E-mail Signature Block Analysis

Hao Chen
CENPARMI, Concordia University
1455 de Maisonneuve Blvd. West
Montreal, Quebec H3G lM8
Canada
hchen @cenparmi .concordia.ca

Jianying Hu Richard W. Sproat
Lucent Technologies Bell Labs
700 Mountain Avenue
Murray Hill, NJ 07974-0636
USA
{jianhu | rws} @research.bell-labs.com

Abstract

The signature block is a common structured component found in e-mail messages. Accurate identification and analysis of signature blocks are important in many multimedia messaging and information retrieval applications such as e-mail text-to-speech rendering. It is also a very challenging task, because signature blocks often appear in complex two-dimensional layouts which are guided only by loose conventions. Traditional text analysis methods designed to deal with sequential text cannot handle 2-dimensional structures, while the highly unconstrained nature of signature blocks makes the application of 2-dimensional grammars very difficult. In this paper we describe an algorithm for signature block analysis which combines two-dimensional structural segmentation with one-dimensional grammatical constraints. The information obtained from both geometrical and linguistic analysis are integrated in the form of weighted finite state transducers (WFST), and the final solution is the optimal interpretation under both constraints.

1. Introduction

The rapidly increasing usage of the Internet in the recent years has made e-mail one of the most common forms of business and personal communication. Signature block is one of the most common structured elements in text e-mail. It contains the sender's contact information and is usually separated from the rest of the message by some sort of border. Accurate identification and parsing of signature blocks is important for many multimedia messaging applications such as e-mail text-to-speech rendering, automatic construction of address databases, and interactive message retrieval. However, parsing of signature blocks is also a very challenging task due to the fact that they often appear in complex two-dimensional layouts which are guided only by loose conventions (e.g., Fig. 1). The only way to extract functional fields from such layouts is to combine two-dimensional layout analysis with linguistic constraints. Two-dimensional grammars (array grammars) [6] and geometric trees [1] have been proposed for logical layout analysis in printed documents, but they are applicable only to known document types with rigid layout rules, which is not the case with signature blocks.

We have developed a new approach to combining two-dimensional structural analysis with one-dimensional grammatical constraints for signature block parsing. The information obtained from both geometrical and linguistic analysis are integrated in the form of weighted finite state transducers (WFST) [3], and the final solution is the optimal interpretation under both constraints.

We define the following terms used throughout the paper. A *signature block* is comprised of several continuous lines of text separated from the rest of an E-mail message. A signature block may be decomposed into *reading blocks.* Reading blocks ensure the coherence of text. Text in a reading block can be read out in a meaningful order by simply following the sequence from top to bottom, and from left to right on each line. A reading block is decomposed further into *functional blocks.* Text in each functional block belongs to the the same *functional class.* Ten functional classes are defined in the current approach (Fig. 3). Signature blocks, reading blocks, and functional blocks constitute a hierarchical text structure, as shown in Fig. 1

The algorithm is currently implemented as a component in a preprocessing system for e-mail text-to-speech rendering called *EMU.* An N-gram character class model based classifier is first applied to propose potential signature blocks, including all material near the end of the message. The potential signature block then undergoes the signature block analysis process described in this paper. If enough non-miscellaneous functional blocks are detected and the ratio of non-miscellaneous vs. miscellaneous text is large, then the block is verified as a signature block and analysis

1051-4651/98 $10.00

results are returned, otherwise it is treated as plain text.

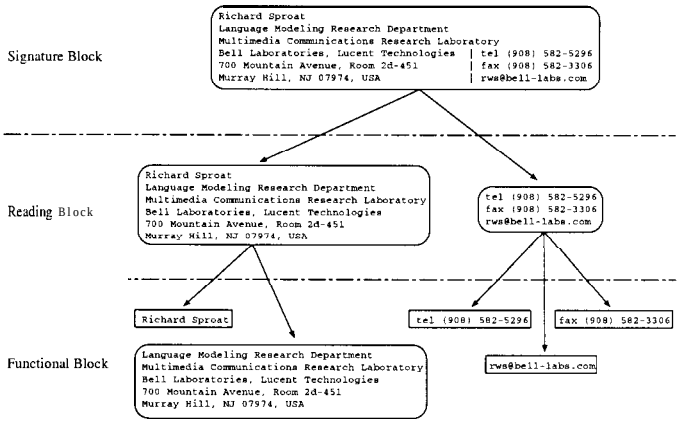

Figure 1. Hierarchical text structure

2. Geometrical Analysis

Geometrical analysis breaks a signature block down to one or more reading blocks, where text in each reading block can be processed continuously. Text in a reading block is usually grouped together spatially, which is identified using a connected component analysis algorithm such as the Line Adjacency Graph (LAG) algorithm [4]. In this algorithm, each line in the text region is broken into several line segments comprised of closely situated characters. Overlapping line segments on adjacent lines are placed into the same connected component and all line segments in a connected component are found from the transitive closure.

