Metadata Extraction from Office Documents

William K. Stumbo and John C. Handley
Xerox Corporation
Webster, New York, USA

Abstract

This paper focuses on using layout-based techniques to automatically extract metadata when scanning office documents to an archive. Many office documents such as letters, inter-office memos, and invoices contain key information that is spatially arranged. Information arrayed in this manner is easy for a reader to identify and understand. However, location of information within office documents varies greatly between documents, unlike forms where layout is static. This poses a challenge for layout based metadata extraction techniques. Our system uses regular expression matching and stochastic grammars on lines of text to efficiently and accurately label text according to function, enabling archived documents to be precisely retrieved.

Introduction

Office documents fall into three rough categories, structured, semi-structured and unstructured. Structured documents have well-defined, consistent layouts and include invoices and other forms. Information extraction from these documents is often handled by template matching. That is, matching a form to the document image and selecting fields that occur in given positions. A set of templates can be matched against a number of document sources and when a template match fails, a new template is constructed and added to the set.

Unstructured documents have no layout clues for text functionality. Typical documents include reports and memos in ASCII and raw OCR results from scanned documents in TIFF or PDF. Linguistic methods are appropriate here.[1]

Semi-structured documents such as business letters, have a well-defined style, but placement of key information varies from document to document. Structure for us refers to spatial or geometric patterns. Over the past several years much emphasis has been placed on metadata extraction techniques from semi-structured documents which are marked-up electronic documents on the World Wide Web. Techniques to extract pertinent information from them to enable effective searching are an area of active research.[2,3]

In our work, we are focused on printed documents that are found in office environments. The ability to electronically archive them and, at a future time, quickly find them is a key enabler of useful electronic storage systems. Documents scanned in an office environment are usually stored as document image files. Text may be extracted via OCR, but much of the information in an office document is location sensitive. For instance, in a business letter, the recipient is identified by position of the name and address. Such information we term *metadata* because it refers to data relevant to the system and user operating on the document – data that transcends the document. In addition to text for full text search, office documents require metadata to accurately describe the document and its intended purpose.

Metadata extraction from a document image is a two step process: document structure analysis and object labeling. The scanned document is first analyzed as an image, converted to a sequence of text lines using segmentation and OCR. Next, the lines of text containing metadata are extracted using a document model.

A primary consideration of our work is speed of conversion to an archive. Metadata identification and extraction must not impose a significant overhead on the complete document archive process. It is our opinion that the time and effort to enter metadata are the two biggest barriers to collection of useful metadata on scanned documents. Casual observation of our coworkers storing documents into a repository corroborates this hypothesis. Documents are often stored with only the required metadata. Our aim is to develop a system that recognizes a few useful fields well rather than an elaborate system that often fails.

Background

Existing techniques for metadata extraction from semi-structured documents try to infer the function of text from position, font information, co-occurring text, etc.

Layout-based metadata extraction focuses on defining a set of models that candidate documents are mapped against. Nagy et al. describe an implementation of a layout based document structure analysis using a grammar to describe the layout of a document.[4] Their work focuses on journals with a well-specified layout. The layout can be encoded as a block grammar. The document being analyzed is segmented into a series of blocks, represented as a tree, and each block is assigned a label by parsing the tree.

Liang and Doermann propose a model-based system that uses both textual and layout information.[5] They create a graph consisting of nodes representing the components of a business letter, and a set of arcs between nodes, representing relationships observed within a training set (e.g., Date is followed by Inside Address). Each arc is assigned a cost function. An input document is assigned a set of preliminary labels by analysis of its physical structure and the textual

content. The graph model is then searched for the lowest cost match that describes the input document.

Approach

In this paper, we describe a system that uses the layout analysis of Nagy augmented with semantic information and the application of a stochastic parsing model. This approach provides a great deal of flexibility when working with semi-structured documents and allows us to use the same layout analysis model and stochastic parser while allowing different grammars to be plugged into the system. While our approach is similar to Liang and Doermann in that we both use layout and textual information, we believe the application of stochastic grammars is a simpler and more robust approach.

Our technique is to create a grammar to describe the layout of a specific document genre; e.g., business letters or business cards. The tokens in the grammar represent the text lines which compose the document, the elements which describe the document layout. Examples of tokens would be: dateline, inside_address, body, etc.

The production rules describe combinations of tokens (layout elements) which represent member documents of the genre. The most probable parse of the document is inspected to uncover the production rules used and the intermediate token types assigned to text blocks during the parse. We then correlate token types with extracted information.

We have chosen to implement our system (Figure 1) using office multifunction devices as capture devices; they are becoming ubiquitous in office environments. Scanned document images are sent to a server where the document image is analyzed. Once analysis is complete and metadata is identified the document image and associated metadata is archived into a document repository. We experimented with two grammars in our system. As part of our document capture workflow, we give the user an opportunity to validate the accuracy of the extraction and correct any errors introduced by OCR or extraction.

