Table Classification Using Both Structure and Content Information: A Case Study of Financial Documents

Quanzhi Li, Sameena Shah, Rui Fang
Research and Development
Thomson Reuters
3 Times Square, NYC, NY 10036
{quanzhi.li, sameena.shah, rui.fang}@thomsonreuters.com

Abstract— **Tables are significant document components. Table extraction and classification are critical for us to explore, retrieve and mine knowledge encoded in tables. This paper presents a learning based approach for classifying tables based on their content and structural information, with focus on financial document tables. To the best of our knowledge, this is the first study on classifying tables in financial domain, and also the first study of table classification based on its semantics, a more fine-grained level than previous studies. The experimental results show that it can effectively classify financial tables. We also analyzed what features are important and how to generate them. The feature identification and generation approach can potentially apply to other domains.**

Keywords-financial table; table classification; financial document; language model

I. INTRODUCTION

Many critical data are encoded in tables, such as product price information, corporate financial report data, and scientific experiment result [7, 18]. To retrieve them, we need to automatically extract those tables and classify them into categories they belong to. After the extraction and classification steps, other applications can then search, analyze, manage, discover or build new knowledge upon them. Examples of related applications include data retrieval and mining, summarization, relationship discovery, and comparative shopping.

Previous studies mainly focus on table extraction from documents, which basically is, through a set of structure features, to determine whether or not a document block is a table [6, 7, 16]. There are some studies related to table classification for web pages, but most of them focus on identifying if a table is a genuine or non-genuine table [17, 10]. A non-genuine table is for page layout purpose, providing formatting or navigational functions, not for encoding information. Beyond the genuine vs. non-genuine table problem, the study presented in [4] identifies general types of tables on the web, such as calendar, form, navigational, enumeration, and vertical listing tables. The study from [10] expands table categories to a more fine-grained level, identifying table types in scientific research papers, such as tables about experimental methods and experimental results. But it only uses table caption and the reference text following the table as features. It does not consider table structure or the content in the table, which represents a table's semantic meaning.

A document usually contains different types of tables, in terms of their content or semantics. For example, tables about product prices and tables for sales offices have different semantic meanings. In many documents, the most important information is in tables, and this is especially true in financial documents. For instance, in a company's annual financial report, the most important information is income statement, cash flow and balance sheet, and they are presented in three separate tables, Income Statement table, Cash Flow table and Balance Sheet table, correspondingly. These three tables are called "the big three" because of their importance in an annual financial report. In this study we focus on classifying these three types of tables, and the experiments were conducted on tables extracted from corporate financial reports. Part of the reason is that it has practical application to financial data workflows and knowledge mining systems, whose table classification processes currently are mainly based on rules and manual efforts. The approach proposed in this study can also apply to other types of tables and domains.

The main contributions of this paper are as follow: To our knowledge, this is the first study of table classification based on table semantics. It provides a more fine-grained level than previous studies. It is also the first study on classifying tables in financial domain. The feature identification and generation approach can apply to other types of tables and domains as well, with small adjustments depending on the types of data in table cells.

In the following sections, we first discuss the related studies, then we present the data collection, feature generation and classification methodology, and finally we analyze the experiment results.

II. RELATED STUDIES

There are several previous studies focusing on table detection from documents, such as PDF documents, plain text files or web pages [17, 13, 7, 20]. Fang et al. [6] identify a set of features that can be used to segregate headers from tabular data, and use them to build a classifier to detect table headers. They can detect table headers using random forest model with 92% of accuracy. Pinto et al. [16] use CRF algorithm for table extraction and compare them to

978-1-4673-9005-7/16/$31.00 ©2016 IEEE

HMM. Their work shows that CRF can use many rich features than HMM, and therefore CRF outperforms HMM greatly. They use sequential features, such as lines before and after a table, to detect table.

