Document Understanding System Using Stochastic Context-Free Grammars

John C. Handley
Xerox Corporation
800 Phillips Road, MS 128-27E
Webster, NY 14580 USA
John.Handley@xeroxlabs.com

Anoop M. Namboodiri
Indian Institute for Information Technology
Hyderabad, India
anoop@iiit.net

Richard Zanibbi
Centre for Pattern Recognition and Machine Intelligence
Concordia University
Montreal, Canada H3G 1M8
zanibbi@cenparmi.concordia.ca

Abstract

We present a document understanding system in which the arrangement of lines of text and block separators within a document are modeled by stochastic context free grammars. A grammar corresponds to a document genre; our system may be adapted to a new genre simply by replacing the input grammar. The system incorporates an optical character recognition system that outputs characters, their positions and font sizes. These features are combined to form a document representation of lines of text and separators. Lines of text are labeled as tokens using regular expression matching. The maximum likelihood parse of this stream of tokens and separators yields a functional labeling of the document lines. We describe business card and business letter applications.

1. Introduction

Document understanding is a mature and commercially viable technology. Although accuracy is far from perfect, with human inspection and correction, performance is adequate for many domains. Still, there is a need for rapid prototyping and flexible, modular systems to address diverse scanning work flow needs in a competative market. It is often too expensive to build one-off document understanding systems.

A typical work flow is to scan a document, perform OCR and identify information for further processing of the document. The aim of our system is to extract sufficient information, which we call metadata, to process the document. For example, in a business letter processing scenario, appropriate metadata might include the recipient's name and address which could be used to route the letter to the recipient. We describe a system that uses stochastic grammars to model documents – each document style or genre has its own grammar.

We built the system with three principal constraints. First, it had to be modular. By using grammars, one for each document type, we can easily and quickly get a working protoype for other business applications. Second, it had to be trainable. We wanted to avoid endless iterations of hand-crafted rules to boost accuracy. We believe that a good abstact model of the recognition process should satisfy the bulk of system requirements with minimal human intervention. Third, the system had to be robust, to produce usable results for the majority of inputs, even at the expense of high accuracy. We wished to avoid a brittle system that could fail dramatically.

A system based on stochastic grammars satisfies these goals. By modeling lines of text with a stochastic context-free grammar (SCFG), we achieved the correct balance of having a model sophisticated enough to extract useful information and simple enough to adequately estimate model parameters. For the business card and business letter document models described in this paper, we achieve training convergence with approximately 200 ground-truth exemplars, which is few enough to build by hand.

Structured documents exhibit a degree of randomness. The basic insight we used is that documents within a genre (e.g., business cards, letters, invoices, etc.) contain metadata at varying positions. The locations of metadata may posess certain regularities, but they are seldom deterministic.

Using stochastic grammars for document recogntion is not new. To our knowledge, the first appearance is [4] where mathematical expressions are recognized. In that work, productions were extended to two dimensions. Terminals where bitmapped images matched to templates. No accuracy results were presented.

2. Stochastic Context Free Grammars

A stochastic grammar $\mathcal{G}$ comprises a set of T terminals $\mathcal{T} = \{a^k; k = 1, \ldots, T\}$, a set of N nonterminals $\mathcal{N} = \{N^i; i = 1, \ldots, N\}$, a designated *start symbol* $S \in \mathcal{N}$, and a set of productions $\{A \rightarrow \omega^j\}$ where ω^j is a sequence of terminals and nonterminals, and a probability measure $P(\cdot)$ such that

$$\sum_j P(A \rightarrow \omega^j) = 1$$

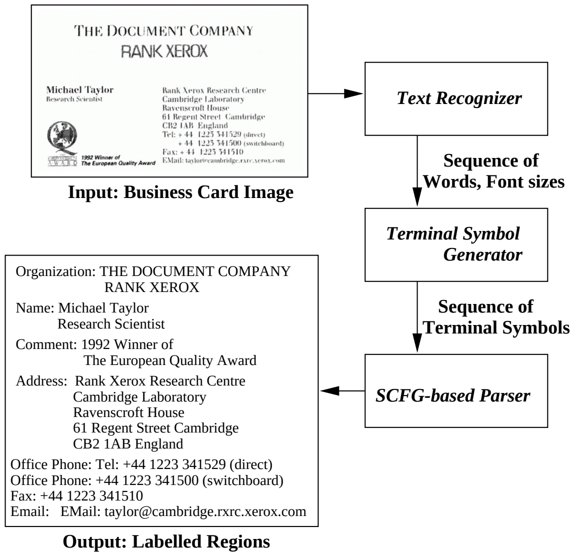
for each production rule. Terminals are traditionally denoted by lower case roman characters (a, b, etc.), nonterminals by upper case roman characters (A,B, etc.), and sequences (or *sentences*) from $\mathcal{V} = \mathcal{T} \cup \mathcal{N}$ by lower case Greek characters (α, ω, etc.). A *parse* of a sequence ω is the determination of a sequence of productions from the start symbol S to ω. The language $L(\mathcal{G})$ generated from $\mathcal{G}$ is the (possibly infinite) set of all sequences that can be generated from S using $\mathcal{G}$. If there exists a sequence ω that may be parsed mutliple ways by $\mathcal{G}$, $\mathcal{G}$ is called an *ambiguous* grammar.

