neural Networks and Semantic Grammars

James Markovitch
Dun & Bradstreet, Search Technologies
April 1995
Email: jsmarkovitch@yahoo.com

Abstract

This article discusses how neural networks and semantic grammars may be used to locate and understand financial statements embedded in news stories received from on-line news wires. A neural net is used to identify where in the news story a financial statement appears to begin. A grammar then is applied to this text in an effort to extract specific facts from the financial statement. Applying grammars to financial statements presents unique parsing problems since the dollar amounts of financial statements are typically arranged in multiple columns, with small paragraphs of text above each column. Text therefore is meant to be read both vertically and horizontally, in contrast to ordinary news text, which is read only horizontally.

1 Introduction

Each year more information becomes available in electronic form. Some of this comes from well known, general sources such as UPI, Reuters and the Internet; still other data comes from lesser known sources such as the PR NEWSWIRE and BUSINESS WIRE, which supply financial and other information.

However, for information to be useful it must be either indexed or digested to suit each individual's or company's needs, inasmuch as it is inevitable that much of the news collected will turn out be of little or no interest. For an overview of the attempt to solve this problem via indexing techniques, and an account of indexing techniques to large data sets such as MEDLINE, see [3]. This source discusses retrieval methods and ranking algorithms that help make the data that has been archived more accessible. The methods employed take advantage of word stemming, Boolean queries, word-weighting and vectoring schemes. See Addison, *et al,* [1] for a discussion of indexing techniques specifically applied to Real-Time news.

An additional method to aid in the processing of large amounts of news requires that the news be understood. Unfortunately this understanding is not easily arrived at in software. See Shirmer and Kuehn, [5] for a discussion of understanding news via word experts and neural nets; they describe a method whose goals are similar to those described here. Also see Bearden, *et al,* [2] for detailed descriptions of how grammars may effectively mimic human understanding in limited domains.

The present article offers a partial solution to one problem in electronic news understanding: the understanding of financial statements embedded in text. Figure 1 represents the upper portion of a typical financial statement.

TYPICAL COMPANY, INC.
CONSOLIDATED FINANCIAL INFORMATION
UNAUDITED (000's)

	Three Months Ended October 31		Six Months Ended October 31	
	1993	1992	1993	1992
Net sales	$64,314	$63,831	$121,037	$122,180
Gross profit	$32,560	$34,281	$ 61,931	$ 63,963

Figure 1

Such a financial statement might be embedded within an accompanying news story that tells the reasons for the rise or fall of that company's profits, sales, etc. Characteristic of such a financial statement is its columnar

presentation of numerical information. Figure 1 has no fewer than four columns of numbers, each representing the financial results for a different time period.

For a human to read such a document presents surprisingly few problems. The column headings, which are isolated paragraphs of text suspended above their respective columns, are easily read as distinct paragraphs. These columns are then readily scanned for the sales and profit information they contain. Unfortunately, for a computer to accomplish this same visual task requires that it have the same visual sense that a human does. It is not clear how to arrive at this visual sense in software, however.

For this reason a method was developed that did not depend on a visual sense of the financial statements, but rather on its *grammatical* sense. Although this method of understanding a financial statement may not accurately reflect how a human reads such a document, it will be shown that the method is a reasonably effective way for a computer to understand financial statements.

2 Using Neural Nets to Find a Financial Statement

Before a financial statement may be examined for its grammatical sense, it is necessary first to locate one. This task is surprisingly difficult to accomplish via programming logic. The first problem is that there is no single recognizable feature that marks the start of a financial statement. And secondly, there are blocks of text, particularly news story headlines announcing earnings results, that look like the start of a financial statement, but are not. Experience has shown that neural networks are effective in recognizing handwriting, speech and visual patterns though a process of statistically based feature extraction [4]. For these reasons a backpropagation network was used to find the start of financial statements in each news story.

In order for the neural net to find the start of the financial statement, a batch program was written to supply it with a "moving window" of fifteen lines of text, where the neural net's single output was trained to produce a score of 1 if the fourth line of this "window" was the start of a financial statement, and to produce a 0 otherwise. For each of the 15 lines the neural net was told about:

1) the line's length,
2) the percentage of letters in the line that were in capitals,
3) the number of leading spaces in the line,
4) the number of embedded spaces in the line,
5) the percentage of the characters in the line that were digits, and,
6) a Boolean value that told whether or not the line was centered.

In addition the neural net was told about two other characteristics of the line: specifically whether certain keywords, such as *inc*, and *company* occurred in the line; and secondly, whether words such as *month*, *quarter*, *year*, *financial*, and *consolidated* were present. In total, the neural net had 120 inputs, representing 8 characteristics, pertaining to 15 lines.