Some previous studies have tried to classify tables, but they mainly focus on identify genuine tables (e.g., attribute/value pairs tables, relational tables) and non-genuine tables [17]. Non-genuine tables refer to tables for formatting. Crestan and Pantel [4] report a tool called TabEx, which can identify tables from the web and put them into several categories, such as formatting, vertical listing, form, calendar, horizontal listing, and enumeration. The main features they used are html element features and layout features. A similar work done by [18] tries to classify tables into nine general categories with focus on table attributes and values. Crestan and Pantel [4] argue that the types in [18] can be collapsed into just two categories: vertical listing and horizontal listing, basically two categories used by Crestan and Pantel.

The studies of [17, 1, 2, 3, 5, 8] focus on extracting the relations from tables, which involve understanding the internal structure of a table. This type of studies show that why our study of table classification based on their content is very useful, because only after correctly classifying tables into semantic categories, the relationship discovery from these tables will be more accurate.

As mentioned in the previous section, previous studies mainly focus on table detection or general classification of tables into usage types based on their table structure. In this study, we classify tables based on their content, e.g. it is about a company cash flow or balance sheet.

III. TABLE CLASSIFICATION IN FINANCIAL DOCUMENTS

This study focuses on financial tables, and the experiments were conducted on financial tables, therefore, in the following subsections we first give an introduction to financial documents and tables, and then we present the data collection process, feature generation approach, experiment design and the learning algorithm tested.

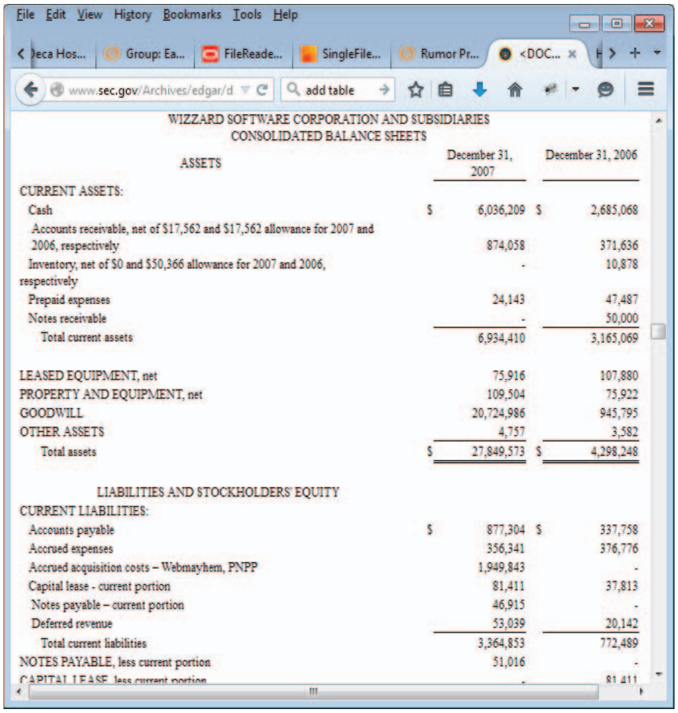
A. Financial Document and Table

The experiments of this study were based on the three most important tables in a company's 10-K report. A 10-K document is an annual report required by SEC, which provides a comprehensive summary of a company's financial performance. Lots of information are included in 10-K, such as financial statements, equity, executive compensation, organizational structure, subsidiaries, and company history, among other information. Its length can reach to a couple of hundred pages, and it contains lots of tables, with different formats and contents.

Companies traded on large stock markets, such as NASDAQ and NYSE, are from different countries and industries, and their sizes are also different. Consequently, their 10-K documents may contain different number of tables and those tables' formats may be different. One challenge for classifying financial tables is that there are hundreds of tables in a document, and the structure and content may vary by company and the year of the financial report. Table 1 presents the table distribution based on 1,000 10-K files. It shows that on average each document has about 270 tables.