In the stochastic setting, we seek the most probable parse. Fortunately this can be done relatively efficiently using a well-known variation of the Cocke-Young-Kasami (CYK) algorithm [10]. Probabilities can be estimated using the Inside/Outside (I/O) algorithm, a special case of expectation-maximization [10]. For the CYK algorithm parsing algorithm, a grammar must be expressed in Chomsky normal form. In a context-free grammar, all production rules are written as $A \rightarrow a$ or $A \rightarrow BC$ in Chomsky normal form.

The Chomsky hierarchy of grammars is a nested set of grammars with increasing restrictions. Type 0 or unrestricted grammars have productions of the form $\alpha A \beta \rightarrow \gamma$. These grammars are too general to be parsed efficiently. Type 1 or context-sensitive grammars have productions of the form $\alpha A \beta \rightarrow \alpha \gamma \beta$ or $A \rightarrow \alpha$. Parsing sequences in context-senstive grammars is NP-complete. Type 2 or context-free grammars are sufficiently expressive for out purposes, and can be parsed in $O(n^3)$ time, where n is the length of the sequence. Type 3 or regular grammars can be parsed efficiently in $O(n)$ time but express too few sequences for our needs.

3. System Overview

The basic idea of our system is illustrated in Figure 1. A scanned business document is converted to a list of words with attributes. The word list is converted to a sequence of tokens representing a linearization of text blocks found in a document; tokens represent text line types and text block separators. The token sequence is then parsed using a SCFG. The most probable parse is used to assign metadata types to text blocks. Detected metadata fields can then be used to route the document or store in correctly in a document database.


Input: Business Card Image
Text Recognizer
Sequence of Words, Font sizes
Terminal Symbol Generator
Sequence of Terminal Symbols
SCFG-based Parser
Output: Labelled Regions

Organization: THE DOCUMENT COMPANY
RANK XEROX
Name: Michael Taylor
Research Scientist
Comment: 1992 Winner of
The European Quality Award
Address: Rank Xerox Research Centre
Cambridge Laboratory
Ravenscroft House
61 Regent Street Cambridge
CB2 1AB England
Office Phone: Tel: +44 1223 341529 (direct)
Office Phone: +44 1223 341500 (switchboard)
Fax: +44 1223 341510
Email: EMail: taylor@cambridge.rxrc.xerox.com

Figure 1. Business card labeling.

Figure 2 shows a more complete system description. From a scanned bitmap, an OCR system recognizes lines of text and their positions. The OCR system also provides font size estimates and indicates whether the text is bold, italic or underlined. These attributes are used later in tokenization. A variant of Nagy and Seth's XY-cut algorithm [11] orders the lines into blocks of text. Separators are infered between blocks and lines of text and lines with their attributes and separators are listed in top down, left to right order. Generalized regular expression matching is used to tokenize text lines. The exact tokens are document-genre specific, but some examples for business cards includes email addresses, urls, and fax numbers. Some tokens are generic such as alphabetic lines. A street address, for example, is modeled simply as an alphanumeric line. A person's name is also an alphabetic line.

We use simple regular expressions to label text lines instead of maintaining a vast database of street names, cities, names, etc. The lack of external databases makes our system simple to adapt to other document genres and robust to recognition errors. A single file containing regular expressions and a stochastic grammar defines the document model (similar to a *.y file in yacc).

Font attributes detected by the OCR system are used to map alphabetic lines to special tokens such as large or huge lines – these help distinguish organization and personal names in business cards, for example.

A sequence of tokens includes tokenized text lines and separators. This sequence is processed by a module that reads the grammar file and computes the most likely parse. Text blocks are labelled within the parse tree by parent non-terminal nodes representing metadata types.

