RC24083 (W0604-079) April 19, 2006
Computer Science

IBM Research Report

An Automatic Method to Extract Data from an Electronic Contract Composed of a Number of Documents in PDF Format

Thomas Y. Kwok, Thao Nguyen
IBM Research Division
Thomas J. Watson Research Center
P.O. Box 704
Yorktown Heights, NY 10598

Research Division
Almaden - Austin - Beijing - Haifa - India - T. J. Watson - Tokyo - Zurich

An Automatic Method to Extract Data from an Electronic Contract Composed of a Number of Documents in PDF Format

Thomas Kwok and Thao Nguyen
IBM Research Division
Thomas J. Watson Research Center
19 Skyline Drive, Hawthorne, NY 10532
kwok@us.ibm.com

Abstract

An electronic contract can encompass a large number of collateral contract documents in PDF format. These contract documents are of different contract document types and converted from different original formats. Data extraction and thus data mining for this kind of electronic contracts is very difficult. In this paper, we present a novel method to automatically extract contract data from this kind of electronic contracts. Our automatic electronic contract data extraction system comprises an administrator module, a PDF parser, a pattern recognition engine and a contract data extraction engine. The administrator module provides templates for inputting document patterns and a list of contract data tags for each contract document type. It also constructs the pattern matrices and stores them in a database. The PDF parser converts the contract PDF document into the contract text document with the insertion of formatting bookmarks, such as a new page, paragraph or line. The pattern recognition engine determines a list of contract document types in the electronic contract by comparing and matching the patterns of all known contract document types with the pattern of the contract text document. The contract data extraction engine retrieves the corresponding list of contract data tags and then extracts contract data accordingly for each contract document type on the list. Our automatic electronic contract data extraction system has found to be very accurate, efficient and useful in extracting contract data for data mining.

1. Introduction

Today, most business between enterprises is conducted under contract. Contracts also constitute the binding relationship between a company and its customers or suppliers. Everyday, hundreds of contracts are created, executed and managed via paper-based manual processes in large enterprises. Automation of the contract lifecycle presents a substantial value creation opportunity for the enterprise. This value stems from improved productivity and security, effectively aggregated contract information, accelerated contract lifecycle processes, reduced contractual errors and risk, enabled revenue forecast and profit optimization, as well as better compliance enforcement [1]. With the advent of Internet technology and electronic commerce, there are growing research activities and implementation efforts on the electronic contract. Currently, the International Association of Contract and Commercial Managers has listed twenty commercial available software products for electronic contract management [2]. Most of the research activities reported is focused on electronic contract creation or representation language [3], negotiation [4], management [5], collaboration [6], execution [7], fulfillment [8] and enforcement [9], performance [10], digital signature [11] and data mining [12]. However, none of these studies has provided an automatic electronic data extraction solution to enable data mining for revenue forecast and profit optimization.

A single electronic contract can encompass a large number of collateral documents including master and customer agreements, supplements, addenda and the like. These various documents are of different contract document types. There can be over a hundred different types of contract document in a large company. A few examples of these contract document types are as follows: "Master Agreement", "Customer Agreement", "Term Lease Supplement", "Addendum to Term Lease Supplement", "Statement of Work for Services" and "Change Authorization for Services". Moreover, they can also be in different file formats, such as PDF, XML, Microsoft Word, Lotus WordPro. Recently, an electronic contract management system [5] can be used to automatically convert all these contract documents of different types into PDF format and then merge them together to form a single electronic contract PDF document. However, data extraction and mining on this kind of electronic contracts are still very difficult if not impossible. First, we have to find out how many contract documents are in an electronic contract composed of a number of contract documents. Second, we have to determine their contract document types. Third, we have to know what contract data to extract and from which

contract document. Fourth, we also need to find out where on the contract document are these contract data located, such as page and line numbers. There are many more different tasks to be overcome before one can implemented a data extraction and mining on this kind of electronic contracts. In this paper, we describe a novel method to extract contract data automatically and efficiently from an electronic contract composed of a number of documents in PDF format.

