International Journal of Computer Engineering and Information Technology

VOL. 9, NO. 2, February 2017, 37–40
Available online at: www.ijceit.org
E-ISSN 2412-8856 (Online)

Text Parsing with Markov Logic Network

Nan Wang

University of Michigan, 599 N Mathilda Ave, Sunnyvale, CA 94085, USA

nancywang0932@gmail.com

ABSTRACT

This document describes a novel way to extract structure information from plain text using Markov Decision Process. In the age of big data, unstructured information such as text, photos and videos becomes abundant. However, data warehouse requires structured data with well-defined schema. It has been a challenge for the computer science community to extract useful data under strict schema from unstructured data schema. Here we proposed an automated system that is able to understand and infer the most likely counterpart in text stream that corresponds to a filed under the requested schema. The designed algorithm formulated the plain text using context dependent grammar with various weights, which would be sued to decide which field of the structured schema a particular piece of unstructured data belongs to. A machine-learning algorithm is used to learn the weights from training data. We implemented this automated system and applied it to extract schema data from plain US bankruptcy petition forms.

Keywords: *Information Retrieval, Markov Logic Network, Regular Expressions, Big Data.*

1. INTRODUCTION

The Internet encapsulates a vast range of useful information that is usually particularly formatted, which makes it difficult to extract relevant data from various sources. One of the most popular standards for transmitting and storing data on the Web is the Portable Document Format (PDF). In particular, various forms such as tax return forms and college applications forms are most likely to store in the PDF format. However, this file format, although ideal for printing, is generally considered “view-only” since Adobe doesn't offer a general data extraction tools for PDF forms. Therefore, the availability of robust, flexible Information Extraction (IE) systems that transform the PDF forms into program-friendly structures such as a relational database will become a great necessity. There do exist text extraction tools and libraries such as iText [1] and Apache PDFBox [2]. However, those extracted text are unstructured, or semi-structured at best. Therefore, the problem of extracting information from PDF forms reduces to the increasing attractive problem of automatically discovering useful knowledge from electronic texts. A variety of recent work on statistical models had aimed at recovering structured data from unstructured text on the Web. Those statistical models applied techniques from the artificial intelligence (AI) field to incorporate deterministic domain knowledge into statistical models that are tolerant to errors and noise in the input texts. Markov Logic Networks (MLNs) are one of the most general approaches, which merge two kinds of models: probabilistic graphical models, namely Markov Random Fields (MRFs), and first-order logic, and gain the representation benefits from both.

In this report, we employed MLN to capture attribute content properties from plain PDF forms. We first extract text sequence from PDF files using iText library. Then for each query attribute, we construct a MLN that leverage both content and structural cues to infer and detect the corresponding attribute values.

2. BACKGROUND

Starting in 2001, for Public Access to Court Electronic Records (PACER), an electronic public access service of United States federal court documents, was being made available over the Web. This opens a great opportunity for scholars to harvest government data and to study the social economic effects of legal decisions made by US government.

On October 17, 2005, the so-called Bankruptcy Abuse Prevention and Consumer Protection Act (BAPCPA), which greatly increases the costs and standards for filing consumer bankruptcy, was enacted by the U.S. congress. The reform is mainly due to tighter regulations requested by major credit card and loan companies to reduce their risks. Prior to the reform, the U.S. bankruptcy law was arguably the most pro-debtor, and policy makers sought to rule out opportunists who were abusing the bankruptcy system, the most famous example perhaps is Donald Trump, who had filed for bankruptcy and avoided paying personal federal income taxes for almost 20 years.

The passage of Bankruptcy Abuse Prevention and Consumer Protection Act brought so many changes to the existing law so that it reshaped the landscape of U.S. bankruptcy law and redefined how the bankruptcy decision was made. With the help of advanced machine learning algorithms and big data technology, economic researchers now can closely examine and quantitatively test the effects of the BAPCPA act at both the macroscopic state level and microscopic individual level. Bankruptcy courts are state institutions. The northern Georgia district bankruptcy court currently has more than 4,000 cases each month, half of which are chapter 7 filings. A total of 231,748 cases were filed through this court between 2003 and 2008. Those petition forms are generally found as plain scanned PDF files and are very difficult to parse. In this paper, we show the magic of MLN and other information extracting techniques in unveiling the hidden data underneath those view-only PDF forms. Finally, we automatically processed and parsed over 5,000 individual petitioners' bankruptcy fillings forms that are filed in the Northern District Georgia bankruptcy court between 2003 and 2008, and studied the effects of BAPCPA before and after 2005.