Occasionally there are more than one reading block in a connected component detected using the LAG algorithm. Fig. 2 is a typical example where two reading blocks are juxtaposed in the middle and the reading block at the top or bottom connects them together. To correctly segment such a component, line segment extraction and connected component analysis are performed on all *background* (space) characters. A background connected component is considered a *separator* if (1) at least one line segment of the background connected component is in the middle of the reading block, in other words, it does not touch the left or right margin of the reading block; and (2) the total height of the background connected component is greater than a threshold. If a separator is found, the corresponding reading block is broken into three new blocks: one above the separator, one below the separator, and one containing the remaining text. Then, each new reading block undergoes the ordinary connected component analysis again (Fig. 2).

Although carefully designed, the geometrical analysis still makes under-segmentation and over-segmentation errors in line segment extraction. They are to be corrected later using knowledge from linguistic analysis.

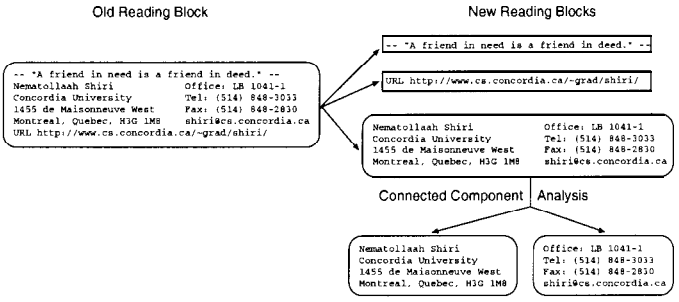

Figure 2. Breaking mixed reading blocks

3. Linguistic Analysis

Linguistic analysis breaks a reading block into several functional blocks and relates each functional block with a functional class. It is carried out using Weighted Finite State Transducers (WFST) [3]. First, the cost of relating a line segment with each functional class is estimated. Then, an input WFST which incorporates all possible choices with their costs is composed with lexicon and grammar WFSTs and the functional class of each line segment is revealed from the optimal path in the final WFST. Most of the over-segmentation and under-segmentation errors are also corrected in linguistic analysis.

Informally, a WFST contains a set of *states* with a distinguished *start state* and one or more *final states* connected by *arcs*. Each arc has an *input symbol*, an *output symbol*, and a *cost*. Each path from the start state to the final state is associated with an *input string*, an *output string*, and a *total cost* (the sum of all costs on the path). The WFST is said to *transduce* the input string into the output string with the total cost. The *composition* of two WFSTs is a new WFST such that if the first WFST transduces string sl into $s2$ with cost cl and the second WFST transduces string $s2$ into $s3$ with cost $c2$, the new WFST transduces sl into $s3$ with cost cl $+ c2$. The *bestpath* algorithm searches an WFST for the optimal path leading from the start state to the final state in the sense that it has the minimum total cost.

WFSTs have been widely used in natural language processing [5]. More recently, they were also shown to be powerful techniques for speech and handwriting recognition, where the recognition process is viewed as a cascade of weighted finite state transductions from the input signal sequence to a word or sentence in a given language [2]. In our current problem, the process of linguistic analysis is formalized as a cascade of transductions from line segments to functional blocks.

3.1. Cost Estimation and Input WFST

For each line segment in the reading block, there are a pair of neighboring nodes in the input WFST connected by several arcs, where the input/output symbol represents a functional class and the cost reflects how likely the line segment is related to that functional class. Ten functional classes are defined as shown in Fig. 3. In addition, two more symbols are used to represent the line break (L) and boundary between reading blocks (B).

Symbol	Functional Class	Example
E	E-mail address	jws@research.bell-labs.com
W	Web address	http://www.bell-labs.com/who/jws
P	Phone number	(908)582-3433
F	Fax number	(908)582-7308
N	Personal name	John W. Smith
A	Postal address	700 Mountain Avenue, ...
T	Title	Associate Professor
Q	Quote	"640K ought to be enough for everyone"
S	Stub	home (following a phone number)
M	Miscellaneous	Address valid until Aug 29, 1997

Figure 3. Functional classes

The first four functional classes (e-mail address, web address, phone and fax numbers) have relatively strict patterns and are termed *strict classes*. The remaining six classes are termed loose *classes*. Strict classes are identified by regular expression matching, during which many under-segmentation errors resulting from geometrical analysis can also be detected. Over-segmentation problem will be taken care of by the language directed segmentation algorithm to be discussed later.

Loose classes are mostly identified by commonly observed conventions and the confidence is much lower than strict classes. This often causes problem in distinguishing personal names from city names, whish are easily confused without semantic knowledge. We propose a personal name identification approach based on e-mail username to ameliorate this problem.