Table 1. Number of tables in a 10-K document

Metric	Value
Average	270.8
SD	236.4
Max	1244
Min	38



WIZZARD SOFTWARE CORPORATION AND SUBSIDIARIES
CONSOLIDATED BALANCE SHEETS

ASSETS		December 31, 2007		December 31, 2006
CURRENT ASSETS:				
Cash	$	6,036,209	$	2,685,068
Accounts receivable, net of $17,562 and $17,562 allowance for 2007 and 2006, respectively		874,058		371,636
Inventory, net of $0 and $50,366 allowance for 2007 and 2006, respectively		-		10,878
Prepaid expenses		24,143		47,487
Notes receivable		-		50,000
Total current assets		6,934,410		3,165,069
LEASED EQUIPMENT, net		75,916		107,880
PROPERTY AND EQUIPMENT, net		109,504		75,922
GOODWILL		20,724,986		945,795
OTHER ASSETS		4,757		3,582
Total assets	$	27,849,573	$	4,298,248
LIABILITIES AND STOCKHOLDERS' EQUITY				
CURRENT LIABILITIES:				
Accounts payable	$	877,304	$	337,758
Accrued expenses		356,341		376,776
Accrued acquisition costs – Webmayhem, PNPP		1,949,843		-
Capital lease - current portion		81,411		37,813
Notes payable – current portion		46,915		-
Deferred revenue		53,039		20,142
Total current liabilities		3,364,853		772,489
NOTES PAYABLE, less current portion		51,016		-
CAPITAL LEASE, less current portion		-		81,411

Figure 1. Part of a Balance Sheet Table.

B. Data Collection

The 10-K documents used in this study were scraped from SEC website. Each document is an html file, with size ranging from 500 KB to 10 MB. After these files were downloaded, they were loaded into an html editor for annotators to label them. Our target tables were Income Statement (IS), Balance Sheet (BS) and Cash Flow (CF), and so these three tables were marked in the html file by adding a special attribute to the html table element. All other types of tables were considered as Other (OT) type, and there was no need to manually label them. From now on, when we refer to the four types of tables, we will use IS, BS, CF and OT, instead of their full names. We used an html parser to parse these 10-K documents, and extracted all the tables in a document.

The total number of tables used in the experiments was 2,500. Each of the three target categories had 390 tables, and the OT class had 1,330 tables. This caused the table distribution in our experiments a little bit unbalanced. We chose to have more OT tables was because, in the 10-K documents, OT tables greatly outnumber the other three types of tables. Basically in each document, each target class has only one table, but there are on average 268 tables for OT class. The OT tables have various contents and formats. We wanted to have more OT type tables in the experiments to make the data more representatives, but also wanted to make the training data close to balanced. The 1,330 OT tables were randomly selected from the extracted OT tables.

Different tables have different numbers of rows and columns, and they have different structures. Table 2 shows the statistics regarding rows and columns of the tables used in this study. Figure 1 shows part of a balance sheet table.

C. *Feature Identification and Generation*

In this study, two types of features are used in classifying a table: content feature and structure feature. Content features include table title, row headers, column headers, and also the ratio of numeric cells to all other table cells. Structure features include the number of rows and the number of columns. We observed that different types of tables usually have different amount of rows. But their column counts are not so different. Very few tables use table header or caption html tags, such as *th* and *tfoot*. Our html parser can correctly extract these features based on the table structure and cell contents.

Table 2. Distribution of row count and column count for the four table types

Dimension	Table Type	Max	Min	Mean	SD
Row	CF	79	37	40.3	9.6
	BS	52	17	35	8.8
	IS	49	13	25.8	11.2
	OT	66	3	14.2	12
Column	CF	5	3	3.6	0.6
	BS	5	3	4.2	0.7
	IS	5	3	3.9	0.47
	OT	12	2	3.9	1.57

Language Model: For table title, row headers and column headers, language models are used to compute their feature values. The BerkeleyLM toolkit is used to generate the language models [14, 19]. Basically, each type of table has three language models built - for table title, row headers, and column headers, respectively. There are totally 12 language models built for the four table classes.