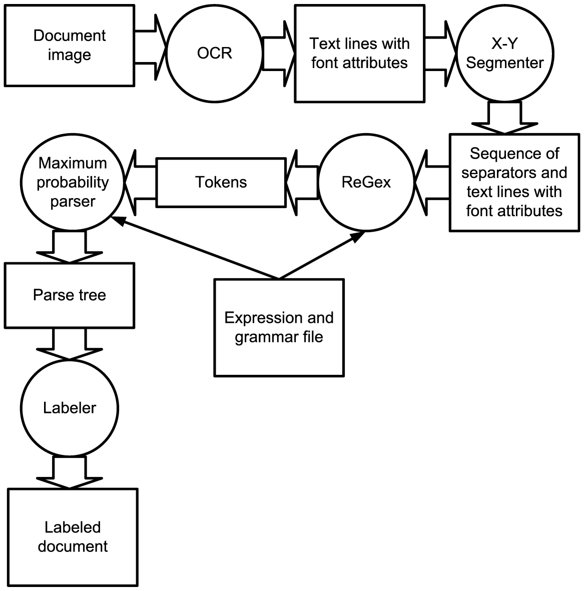

Document image
OCR
Text lines with font attributes
X-Y Segmenter
Maximum probability parser
Tokens
ReGex
Sequence of separators and text lines with font attributes
Parse tree
Expression and grammar file
Labeler
Labeled document

Figure 2. System flowchart.

4. Business Cards

The most common approach to business card recognition is to use a text recognizer to convert the scanned images of the cards into plain text and then employ a dictionary of names for people, titles, organizations and places to label the contents of a card. Such approaches usually discard the structural information present in the business cards.

Most reported research on business card recognition deals with the problem of text recognition in the context of business cards. Kise et al., [6] reported one of the first business card recognition systems that attempts the automatic conversion of Japanese business cards. They use a rule-based system on a tree representation of the structure of business cards to reduce the search time and improve the accuracy of recognition. Saiga et al., [13] attempt the recognition of Japanese business cards using a dictionary-based approach. Watenabe and Huang [15] describes a rule-based system using the logical relationship between blocks of text to help the identification of the contents of Chinese business cards. Pan and Wang [12] use a rule-based system for structure analysis and extraction of text from Chinese and Chinese-English business cards. However, they do not utilize the structure information to assist the recognition process. Problems related to text recognition, such as text segmentation and orientation detection in business cards have also been addressed. Chiou and Lee [3] deals with the problem of separation of text from multi-colored Chinese business cards. Watenabe and Huang [15] also deals with the problem of orientation detection of text blocks in Japanese business cards.

The first step in designing a grammar is to define a set of terminal symbols for text line types which could appear in a business card. Each terminal in the grammar represents a single line of text from the card. Lines are labeled according to their contents as one of the following terminals: alphabetic line (**a_line**), alpha-numeric line (**an_line**), large-font or bold lines (**emph_line**), huge font-line (**huge_line**), line with office/main phone number (**office_line**), line with fax number (**fax_line**), line with mobile phone number (**mobile_line**), line with pager number (**pager_line**), line with any other phone number (**other_line**), line with an email address (**email**) and line with a url (**url**).

In addition to the above terminals, the terminal **separator** is used between any two lines if they are separated by a significant vertical or horizontal gap, as detected by the XY-cut algorithm.

Non-terminals of the grammar are selected to represent different regions to be labeled in a business card. These include: name and title of the person (NAME, AFFILIATION, ID_BLOCK), name of the organization (ORG_NAME, ORG_BLOCK), address of the person/office (ADDRESS_BLOCK), phone numbers (PHONE_BLOCK), email address and URL (INTERNET_BLOCK) and a set of lines which does not fit any of the above labels (COMMENT_BLOCK). Other non-terminals such as PHONE_LINE, ADDRESS_NAME, A_LINES, etc. are included to abstract multiple terminals into one group.

Every business card is assumed to contain a single ID_BLOCK and a single ADDRESS_BLOCK. Any other block of contents of the card could occupy one of three different positions, with respect to the ID_BLOCK and the ADDRESS_BLOCK. Those blocks, which occur prior to both ID_BLOCK and ADDRESS_BLOCK are grouped together as S_BLOCKS (start blocks), those which occur in between the two are called M_BLOCKS (middle blocks) and those which occur after the two are called E_BLOCKS (end blocks). The grammar models the probability of the labels to belong to S_BLOCKS, M_BLOCKS or E_BLOCKS. It also considers the two possible orderings of ID_BLOCK and ADDRESS_BLOCK. Terminal symbols are named using lower case alphabets and non-terminal symbols, using upper case. The number preceding each production denotes the probability of the production being applied in the derivation of a terminal string when the non-terminal on the left hand side is encountered.