2. A framework of an automatic contract data extraction system

An architectural framework of an automatic contract data extraction system for an electronic contract composed of a number of documents in PDF format is shown in Figure 1. The system comprises of four individual modules: an administrator module, a PDF parser, a contract pattern recognition engine and a contract data extraction engine. The framework also illustrates the interaction between the PDF parser and an electronic contract management system. A typical Web-based electronic contract management system shown in Figure 1 has been described in a previous publication [5]. This particular system merges all related documents of an electronic contract into one PDF document. The administrator module provides templates for inputting document patterns and contract data tags for each new contract document type. It also constructs the pattern matrices and generates a list of contract data tags, and stores them in a local or remote database.

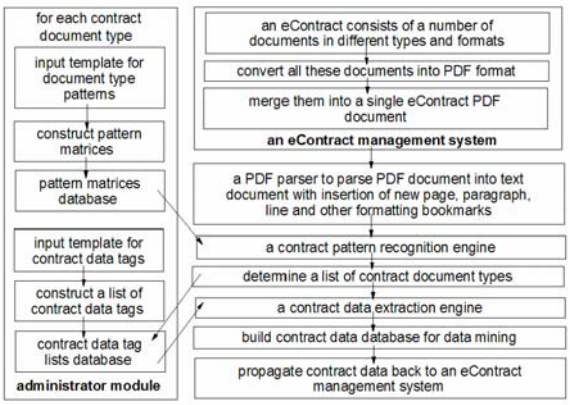

Figure 1. An Architecture framework of an automatic data extraction system for electronic contract PDF documents.

When a user uploads an electronic contract into a Web-based electronic contract management and process system, a PDF parser is used to convert the PDF document into a text document. There are a number of special functions or features in this PDF parser. First, the PDF formatting characters or words originally in the PDF document are removed and replaced with either a blank space or line feed in the text document depending on their locations in the contract document. Second, consecutive blank spaces or line feeds are reduced into a single blank space or line feed, respectively. Third, an extra text line with special coding characters or words indicating a new page or a new paragraph is inserted into the text document whenever the parser finds a new page or a new paragraph. Fourth, other special coding characters or formatting bookmarks are also used to indicate a new line, a change of fonts, an image or a picture, as well as to replace other PDF formats. These special coding characters or words are readable and decoded by both the contract pattern recognition engine and the contract data extraction engine. Then, the pattern recognition engine uses the pattern matrices of all known contract document types to determine a list of contract document types in the electronic contract by comparing and matching the pattern matrices with the pattern of the contract text document. Finally, the contract data extraction engine retrieves the corresponding list of contract data tags and then extracts contract data accordingly for each contract document type on the list. The retrieved contract data are stored on either the local or remote database for data mining. Those intrinsic metadata from the retrieved contract data are propagated back to the Web-based electronic contract management and process system to assist the user in filling up the contract input template.

3. The flow steps in the administration module

As described in Figure 2, an administrator uses the administrator module to construct document type pattern matrices and to generate a list of contract data tags for each new contract document type entering into the electronic contract management and process system. For document type patterns, the administrator first has to determine the characteristics of the beginning and ending patterns of this new contract document type. These characteristics should be special or unique to the pattern of this new contract document type and at the same time different from the pattern characteristics of other contract document types. Usually, one to two characteristics for each of the beginning or ending patterns are enough to distinguish this new contract document type from other existing or known contract document types stored in the system. Each pattern characteristic usually involves two to five lines and two to ten words or terms per line. These pattern characteristics can be a specific contract document type number written either on the first or the last page of the contract document. Characteristics of the beginning pattern are usually the first few lines of descriptions on parties involved in a contract document in the first page.

Characteristics of the ending pattern are usually the last few lines of the agreement or signing information of parties involved in a contract document in the last page. If there is no pattern characteristic which can distinct a new contract document type from other known contract document types on either the first page or the last page of this new contract document type, pattern characteristics at other pages besides the first and last pages can also be used. Other examples of pattern characteristics are special header or footer of a contract document. Then, the administrator enters the location of each pattern on the template for document type patterns. The location is a specific page number, paragraph number, line number and word number. Instead of an exact number, a range of numbers or within certain numbers can also be used. This is particularly true for the line location as the exact line number can vary and depend on the length of contract document content for different contract documents of the same contract document type. The number of lines and number of words or terms for each line involved in the pattern are also entered.