When preparing the training data (i.e., development data) and test data (i.e., holdout data), a line was regarded as containing the start of a financial statement if it had a company name, at least one date, and at least one column of numbers. The requirement that a company name be present was an arbitrary requirement that might not be suitable in all circumstances. The line containing the company name was regarded as the start of the financial statement.

3 Using Semantic Grammars to Parse a Financial Statement

Once the start of a financial statement has been identified, it must be analyzed and understood. When the financial statement of Figure 1 is rearranged as a stream of text, it appears as in Figure 2.

TYPICAL COMPANY, INC. <RETURN> CONSOLIDATED FINANCIAL INFORMATION <RETURN> UNAUDITED (000's) <RETURN> Three Months Ended Six Months Ended <RETURN> October 31 October 31 <RETURN> 1993 1992 1993 1992 <RETURN> Net sales $64,314 $63,831 $121,037 $122,180 <RETURN> Gross profit $32,560 $34,281 $ 61,931 $ 63,963 <RETURN>

Figure 2

A close examination of this stream of text and others similar to it reveals underlying regularities that may be exploited by using a semantic grammar. Semantic grammars, which are described in [2], are an effective means for understanding sentences within a restricted domain. The world of financial statements is clearly such a restricted domain, but can financial statements be viewed as sentences with their own grammar?

To resolve this question several hundred financial statements were analyzed to uncover sentence-like regularities. From this analysis a context-free grammar and a lexicon emerged that allowed a large percentage of financial statements to be processed.

HEADING ::= for-the 12-PL end for-the 34-PL end RETURN end 12-PE end 34-PE RETURN 1-Y-E 2-Y-E 3-Y-E 4-Y-E RETURN

12-PL ::= 12-N-L 12-T-L

12-PE ::= 12-M-E | 12-M-E 12-D-E

34-PL ::= 34-N-L 34-T-L

34-PE ::= 34-M-E | 34-M-E 34-D-E

Figure 3

A portion of the context-free grammar is shown in Figure 3, in particular, that portion of the grammar that helps interpret four-column financial statements. In this figure and the remaining figures the symbol *::=* is read as *is defined as*, which follows [2].

12-N-L **::=** *3* | *6* | *9* | *12* | *three* | *six* | *nine* | *twelve*

12-T-L **::=** *week* | *month* | *year* | *weeks* | *months*

34-N-L **::=** *3* | *6* | *9* | *12* | *three* | *six* | *nine* | *twelve*

34-T-L **::=** *week* | *month* | *year* | *weeks* | *months*

12-M-E ::= *Jan* | *Feb* | *Mar* | *Apr* | *May* | *Jun* | etc.

12-D-E ::= *first* | *second* | *third* | *fourth* | *fifth* | etc.

34-M-E ::= *Jan* | *Feb* | *Mar* | *Apr* | *May* | *Jun* | etc.

34-D-E ::= *first* | *second* | *third* | *fourth* | *fifth* | etc.

1-Y-E ::= *1990* | *1991* | *1992* | *1993* | *1994* | *1995* | etc.

2-Y-E ::= *1990* | *1991* | *1992* | *1993* | *1994* | *1995* | etc.

3-Y-E ::= *1990* | *1991* | *1992* | *1993* | *1994* | *1995* | etc.

4-Y-E ::= *1990* | *1991* | *1992* | *1993* | *1994* | *1995* | etc.

for-the ::= *for the* | *for*

end ::= *ending* | *ended*

Figure 4

Likewise a portion of the lexicon used is shown in Figure 4.

4 One Special Problem

One characteristic of grammars that are designed to understand natural language is that the form of the representation is unimportant. Accordingly, if in a grammar a verb phrase is represented as *VP, VerbP* or *V-P*, it does not alter the effectiveness of the grammar in the slightest. This is not entirely the case when parsing financial statements, however. With financial statements it is necessary, once parsing is complete, to distinguish between a dollar amount found in the first column, and a dollar

amount found in the second column. It is also necessary to distinguish between a "period end date" that applies only to column one, and a period end date that applies to columns three and four. The easiest way to do this is to build "cases" into the language that are analogous to the cases used in normal languages, and *to reflect these case differences by using the symbols of the grammar itself*. Accordingly, a period end date that applies only to column one is represented in the grammar as *1-PE*, while a period end date applying to columns three and four appears as *34-PE*. This small restriction on the formation of the grammar has valuable practical consequences when at a later time the meaning of the parsed financial statement must be determined. It allows the components of the parse tree to be processed with relative ease to find what period end dates, and what period lengths, apply to which columns.