3. TABLES, FIGURES AND EQUATIONS

An individual Bankruptcy petition forms process a large amount of debtor’s personal information, including address, marital status, real estate value and mortgage, personal assets, credit card balance, various sources of incomes and expenses. A summary of selected attributes we are interested is listed in Table.1. For each attribute, we would like to detect the corresponding value(s) from the extracted unstructured text generated by iText library. Sample PDFs and the Java program we developed can be downloaded from Github [3].

There are a number of challenges for inferring attributing values. The extracted text are unstructured, or semi-structured at best. First of all, a particular attribute may be associated with multiple values. For example, the debtor might have multiple credit card accounts and several real estate properties. Secondly, the location of values in the extracted text is unknown. There did exist some empirical rules in finding the attribute values. The total mortgage of a debtor might appear near the keyword “Total” and in the same page as keyword ``mortgage''. The amount of payroll reduction is the seventeenth element in the array containing tokens that match ``money'' regular expression on page ``monthly income''. But those rules are not always true. Just like tax return forms, the organization and structure of those PDF forms varies from different states and different years. The amount of payroll reduction could be the fifteen elements in the array since they might change the order of attributes in certain year. Thirdly, there is no simple pattern identify the types of tokens in the extracted text. For example, the type of attributes we are looking for is money. The candidate value token should follow certain pattern of money, for example, a token full of digits. However, a regular expression that represents digits could be corresponding to zip codes or telephone number.

Table 1 : Description of Attributes

Attribute Name	Description of Attribute
Secured Claim	If a creditor has a security interest (lien) on your property, then it has a secured claim. The most common types of secured claims are your mortgage and car loan.
Unsecured Claim	Un-dischargeable assets in bankruptcy, common examples are credit card debt. Medical bills, and personal loans.
Credit Card	Credit Card Credit card Debt Balance
Medical Bill	Debt balance from medical bills
Student Loan	The amount of student loans
Real Property	Estimated Value of real property housing price.
Real Equity	Real property asset value - the outstanding mortgage amount (secured debt).
Automobile	Value of the automobile vehicle
Personal Property	Amount personal property value
Monthly Income	CPI adjusted monthly income before tax deductions
Income year 1	Annual income this year
Income year 2	Annual income last year
Income year 3	Annual income two years ago
Monthly Expense	Estimated monthly expense
Rent Mortgage	Monthly mortgage payment.
Business Owning	Monthly business expense
Alimony Payments	Monthly alimony payments
Support Dependent	Monthly payments to additional dependents
Payment to Creditors	Total monthly payment to various creditors
Legal and Court Fees	Bankruptcy filing costs and lawyer fees
Marital Status	One of Single, Married or Divorced

Therefore, to overcome the above stated difficulties, we need to a model that can incorporate those empirical rules while having the freedom to tolerate the noise in the input text. A MLN[4] model naturally fits our requirements. We first express the empirical rules in the first order logic format. For example, we defined the probability that a token x is a zip code as:

$$\text{Prob}(\text{x is zipcode}) = \frac{1}{Z} e^{\sum w_i \text{Matches}(X, \text{Regex}_i)}, \tag{1}$$

where Matches are predicates that indicates if a token x matches a regular expression Regex_i, and w_i are weights. Table (2) summarized the regular expressions and the corresponding weights. For example, five-digit numeric expression like "98095" might be the value of a vehicle. But five-digit numeric expression following a two-letter state has a very high probability of being a zip code.

Table 2: regular expressions and corresponding weights for equation 1.