In the experiments, because the cells of the big three tables contain mainly numeric values, and their contents are not used as features, therefore, there is no language model built for them. But in other table classification applications, we may need to build a language model for table cell data, if they are used as one of the classifier features. To demonstrate this, and also to show that the classification and feature generation approach proposed in this study can apply to other types of tables with appropriate adjustment, we also conducted an experiment on classifying *executive officers* table whose structure and content are different from the big three table. The cells of *executive officers* table contain mainly textual data, which are the titles of the officers. And its row headers are names of these officers. The result for classifying *executive officers* table is reported in the subsection D of the Experiment Result section. Bigram and unigram are used together to build the language models. These language models are built using the annotated data.

There are other alternatives of generating feature values for textual features, e.g. title and headers in this study. One option is to use cosine similarity based on some sort of term weight schema, such as the tf-idf method. We also tested this option, and found that there was no big difference between using language model and cosine similarity with tf.idf in terms of classification performance. Therefore, in this paper, we just report the result using language model for computing feature values for title, row headers and column headers.

Title: usually each table will have a title, although it is not always the case. Our program parses the two sentences preceding a table to determine its title. Based on our manual check, most of the time, the titles are correctly identified, with an accuracy of 96%. The feature value of title is generated based on the corresponding language model. The values of the next two features, row headers and column headers, are also generated based on their corresponding language models.

Row Headers: for classifying financial tables, we expect that row headers will play an important role, since in many financial data tables, the columns are usually date period (year or quarter), but the row headers are the real concepts. Many tables use lots of cells for formatting purpose and those cells don't have any content. Our program is able to handle those cells and identify the headers correctly.

Column Headers: the column headers in many financial tables, especially the big three tables in our case, are date period. In order to build a meaningful language model for column headers, tokens in column headers are normalized. For example, all date tokens are normalized to symbol "DATE".

Ratio of Numeric Cells: financial tables have many numeric values. This feature may help distinguish the big three tables from other types of tables, such as a table about the company's directors, whose cells contain mainly textual data, instead of numeric data.

Number of Rows: this feature may differentiate some tables. Table 2 shows that the number of rows may be pretty different for different types of tables. For instance, the table about company directors and executives will have fewer rows than the IS table.

Number of Columns: the number of rows may be useful for identifying certain tables, but we don't expect it to have a big impact. This can be observed from Table 2, which shows that, on average, the four types of tables have similar number of columns. But this may not be the case for certain OT tables, which is why the standard deviation for OT is larger than that of other table types.

Other Features: there are other features that can be exploited, but are not used in this study. These include the heading of the section where the table is, the table type preceding the current table, the textual content following this table, and whether a row header is indented or not.

D. Methodology and Algorithms

We take identifying the IS/BS/CF table task as a classification problem. Several learning algorithm are used in this experiment. Three feature settings are tested in this study: 1. using just table title, 2. using just the table row headers, and 3. integrating all the features discussed in last section together. The first two settings are considered as baselines. Previous studies about table classification mainly focus on classifying tables into two general categories: genuine vs. non-genuine. This study is to classify tables in terms of their content and semantics. Therefore, there is no baseline from previous studies for us to compare to. Another reason we choose title and row headers as baselines is that, usually the title can represent a table, and by just looking at the title one may already be able to judge what type of content this table is about. But this is not always the case. For instance, some companies' 10-K files have several tables having the phrase "income statement" appearing in their titles, but actually only one table is the complete and correct Income Statement table for this company. Other tables may be income statements for its subsidiaries, divisions, or certain products.

Depending on the table type, row headers may also play an important role in identifying some financial tables, as mentioned before. We also wanted to see how they would perform without other features. For example, in Figure 1, the row headers in a balance sheet table have terms like "assets", "goodwill" and "liabilities", which may not appear in other types of tables.