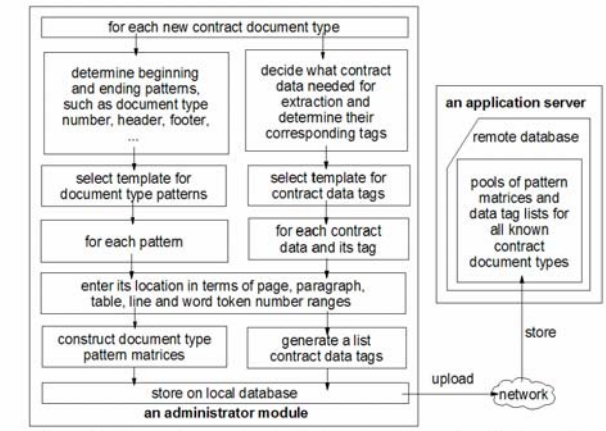

Figure 2. A Block diagram showing the flow steps in an administrator module.

For the contract data, the administrator first has to decide what contract data is needed for extraction for this new contract document type. Then, the administrator has also to determine their corresponding data tags of these contract data and enters their locations on the template for contract data tags. Again, the location is a specific page number, paragraph number, line number and word number. Usually, two to five words or terms are enough to identify a contract data tag. The number of word tokens to be extracted for a contract data corresponding to a particular contract data tag is also entered. Word tokens extracted do not necessary locate in the same line. They can be located on several consecutive lines. Finally, this administrator module will construct the document type pattern matrices and generate a list of contract data tags for this new contract document type entering the system.

4. The algorithm of a contract pattern recognition engine

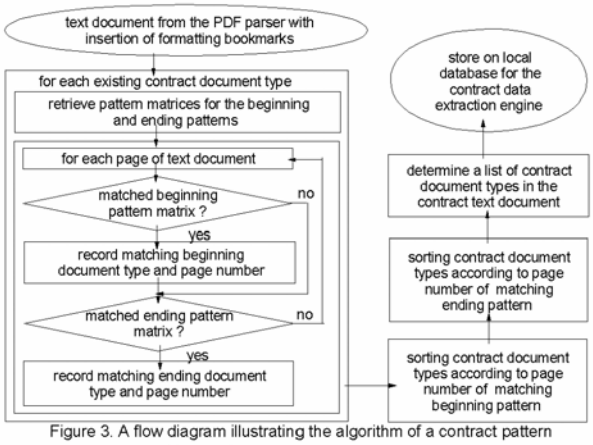

Figure 3. A flow diagram illustrating the algorithm of a contract pattern recognition engine.

Figure 3 is a flow diagram illustrates the algorithm of a contract pattern recognition engine for a contract text document. First, the engine has to read and decode those extra text lines with special coding characters or words from this contract text document generated by the PDF parser. These extra text lines are originally inserted into the contract text document by the PDF parser to indicate a new page or a new paragraph. Second, it retrieves the pattern matrices containing the beginning and ending patterns of each known or existing contract document type in the system. Third, it compares the beginning and ending patterns of each known or existing contract document type to each page pattern of this contract text document. Fourth, it records the contract document type and the page number of a particular page in the contract text document if all the characteristics of its beginning pattern matching a particular page in the contract text document. Fifth, it also records the contract document type and the page number of a particular page in the contract text document if all the characteristics of its ending pattern matching a particular page in the contract text document. Sixth, the matching page numbers along with the matching contract document types are sorted for both the beginning and ending patterns. Seventh, based on this information, the engine can then determine a list of contract document types in the contract text document. The page numbers on the contract text document corresponding to the beginning and ending pages of those contract document types in the list are also determined. Eighth, this information is stored on the local database for the contract data extraction engine to use.