Regular Expression	Weight
^[0-9]+$	0.2
^\d{5}(?:[-\s]\d{4})?$	0.5
^(?:(A[KLRZ]|C[AOT]|D[CE]|FL|GA|HI |I[ADLN]|K[SY]|LA|M[ADEINOST]|N[CDEH JMVY]|O[HKR]|P[AR]|RI|S[CD]|T[NX]|UT| V[AIT]|W[AIVY])),\s d{5}(?:[-\s]\d{4})?$	0.99

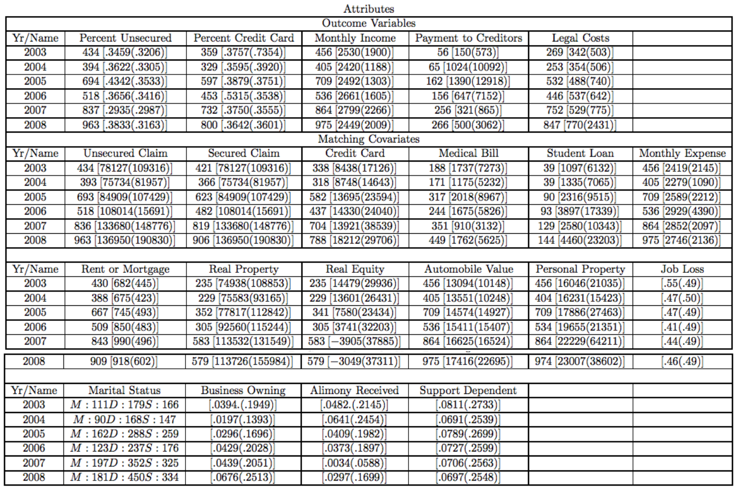
For each attribute, we could find a set of such rules that detect the attribute value. No all the rules are exact. Therefore, we assign weights to each rule and compute the probability of a token being the attribute value using standard MLN formulation. Since an attribute might correspond to multiple tokens, a hard threshold for token probability is set.

4. RESULTS

Using the MLN models, we provide economist a data set consist of 40 attributes and 3, 945 records (PDFs). The integrity of each record is enforced by several self-consistency checks. For example, the sum of various income sources (wage, alimony income, etc) should be the same as the total income. Table 3 reports the summary statistics of analysis attributes. From the data set we provided, economists found that the estimated effect of the BAPCPA on petitioner behaviors. For example, they found that an increase of $3, 178 in credit cards debts after the passing of BAPCPA. They also found that the legal cost increased substantially (by 122%, equivalently $235) after the BAPCPA was enacted, suggesting that petitioners nowadays rely more heavily on bankruptcy lawyers to help them to gain higher financial benefits.

To study the performance of MLN, we compare the recall rate of MLN approach and the rule-based approach. For the rule-based approach, we require the candidate token to follow the rules exactly. In a data set of 76 PDF files, we manually compare the reported and the true values over 16 key attributes, in order to check the precision of each approach. Under 100% precision as requested by the economist clients, the MLN approach achieves 50/76 recall rate while the rule based approach scores 42/76. However, the MLN suffers the speed issue since the computation time spent on each token is much longer in the MLN model. Moreover, the recall rate is also related to the amount of time consumed in manually finding and designing those rules.

Table 3: Summary Statistics of Analysis Attributes. Number of Observations, [Mean (SD)]