Grammar production rule probabilities, the parameters in our model, are learned from hand-labeled examples using the Inside/Outside algorithm adapted for SCFGs. The size of the training set depends on the complexity of the grammar. To determine the required size, we trained the grammar using random initializations over sets of increasing size. When the set is large enough, all estimates converge to the same values. We found that a training data size of about 150 was sufficient. The grammar used for business card labeling contained 164 rules, when expressed in the Chomsky Normal Form.

The OCR system estimates font sizes of each character. Classification of font sizes as terminal symbols *a_line*, *emph_line* and *huge_line* is done by assigning each a Gaussian probability distribution.

To cope with the variability in font sizes across different cards, we normalize font sizes within each card to the range $[0, 1]$.

To incorporate these probabilities into the CYK algorithm, (or while training the grammar using the I/O algorithm), the initialization step is modified. The probability of a nonterminal deriving a particular terminal symbol of a specific font-size is the product of the probabilities of the non-terminal deriving the terminal symbol and that of the terminal symbol having the font-size observed: $P(NT \rightarrow T_{fs}) = P(NT \rightarrow T)P(fs|T)$ where T_{fs} is a terminal symbol of font-size fs and NT is a non-terminal. In the case of the CYK algorithm, we compute the maximum probability of $P(NT \rightarrow T_{fs})$ over all T and in the case of I/O algorithm, we find the sum of the probabilities over all Ts.

The data set for training and testing the stochastic grammar-based business card recognition system consisted of 180 business cards. The data were randomly divided into two sets: 120 card for training and 60 cards for testing. A labeling of a business card is considered erroneous if any of the region labels assigned by the system is wrong. The accuracy of labeling on three-fold cross validation (with 60 cards per partition) was 83.5% with the trained grammar.

5. Business Letters

For English business letters, we wish to classify text blocks according to content type, including a letter's date, opening ("Dear Ramkumar,"), closing ("Sincerely,"), body text, and recipient address. Content types of interest vary by application: automatic mail distribution [2, 5] may require a different set than indexing letters in a database [9] or automatically generating responses to a class of letters [1].

Previously researchers have assigned metadata types to text blocks using rule-based techniques [2, 5, 7, 9], registering text blocks to layout models (e.g. of letterhead for organizations [2]), and approximate graph matching [8, 14]. The body of a letter is usually sandwiched between the opening and closing of a letter, and most strategies take advantage of this fact. Features used for classification include text block geometry and text content. Regular expression matching is commonly used to analyze text. [1, 2, 7, 8, 9], More sophisticated parsing techniques have also been employed [5]. OCR errors have been accommodated by string distances, equivalent dictionary-based techniques [2, 7, 9], and fuzzy rules [7].

For our system, we use another stochastic grammar to model the metadata types of a linearized sequence of text blocks. The OCR system eliminates graphic areas, and this sometimes includes text intersected by signatures.

We defined our stochastic grammar model primarily by observing samples, using writing guides for naming metadata types.

Our business letter grammar contains six text line token types. As indicated earlier, text line types are defined using regular expressions. The six text line types in the business letter grammar are: Date, Open/Close ("Dear Ramkumar","Sincerely"), Contact information (addresses, phone numbers, etc.), Name, Tag line (lines with prefix tags, e.g. P.S., To:, Enclosure) and Other (default text line type) As in the business card grammar, an additional **separator** token type is also defined, to represent significant horizontal and vertical gaps between text blocks, as detected by the X-Y cut algorithm.

The following business letter region classes (metadata types) were used in the grammar: Dateline, Signor (Signor's identification), Inside Address, Letterhead (Name, Titles of sender), Opening, Letterhead contact information (contact information of sending organization), Body text, Closing Tag line (this includes all labelled regions, e.g. To:, P.S., Enclosure, cc:, identifcation line, etc.) and Other regions (anything else)

We separate letterhead regions which simply name or describe an organization in text from letterhead 'contact' regions, which contain contact information for the organization of the sender. The text content of these regions are quite different, and so it made sense to treat these as separate classes.

Only body text regions were always present in our study set. All other regions types were absent in one more examples, due to elimination by OCR preprocessing (e.g. closing regions) or simply variation in letter formats.

In our study set, the linearized ordering of regions in the training data had a great deal of variation both before and after the body text of a business letter. To accomodate this, the grammar describes a model where a non-empty set of regions are at the top of the letter, followed by body text regions, and then a non-empty set of regions at the bottom of the letter. The regions that can appear above the body text and below differ, in that the dateline, opening and inside address of a letter always precede body text, while the closing and signor identification always follow body text.