Processing starts with a user scanning a document on an office multifunction device. The document is then deskewed. Once this operation is complete the document is ready for metadata extraction.

Layout Analysis

Metadata extraction starts with the document being processed by an OCR package to identify the text segments and encoded characters. The OCR output is further analyzed; text segments are examined for gaps which could represent a column. If a gap exceeds a parameterized value is identified, the text segment is cut into two segments. This processing continues until no additional cuts can be made. Using a variation of the XY-cut algorithm the collection of text segments is then segmented into page regions.[4]

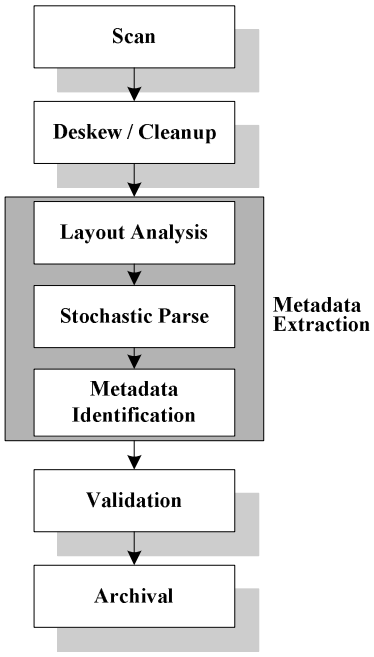

Figure 1.

Processing continues with conversion of the tree resulting from the XY-cut into a linear list by application of a depth-first traversal. The result is a top down, left to right ordering of text lines. We then classify each text segment into a category. For business letter metadata extraction we use the following categories: separator, date, open/close, contact, tagged line, name and other text. Each of these classifications is represented by a regular expression. If the regular expression is able to describe the text segment, it is decorated with the matching label.

At this point the layout of the document is well described and could be mapped against a static, structured representation, e.g., a form or the prescribed format for a journal article. However, our target documents contain too much variation and can not reliably map into a static representation. The application of a stochastic parser allows us to account for this variation.

Stochastic Layout Parsing Model

We continue processing by applying a probabilistic classifier against the layout representation of the document. Document layout is described using a stochastic context free grammar. Position of a text block, font information and text classification are used to define the production rules of the grammar. Each transition in the grammar is then decorated with a probability, the likelihood the step is correct based on a training set or observation. Document parsing consists of applying the grammar to the output of layout analysis. A stochastic grammar may contain multiple paths that lead to a successful parse of the input document. The probability of each complete path (derivation) is calculated as the product of the probability assigned to each step:[6]

$$P(t_1 \circ \ldots \circ t_n) = \prod_i P(t_i) \quad (1)$$

The parse path with the highest resulting probability, the most probable parse, is chosen as the correct parse of the input document. This can be done efficiently using a well-known dynamic programming method.[6]

Metadata Extraction

The final step in processing the document is to extract the metadata. With parsing complete, we have categorized and assigned a meaning to each of the text blocks that compose the input document. Metadata extraction consists of identifying a subset of the categorized text blocks that describe the purpose of the document. For business letters we extract the date, addressee, return address and signor. Our focus is on capturing and saving metadata that can be identified via layout. Semantic analysis of the body of a document would provide additional metadata, specifically keywords further describing the intent and subject of the letter. However, deep analysis of textual content is beyond the scope of our activity.

The layout based extracted metadata provides information that is not as readily assessable via full text searches. Full text searches could find all the instances of a specific date or a name with in the document repository. However, they cannot distinguish between a date of a letter and a date used within the body of the letter. The same holds true for names. A full text search could easily identify each usage of a name, but without providing information on whether the name appears as the sender, recipient, or somewhere within the body of a letter. The role of information derived from layout analysis of a document can be determined by its position within the document. This ability provides a key component of layout based metadata extraction.

Grammar Implementation

In this section we take a closer look at the parsing model and the grammars developed to represent office documents. Nagy notes that individual journals and other structured documents have a well-defined layout.[4] Application of a block grammar that fully defines the layout of a structured document will yield a successful parse. This approach works well when the documents are from a single source and formatting is consistent. Office documents do not adhere to strict formatting conventions. Documents from different clients and vendors will have different formatting conventions. However, the components of formal business communication are generally the same. What differs is their location, font selection and phrasing.

To compensate for the inter-document variations we use a stochastic parsing model. For each document genre, business letter, business card, invoice, etc., we create a grammar that describes prototypical layouts. The grammar needs to be general enough to describe multiple document formats.

We create our grammars by collecting a set of sample documents. We use this set of documents to develop an understanding of the variations in document layout and from this understanding we develop a prototype grammar.