Pattern matrices are used to compare the pattern characteristics of all the existing or known contract document types with the contract text document of an

electronic contract. Let a pattern matrix of a known contract document type be **q**, the entries in **q** are simply the occurrence of a list of terms or words in different lines. The occurrence of terms has to be in a specific order according to the term list, and the same term can appear in the term list more than one time. Occurrence is set to 1 while non occurrence is set to 0. . As mentioned in previous sections, each pattern characteristic usually involves two to five lines and two to ten terms per line. If there are 5 lines and each line consists of a list of 10 terms in a specific order, the maximum dimensions of a pattern matrix is 50 x 5 assuming that no more than one line has the same terms arranged in the same order in the term list. A matrix **r** of the same dimension as matrix **q** is formed from the contract text document. Similarly, the entries in **r** are the occurrence of the same list of terms in different lines. Different matrix **r** can be formed by shifting the first starting line down a particular page of the contract text document. The number of these different matrices **r** in this particular page is the number of lines in the page minus the number of lines in the pattern, such as 5 in this case. Let P be the dot product of matrix **q** and **r** divided by the dot product of matrix **q** and **q** according to Equation (1)

$$P = (\mathbf{q} \cdot \mathbf{r}) / (\mathbf{q} \cdot \mathbf{q}) \quad (1)$$

As a result, if P equals to 1 for any matrix **r** in a particular page, then this particular page matches the pattern characteristics of a known contract document type. Thus, the particular contract text document consists of this known contract document type. If the pattern matrix **q** is from the beginning pattern in the first page, then this particular page is the first page of a known contract document type. Similarly, if the pattern matrix **q** is from the ending pattern in the last page, then this particular page is the last page of a known contract document type. Other pages in the known contract document type can also be identified in a similar way.

5. The algorithm of a contract data extraction engine

For the contract text document generated by the PDF parser when an electronic contract composed of a number of documents in PDF format entering the system, the contract pattern recognition engine will first determine and construct a list of contract document types in this contract text document as described in the last section. Then, the contract data extraction engine will retrieve this list of contract document types from the local database as shown in Figure 4. It will also retrieve a list of contract data tags corresponding to each document type on the type list. For each contract data tag on the tag list, it will parse its location in terms of page, paragraph, table, line and word token number. Again, a range of numbers or within certain numbers can also be used instead of an exact number. The number of terms involved in this contract data tag is also parsed. Moreover, the number of word tokens to be extracted for the contract data corresponding to this particular contract data tag is also parsed.

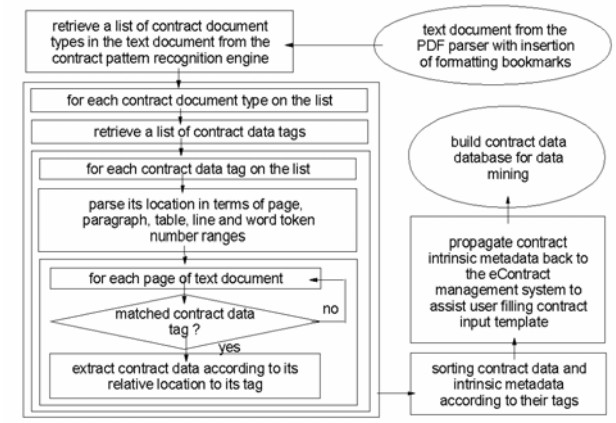

Figure 4. A flow diagram illustrating the algorithm of a contract data extraction engine.

Similar to the contract pattern recognition engine, the contract data extraction engine compares the pattern of a contract data tag to the line pattern of each page of this contract text document. Instead of using pattern matrix for the comparison as described in the last section, it uses pattern vector in a similar way for comparison. If a particular line pattern matches the pattern of a contract data tag, contract data corresponding to this particular contract data tag will be extracted according to the parsed number of word tokens to be extracted. Word tokens to be extracted need not necessarily be located along the same line. They can be located on several lines.