IV. EXPERIEMNT RESULTS

A. Result of the Two Baselines

We used the Weka toolkit [9] for training, validation and testing. Table 3 and 4 present the experiment results of the two baselines – using just title and just row headers. We tested several learning algorithms for each baseline, including SVM, logistic regression, Naïve Bayes, and others. The ones listed in Table 3 and 4 have the best performance. From Table 3, we can see that the performance of using just title is not bad. Its overall accuracy is 0.88 (not shown in the table). From Table 4 we can see that using just row headers, whose overall accuracy is 0.70 (not shown in the table), does not perform so well.

Table 3. Performance of using just table title as feature. The result is based on Logistic regression classifier

Table Type	Precision	Recall	F measure
CF	0.777	0.905	0.836
BS	0.78	1	0.877
IS	0.792	0.742	0.766
OT	1	0.887	0.94

Table 4. Performance of using just row headers as feature. The result is based on SVM classifier

Table Type	Precision	Recall	F measure
CF	0.57	0.896	0.697
BS	0.347	0.237	0.282
IS	0.437	0.303	0.358
OT	0.89	0.903	0.896

In order to limit the problem of overfitting and make the model generalize to an independent dataset, all the experiments in this study are conducted using a 10-fold cross-validation. Each round of the 10 folds partitions the data into two complementary subsets, 10% as test data and 90% as training data. Validation results are averaged over the rounds.

B. Feature Selection

Before applying all the features, we analyzed the distribution of feature values, to see how these features perform. We used a best-first feature selection method to find the best set of features. It starts with an empty set of features and searches forward to find the best set of features. The merit value of this selection process is 0.80. The result shows that the best feature set are: title, row headers, column headers, number of rows and the ratio of numeric cells. The *Number of columns* feature is not in the best feature list, which is not a surprise. This actually can be observed from Table 2, which shows that column count doesn't change much over different types of tables, and so it is not a good feature for differentiating the four types of tables.

We also tried Ranker method with information gain for feature selection. The result is same as the best-first approach. Both approaches show that column count is not so useful.

Table 5. Result of using all features with feature selection. The result is based on SVM classifier

Table Type	Precision	Recall	F measure
CF	0.981	0.969	0.972
BS	0.797	0.813	0.805
IS	0.957	0.955	0.956
OT	0.931	0.927	0.929

C. *Results of Using All Features*

Table 5 presents the result of using all the features after feature selection. SVM has the best performance, with an overall accuracy of 0.93. Other models were also tested but their performances were not as good as SVM. This feature set outperformed the two baselines significantly at the level of p=0.01. The result shows that title, row headers and column headers together can make a much better prediction than any of them alone.

D. *Result of Classifying Executive Officer Tables*

As explained in the language model feature paragraph in the subsection C of section III, we also conducted an experiment on classifying executive officer table, since it has a different table structure and cell contents from the big three tables. We want to see how our approach performs on tables other than the big three. For executive officers table, since its cells are textual data, we build a language model to represent its cell content and use it as one of the features. Other features are similar to the big three tables. 400 executive officer tables are annotated and, together with 2,100 other types of tables, are used in this experiment. The result shows that its accuracy is 0.95 using SVM classifier. Applying feature selection process on its features shows that the cell content, title and column headers are in the best feature set. Row headers are a bad feature since they are basically the names of executive officers, which vary greatly by company. The table cell content is an important feature, because cells contain officers' titles, which do not vary much by company. This experiment shows that the classification and feature generation approach used for identifying big three tables can apply to other types of tables and achieve a good result, with appropriate feature adjustment.

V. Future Work and Summary

In this study, we directly classify tables into the target classes based on their content and structure. One of our future work would be to classify the tables into genuine vs. non-genuine first, and then apply this classification method to classify the genuine tables, to see how they perform.

This study focuses on classifying financial tables, but the methodology can be used in other domains. Our research plan is to expand this study to other financial documents, such as 10-Q and 8-k, as well as documents in other domains, such as intellectual property and product analysis.

In this paper, we show how we classify tables in financial documents. It focuses on the big three tables in company financial reports: income statement, cash flow and balance sheet. The main contributions of this paper are: To our knowledge, this is the first study of table classification based on table semantics or content, a more fine-grained level than previous studies. This is also the first experiment on financial table classification. The feature generation approach can also apply to other types of tables and domains.