As with business cards, training the probabilities was done using the I/O algorithm. We used a number of random initializations and then inspected the grammars after training, to insure the grammars were converging. For the initial grammar five random intializations were used, which converged roughly to the same rule probabilities.

The grammars were trained on a set of 169 ground truth files constructed from the OCR output for the training data. These files are in the same format as the region summary files produced from the CYK parse results, where region types are used to label the text lines in the OCR text output.

6. Conclusion

We have presented a modular, extensible and robust document understanding system for high-value business documents. In this system, each document genre has its own stochastic grammar and regular expression tokenizer. Models are maintained in a single file that can be adapted to new document types as business needs require. Information extracted from documents in this model are critical to business processes: names, phone numbers, email addresses, etc. The key insight in the system design is that lines of text within many classes of documents can be modeled using syntactic pattern recognition. We have demonstrated the efficacy of this approach though two applications: business cards and business letters.

References

[1] T. A. Bayer and H. Walischewski. Experiments on extracting structural information from paper documents using syntactic pattern analysis. In *Proceedings Third International Conference on Document Analysis and Recognition*, pages 476–478, 1995.

[2] T. Brückner, P. Suda, H. U. Block, and G. Maderlechner. Inhouse mail distribution by automatic address and content interpretation. In *Proceedings Fourth Annual Symposium on Document Analysis and Information Retrieval*, pages 67–75, 1996.

[3] Y.-H. Chiou and H.-J. Lee. Recognition of Chinese business cards. In *Proceedings of the* 4^{th} *International Conference Document Analysis and Recognition*, pages 1028–1032, Ulm, Germany, Aug. 1997.

[4] P. A. Chou. Recognition of equations using a two-dimensional stochastic context-free grammar. In *Proceedings of the SPIE Visual Communications and Image Processing IV*, volume 1199, pages 852–863, Philadelphia, PA, November 1989.

[5] A. Dengel, R. Bleisinger, F. Fein, R. Hoch, F. Hönes, and M. Malburg. Officemaid – A system for office mail analysis, interpretation and delivery. In A. L. Spitz and A. Dengel, editors, *International Association for Pattern Recognition Workshop on Document Analysis Systems*, pages 53–75, 1995.

[6] K. Kise, K. Yamada, N. Tanaka, N. Babaguchi, and Y. Tezuka. Visiting card understanding system. In *Proceedings of the* 9^{th} *International Conference on Pattern Recognition*, pages 425–429, Rome, Italy, Nov. 1988.

[7] S. Klink, A. Dengel, and T. Kieninger. Document structure analysis based on layout information and textual features. In *International Association for Pattern Recognition Workshop on Document Analysis Systems*, pages 99–111, 2000.

[8] J. Liang and D. Doermann. Content features for logical document labeling. In T. Kanungo, E. Barney-Smith, J. Hu, and P. B. Kantor, editors, *Proceedings of SPIE-IS&T Electronic Imaging Vol. 5010*, pages 189–196, 2003.

[9] M. Lipshutz and S. Taylor Liebowitz. Functional decomposition of business letters. In *Proceedings Fourth Annual Symposium on Document Analysis and Information Retrieval*, pages 435–447, 1995.

[10] C. D. Manning and H. Schütze. *Foundations of Natural Language Processing*. The MIT Press, Cambridge, Mass, 1999.

[11] G. Nagy and S. Seth. Hierarchical representation of optically scanned documents. In *Proc. Int'l Conf. Pattern Recognition*, pages 437–349, 1984.

[12] W. Pan, J. Jin, G. Shi, and Q. Wang. A system for automatic Chinese business card recognition. In *Proceedings of the* 6^{th} *International Conference Document Analysis and Recognition*, pages 577–581, Seattle, WA, Sept. 2001.

[13] H. Saiga, Y. Nakamura, Y. Kitamura, and T. Morita. An OCR system for business cards. In *Proceedings of the* 2^{nd} *International Conference Document Analysis and Recognition*, pages 802–805, Tsukuba City, Japan, Oct. 1993.

[14] H. Walischewski. Automatic knowledge acquisition for spatial document interpretation. In *Proceedings Fourth International Conference on Document Analysis and Recognition*, pages 243–247, 1997.

[15] T. Watanabe and X. Huang. Automatic acquisition of layout knowledge for understanding business cards. In *Proceedings of the* 4^{th} *International Conference Document Analysis and Recognition*, pages 216–220, Ulm, Germany, Aug. 1997.