Attributes

Outcome Variables						
Yr/Name	Percent Unsecured	Percent Credit Card	Monthly Income	Payment to Creditors	Legal Costs	
2003	434 [.3459(.3206)]	359 [.3757(.7354)]	456 [2530(1900)]	56 [150(573)]	269 [342(503)]	
2004	394 [.3622(.3305)]	329 [.3595(.3920)]	405 [2420(1188)]	65 [1024(10092)]	253 [354(506)]	
2005	694 [.4342(.3533)]	597 [.3879(.3751)]	709 [2492(1303)]	162 [1390(12918)]	532 [488(740)]	
2006	518 [.3656(.3416)]	453 [.5315(.3538)]	536 [2661(1605)]	156 [647(7152)]	446 [537(642)]	
2007	837 [.2935(.2987)]	732 [.3750(.3555)]	864 [2799(2266)]	256 [321(865)]	752 [529(775)]	
2008	963 [.3833(.3163)]	800 [.3642(.3601)]	975 [2449(2009)]	266 [500(3062)]	847 [770(2431)]	
Matching Covariates						
Yr/Name	Unsecured Claim	Secured Claim	Credit Card	Medical Bill	Student Loan	Monthly Expense
2003	434 [78127(109316)]	421 [78127(109316)]	338 [8438(17126)]	188 [1737(7273)]	39 [1097(6132)]	456 [2419(2145)]
2004	393 [75734(81957)]	366 [75734(81957)]	318 [8748(14643)]	171 [1175(5232)]	39 [1335(7065)]	405 [2279(1090)]
2005	693 [84909(107429)]	623 [84909(107429)]	582 [13695(23594)]	317 [2018(8967)]	90 [2316(9515)]	709 [2589(2212)]
2006	518 [108014(15691)]	482 [108014(15691)]	437 [14330(24040)]	244 [1675(5826)]	93 [3897(17339)]	536 [2929(4390)]
2007	836 [133680(148776)]	819 [133680(148776)]	704 [13921(38539)]	351 [910(3132)]	129 [2580(10343)]	864 [2852(2097)]
2008	963 [136950(190830)]	906 [136950(190830)]	788 [18212(29706)]	449 [1762(5625)]	144 [4460(23203)]	975 [2746(2136)]
Yr/Name	Rent or Mortgage	Real Property	Real Equity	Automobile Value	Personal Property	Job Loss
2003	430 [682(445)]	235 [74938(108853)]	235 [14479(29936)]	456 [13094(10148)]	456 [16046(21035)]	[.55(.49)]
2004	388 [675(423)]	229 [75583(93165)]	229 [13601(26431)]	405 [13551(10248)]	404 [16231(15423)]	[.47(.50)]
2005	667 [745(493)]	352 [77817(112842)]	341 [7580(23434)]	709 [14574(14927)]	709 [17886(27463)]	[.47(.49)]
2006	509 [850(483)]	305 [92560(115244)]	305 [3741(32203)]	536 [15411(15407)]	534 [19655(21351)]	[.41(.49)]
2007	843 [990(496)]	583 [113532(131549)]	583 [−3905(37885)]	864 [16625(16524)]	864 [22229(64211)]	[.44(.49)]
2008	909 [918(602)]	579 [113726(155984)]	579 [−3049(37311)]	975 [17416(22695)]	974 [23007(38602)]	[.46(.49)]
Yr/Name	Marital Status	Business Owning	Alimony Received	Support Dependent		
2003	$M : 111D : 179S : 166$	[.0394.(.1949)]	[.0482.(.2145)]	[.0811(.2733)]		
2004	$M : 90D : 168S : 147$	[.0197(.1393)]	[.0641(.2454)]	[.0691(.2539)]		
2005	$M : 162D : 288S : 259$	[.0296(.1696)]	[.0409(.1982)]	[.0789(.2699)]		
2006	$M : 123D : 237S : 176$	[.0429(.2028)]	[.0373(.1897)]	[.0727(.2599)]		
2007	$M : 197D : 352S : 325$	[.0439(.2051)]	[.0034(.0588)]	[.0706(.2563)]		
2008	$M : 181D : 450S : 334$	[.0676(.2513)]	[.0297(.1699)]	[.0697(.2548)]		

5. RESULTS

The economists have hired an undergrad to manual produce the training data which consists of all the true attribute values and about 1, 000 forms. From the training data, we will learn the rules and the associated weights for each attribute using ILP and Alchemy. Furthermore, we will be able study the ROC curves for the MLN approach and the rule based approach more systematically. To our knowledge, it is the first work on extracting useful data from PDF forms. And we demonstrate the power of machine learning techniques on knowledge discovery from unstructured or semi-structured text.

REFERENCES

[1] Lowagie, Bruno "iText in Action (2nd ed.)". Manning Publications. Lowagie, Bruno, Summer 2010.

[2] Apache Foundation "The Apache PDFBox Open Source Project on Open Hub". openhub.net. Retrieved 2016-03-21

[3] Github Repository https://github.com/summerdays/deep Parser.

[4] Richardson, Matthew; Domingos, Pedro "Markov Logic Networks". Machine Learning. 62 (1-2): 107–136. 2016

AUTHOR PROFILES:

Ekin Ekinci i Nan Wang Nan Wang is a data scientist at LinkedIn. She current focuses on growth, lifecycle, and engagement, involving understanding user behavior in communications, re-engaging and elevating less active members, and discovering user intents using big data tools. Prior to this, she improved hierarchical product category structure for billions of products through data mining techniques at Amazon. Nan is also passionate about data science education. She is the authority of the best-selling book "data science interviews exposed", which has helped thousands of readers find their jobs in data science fields.