Accurately classifying document tables will help document analysis, information retrieval, data mining and other applications, by making data more accurate and available for retrieval and relationship discovery

References

[1] Cafarella, M. J.; Halevy, A.; Wang, D. Z.; Wu, E.; and Zhang, Y., WebTables: Exploring the Power of Tables on the Web. In Proceedings of the 34th International Conf. on VLDB, 2008.

[2] Cafarella, M. J.; Halevy, A.; Zhang, Y.; Wang, D. Z.; and Wu, E. Uncovering the Relational Web. In WebDB, Vancouver, Canada, 2008.

[3] Chen, H.; Tsai, S.; and Tsai, J. 2000. Mining Tables from Large-Scale HTML Texts. In Proceedings of COLING-2000. Saarbrücken, Germany.

[4] Crestan, E.; Pantel, P., EWeb-scale Table Census and Classification. In Proceedings of WSDM, Hong Kong, China. 2011

[5] Elmeleegy, H.; Madhavan, J.; and Halevy, A. Harvesting Relational Tables from Lists on the Web. In Proceedings of the VLDB, 2009

[6] Fang, J.; Mitra, P.; Tang, Z.; Giles, C.L., Association for the Advancement of Artificial Intelligence. 2012

[7] Gazen, B. and Minton, S. Overview of Autofeed: An Unsupervised Learning System for Generating Webfeeds. In Proceedings of AAAI-06. Boston, MA. 2006

[8] Gatterbauer, W.; Bohunsky, P.; Herzog, M.; Krupl, B.; and Pollak, B., Towards Domain-Independent Information Extraction from Web Tables. In Proceedings WWW-2007, Banff, Canada. 2007

[9] M. Hall; E. Frank; G. Holmes, B. Pfahringer; P. Reutemann and I.H. Witten., The WEKA data mining software: an update. SIGKDD Explore, 2009.

[10] S. Kim; K. Han; S. Kim and Y. Liu, Scientific table type classification in digital library. Document Engineering, Paris, France, Sept. 4-7, 2012

[11] S. Kim and Y. Liu, Functional-Based Table Category identification in Digital Library. In Proceedings of the 11th International Conference on Document. Beijing, China, 2011

[12] Lin, D.; Zhao, S.; Qin, L.; and Zhou, M. Identifying Synonyms among Distributionally Similar Words. In Proceedings of IJCAI-2003, Acapulco, Mexico. 2003

[13] Y. Liu; K. Bai; P. Mitra and C. Giles, tableSeer: Automatic table metadata extraction and searching in digital libraries.

Joint Conference of Digital Library. Vancouver, Canada. June 2007.

[14] A. Pauls and D. Klein, Faster and smaller N-gram language models. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies. 2011

[15] Penn, G.; Hu, J.; Luo, H.; and McDonald,, R. Flexible Web Document Analysis for Delivery to Narrow-Bandwidth Devices. In Proceedings of the Sixth International Conference on Document Analysis and Recognition. Seattle, WA. 2001

[16] Pinto, D.; McCallum A.; Wei ,X and Croft, B.W., Table Extraction Using Conditional Random Field. In Proceedings of SIGIR, Toronto, Canada. 2003

[17] Wang, Y. and Hu, J. A Machine Learning Based Approach for Table Detection on the Web. In Proceedings of WWW-2002. Honolulu, Hawaii. 2002

[18] Yoshida, M.; Torisawa, K.; and Tsujii, J. A Method to Integrate Tables of the World Wide Web. In Proceedings of Workshop on Web Document Analysis. 2001

[19] BerkeleyLM: a library for estimating storing large n-gram language models in memory and accessing them efficiently. https://code.google.com/archive/p/berkeleylm/

[20] Adelfio, Marco David, Automated Structural and Spatial Comprehension of Data Tables, http://drum.lib.umd.edu/handle/1903/16410, 2014