5 Neural Network Technical Details

The neural network used had a hidden layer of 16 nodes, in addition to its input layer of 120 nodes, and its single output node. The training data was composed of 26,880 lines of text, which contained 228 financial statements. This data was captured from the PR NEWSWIRE and the BUSINESS WIRE on Nov. 11, 1993 and Nov. 24, 1993. The single output was trained to be a 1 in the presence of a financial statement, and a 0 otherwise. Training on the data was terminated when the worst error for any member of the training data set was no greater than .25. This was accomplished after 135 training iterations.

The test data was composed of 13,736 lines of text, which contained 125 financial statements. This data was captured from the PR NEWSWIRE and the BUSINESS WIRE on Dec 10, 1993 and Dec 17, 1993. When validating using the test data, an output node score greater than .1 was treated as signifying the presence of a financial statement.

6 Neural Network Results

When the network was run against the test data, it proved effective. 118 of 125 financial statements were identified for a rate of 94.4%. Among the 13,611 lines that contained no statement, just twelve were wrongly identified as financial statements, for an error rate of .0881% (less than 1 in a 1,000).

In general the network performed very effectively, with only occasional lapses for "unusual" financial statements. An unusual financial statement might be one whose company name is not centered, but rather appears flush left, or one whose company name lacks the word *incorporated*, *co.*, or *inc.*, etc.

7 Grammar Results

A program employing a more robust version of the grammar and lexicon described earlier proved fairly effective in parsing financial statements. Specifically, the headings of 72 of the 125 financial statements were understood by the grammar (here the word *heading* refers to the lines that tell start and end dates, as well as period lengths). The grammar was slanted towards understanding income statements and it frequently failed in circumstances where it met with some other form of statement, such as a statement of cash flow. Adding new statement types to the grammar can readily expand its comprehension, however. Currently it supports only sixteen.

Problem cases arose as a consequence of the wide variety of financial statements present in news stories. In particular, "unique" financial statements typically appeared at times other than the end of a quarter, and sometimes appeared to be edited by hand for special release.

Still other problems arose from the phrases *in thousands* and *in millions* that sometimes acted as a multiplier on all of the dollar amounts of the financial statement, or sometimes on just a limited portion of it. It is hard to anticipate the variety of ways this multiplier might appear, and failure to anticipate correctly leads to a very

large error. Complex code had to be written to handle these cases, as well as to search the lines of text following the headings for the specific financial information required: e.g. net income, total sales, etc. This task was done using traditional programming methods.

One lesson learned is that the neural network ideally should provide more information than merely where a financial statement starts. For instance, it is useful to know where a statement ends (so as to avoid "falling through" to unrelated text and data). Likewise it is useful to know specifically on which line the grammar should be applied. This line is often many lines after the company name. Lastly, it appeared that a moving window of just 15 lines was too short for a proper understanding of some statements.

8 Conclusions

The above problems notwithstanding, it proved possible to process financial statements embedded in news stories. The method used also proved flexible enough to accommodate new financial statement types as they were discovered. In addition, the grammar used showed a large degree of resistance to misinterpretation. If text other than a financial statement was presented to the grammar, it would readily reject it as unparsable. Currently, the system described has not been developed into a deployable system, but the results achieved here indicate that these methods can be used to create a practical system to automate the understanding of financial statements. The general conclusion is that grammars may prove more effective in a wider array of contexts than is readily apparent.

9 References

[1] Edwin Addison, Judith Feder, Paul Nelson and Tom J. Schwartz, "Extracting and Disseminating Information from Real-Time News." In *Proceedings of the Second International Conference on Artificial Intelligence Applications on Wall Street*. New York, NY, April, 1993, Gaithersburg, MD: Software Engineering Press.

[2] Colin Beardon, David Lumsden, and Geoff Holmes, 1991. *Natural Language and Computational Linguistics An Introduction*. Chichester, England: Ellis Horwood Limited.

[3] W. B. Frakes, and R. Baeza-Yates, 1992. *Information Retrieval: Data Structures and Algorithms*. Englewood Cliffs, NJ, Prentice-Hall.

[4] Marilyn McCord Nelson, and W. T. Illingworth, 1991. *A Practical Guide to Neural Nets*. Reading, MA, Addison-Wesley.

[5] Kai Schirmer and Michael Kuehn, "Fact Extraction from Financial News." In *Proceedings of the Second International Conference on Artificial Intelligence Applications on Wall Street*. New York, NY, April, 1993, Gaithersburg, MD: Software Engineering Press.