The grammar consists of a set of terminals representing the collection of labeled text blocks identified by the layout analysis step. Secondly we create a set of nonterminal tokens which represent logical groupings of terminal and nonterminal tokens. We define a distinguished symbol that represents the complete document. We next define a set of rules, productions, that specify how to take combinations of terminals and nonterminals and reduce than to a single nonterminal or the distinguished symbol. The goal of the parser is to take the collection of terminals, consume them all and yield only the distinguished symbol. When this occurs the parse is successful. Lastly, we assign a probability to each rule representing the likelihood that the production rule is correct based off our sample data. Figure 2 illustrates a couple production rules with their associated probabilities:

[1.0] Letter → Top Body Bottom

[0.10738] Top → Dateline
[0.07347] | Inside_Address
[0.16787] | Opening
[0.07234] | Letterhead
[0.06871] | Letterhead_Contact
[0.49597] | Other_Region
[0.01426] | Tag_Line

Figure 2.

More formally, we describe a stochastic context free grammar as:[6]

A Stochastic Context Free Grammar is a 5-tuple ($\mathbf{V_N}$, $\mathbf{V_T}$, **S**, **R**, **P**) where:

- $\mathbf{V_N}$ is a finite set of nonterminal symbols
- $\mathbf{V_T}$ is a finite set of terminal symbols
- $\mathbf{S} \in V_N$ is the distinguished symbol
- **R** is a finite set of production rules of the form:

$$x \rightarrow y \quad (2)$$

  where $x \in V_N$ and $y \in (V_N \cup V_T)$.
- **P** is a function which assigns to every production rule $\tau \in R$ a probability $P(\tau)$. For a production rule τ with a left-hand side symbol $lhs(\tau) = \alpha$, $P(\tau)$ is interpreted as the probability of substituting τ on a node α. We require, therefore, that $0 < p(\tau) \leq 1$ and $\Sigma_{\tau:lhs(\tau)=\alpha} P(\tau) = 1$.

The process of parsing consumes the sequence of layout blocks from left to right. At each step of the parse, we identify the set of valid production rules and apply them. We then examine the current probability for each parse path. When a parse path goes below a parameterized threshold we terminate processing of that path. This keeps the parser from following paths that have a low probability of yielding a successful path and speeds the overall parse process.

Validation and Archival

In our system, we have a validation step between information extraction and archival in a document repository. Stochastic grammars provide a highest probability result, which sometimes provides incorrect results. In addition, OCR may incorrectly recognize portions of the text. Combinations of poorly scanned documents and handwritten annotations along with the difficulties of character recognition can all lead to errors in the extracted metadata text. These errors can result in incorrect classification of metadata during the extraction process or, more commonly, misspelled text being provided as metadata.

The validation step provides the user with an image of the scanned document and the metadata that was extracted. The user is given the opportunity to correct misspellings in the metadata fields, remove or reassign misfiled information. In addition, if the document is incorrectly scanned or some other catastrophic failure occurred the user can abort the process.

The extracted metadata can then be archived along with the document in a document repository. In our implementation we used Xerox's DocuShare as our document repository. DocuShare allows the creation of new document metadata fields enabling us to map our extracted metadata directly into the document repository's representation of the document. With other document repositories the metadata may need to be mapped to an existing value.

Results

We have developed two grammars that implement our layout based parsing model, one for describing business cards and the other for letters. Both grammars were developed by hand, tested and refined against a training set. The stochastic weighing of each transition in the parse tree was heavily influenced by the outcome of the training sessions. Based on observations and domain knowledge further refinements were made to the weights to help direct the parser to the correct solution.

The current implementation demonstrates the efficacy of our approach. Regular expressions and stochastic grammars for each document are maintained in a simple file. It is straightforward to add new document models.

Conclusion

We are currently developing grammars for new document genres and exploring how to improve the set of production rules to increase the recall rate and precision of our extraction techniques. We are encouraged by our initial results. It appears that application of stochastic layout grammars to office documents can provide a flexible and powerful tool for information extraction.

References

1. C. Cardie, "Empirical methods in information extraction," AI Magazine 18(4), pg. 65-79 (1997).
2. S. Soderland, "Learning Information Extraction Rules for Semi-Structured and Free Text", Machine Learning 34 pg 233-272. (1999).
3. S. Abiteboul, "Querying Semi-Structured Text", in Proceedings of International Conference on Database Theory, pg. 1-18. (1997).
4. G. Nagy, S. Seth and M. Viswanathan, "A Prototype Document Image Analysis System for Technical Journals", IEEE Computer 25 pg 10-21. (1992)
5. J. Lian and D. Doermann, "Content Features for Logical Document Labeling," Document Recognition and Retrieval X, pg. 189. (2003).
6. K. Lari and S. J. Young, "Applications of stochastic context-free grammars using the Inside-Outside algorithm," Computer Speech and Language 5 pg. 237-257 (1991).

Biography

William Stumbo is a member of the research staff in Xerox's Imaging and Services Technology Center in Webster, NY. During his nineteen years with Xerox he has held a variety of technical positions in product development and research. He completed his Bachelors degree in Computer Science at Purdue University in 1983 and received a Masters degree in Computer Science from Rochester Institute of Technology in 1993. E-mail: wstumbo@crt.xerox.com.