6. Implementation and Industrial Experience

Most of the features and functions of the automatic contract data extraction system described in this paper have been implemented and integrated with an electronic management and process system [5, 11] in IBM research division. The Adobe PDF library APIs are used to parse the PDF document into a text document in the PDF parser of this system. However, other PDF parsers from third parties can also be used.

7. Conclusion and Discussion

The contract data extraction system described in this paper provides an automated and efficient way to extract contract data for electronic contracts composed of a

number of documents in PDF format. The system has solved a number of difficult problems involving the comparison of patterns or pattern recognition. The pattern recognition technique described in this paper is based on the calculation of the dot product of pattern matrices. Thus, the system requires an administrator to input a distinct pattern of a new contract document type. This is a rather time consuming processing step for the user. However, this step is a one time setup requirement for any new contract document type coming into the system.

There are many advantages of this system. First, the number of different contract document types can vary. New contract document type can be inputted into the system at any time. Second, a list of contract data to be extracted can also be modified by modifying the corresponding list of data tags. Third, intrinsic contract metadata extracted from the electronic contract document can be propagated back in the electronic contract management system right after the user upload the electronic contract document into the system. The extracted intrinsic contract metadata can be displayed in the graphical user interface of the user module to assist the user in entering contract information of the electronic contract that he or she just submitted into the system. It has found to save users a lot of time from manually entering contract data from scratch. It also avoids wrong contract metadata mistakenly entered by the user. Fourth, this system also provides an efficient way to generate contract text content database for key word or term search.

8. Acknowledgment

The authors would also like to acknowledge the review of the manuscripts, suggestions for improvement and support from G. Pacifici and D. Dias.

9. References

[1] W.M. McGovern and L. Lawrence, Contracts and Sales: Cases and Problems, Matthew Bender, 1986.

[2] International Association of Contract and Commercial Managers, http://www.iaccm.com

[3] Y-H Tan and W. Thoen, DocLog: an electronic contract representation language, in *Proc. of 35th Annual Hawaii Int'l Conf. on System Sciences*, pages 2198-2206, *2002*.

[4] F. Griffel, M. Boger, H. Weinreich, W. Lamersdorf and M. Merz, Electronic contracting with COSMOS – how to establish, negotiate and execute electronic contracts on the Internet, in *Proc. of the 2nd Int'l Enterprise Distributed Object Computing Workshop*, pages 46-55, 1998.

[5] T. Kwok and T. Nguyen, A Secure Electronic Contract Management and Process System Automated with Predefined Tasks, in *Proc. of IEEE Int'l Conf. on e-Technology, e-Commerce and e-Service*, IEEE Computer Society, pages 276-281, 2005.

[6] O. Perrin and C. Godart, An approach to implement contracts as trusted intermediaries, in *Proc. of 1st IEEE Int'l Workshop on Electronic Contracting*, pages 71-78, 2004.

[7] M. Iwaihara, H. Jiang and Y. Kambayashi, An integrated system for supporting problem solution in e-contract execution, in *Proc. of 1st IEEE Int'l Workshop on Electronic Contracting*, pages 9-16, 2004.

[8] L. Xu, Monitorable electronic contract, in *Proc. of IEEE Int'l Conf. on E-Commerce*, pages 92-99, 2003.

[9] Z. Milosevic, A. Josang, T. Dimitrakos and M.A. Patton, Discretionary enforcement of electronic contracts, in *Proc .of 6th Int'l Enterprise Distributed Object Computing Conf.*, pages 39-50, 2002.

[10] A. Daskalopulu and T. Maibaum, Towards electronic contract performance, in *Proc .of 12th Int'l Workshop on Database and Expert Systems Applications.*, pages 771-777, 2002.

[11] T. Kwok and T. Nguyen, An Automatic Electronic Contract Document Signing System in a Secure Environment, in *Proc. of 7th IEEE Int'l Conf. on e-Commerce Technology*, IEEE Computer Society, pages 497-502, 2005.

[12] M. Castellanos and U. Dayal, FACTS: an approach to unearth legacy contracts, in *Proc. of 1st IEEE Int'l Workshop on Electronic Contracting*, pages 40-45, 2004.