The e-mail username is often derived from the real personal name using the following conventions: 1) a username is constructed by concatenating letter strings directly or via any punctuation characters; 2) the letter strings are prefixes of the first name, middle name, or family name; and 3) each of the first name, middle name, or family name may contribute zero or one prefix as a substring of the username. Usernames constructed by these rules are termed *well-formed* usernames. To estimate if a candidate phrase is a personal name, a well-formed username FST (Fig. 4) is constructed from the candidate phrase, assuming that it is a personal name. Then, a single path FST which generates the username is constructed and composed with the well-formed username FST. If the final FST is non-empty, a low cost is assigned for the phrase to be related to the personal name functional class. Sometimes, the middle initial appears in the username but is omitted from the written personal name. In this case, all 26 letters are considered as candidates for the middle initial.

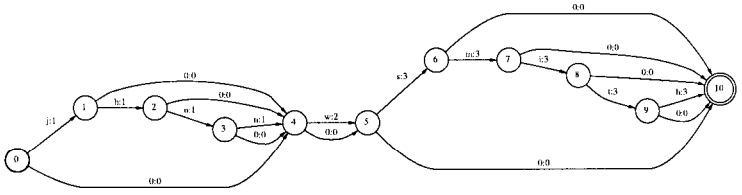

Figure 4. Well-formed username FST for "John W. Smith"

After cost estimation, an input WFST is built for each reading block, as shown in Fig. 5. (Arcs representing line breaks are removed for ease of reading.)

John W. Smith
Rm. 2D-510
Bell Laboratories
700 Mountain Avenue
Murray Hill, NJ 07974
Tel: (908) 582.3433
Fax: (908) 582.7308
e-mail: jws@bell-labs.com

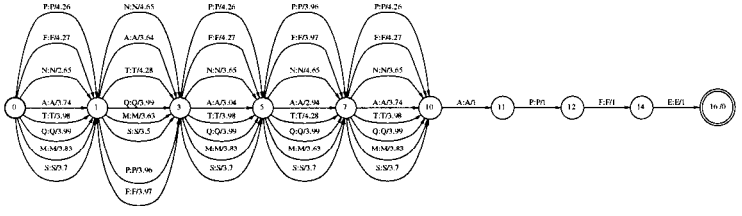

Figure 5. A reading block and its input WFST

One of the two common errors in geometric analysis is over-segmentation of line segments, which may cause serious problems for cost estimation. A pattern in an entire line segment may not be carried by its sub-segments. For example, while "John Smith" is identified as a personal name with regard to the username "jws" by the personal name identification algorithm, neither of the first name or family name alone can be identified in this way.

To solve this problem, a language directed segmentation approach is used. For all the line segments on the same line in a reading block, all possible segmentation positions are evaluated and all the possible combinations are built into the input WFST. Therefore, the input WFST contains choices for not only functional class of each line segment but also segmentation positions on each line of the reading block.

The best choices of both of them are to be determined together after the input WFST is composed with the lexicon and grammar WFSTs. Fig. 6 shows the input WFST built for text line "Dr. John W. Smith", which is broken into four line segments each containing one word (represented as A, B, C, and D respectively in the figure).

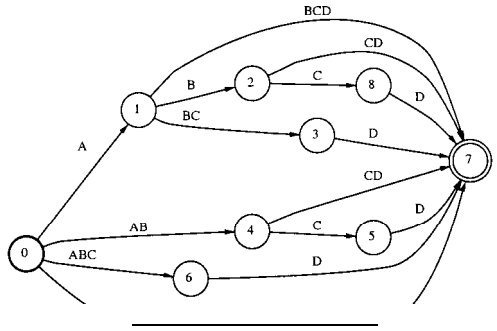

Figure 6. Input WFST incorporated with choices of segmentation positions

3.2. The Lexicon and Grammar WFST

The lexicon WFST describes the construction of a functional block from line segments. For example, a complete postal address could be composed of one or more lines, while a personal name is usually written in one line. The grammar WFST describes the construction of a reading block from functional blocks in a similar fashion. Here to discourage the splitting of one functional block into two or more, a moderate cost is assigned to the transition between functional blocks. Currently weights in both WFSTs are hand selected. They could also be automatically trained once enough labeled training samples are available.

4. Experiments

The signature block analysis algorithm is tested on 1361 signature blocks collected from Lucent Technologies, Department of Computer Science at Concordia University, and various other external sources. They represent a variety of geometrical layouts and writing styles. There are all together 5491 functional blocks in the testing samples and 97% of them are classified correctly. The dominant cause of errors is the mis-segmentation of reading blocks – incorrect merging of two reading blocks due to uncommonly small inter-reading-block spacing or incorrect splitting of one reading block due to uncommonly large intra-reading-block spacing. One possible way to solve this problem is to consider different potential segmentations of reading blocks in a similar fashion as alternative line segments are treated. The challenge is to find an efficient implementation. Currently the average speed of the system is 0.58 second per signature block on a 150 MHz SGI Indy.

The identification performance of the algorithm is evaluated on 347 e-mail messages. The overall recall is 53% (97/183). Among the 86 errors, 79 are trivial one-line signatures such as "-John" or "-J". When excluding these, the recall becomes 93%(97/104). The precision is 90%(97/108).

5. Conclusion

In this paper, we describe a new approach that combines two-dimensional structural analysis with one-dimensional grammatical constraints for analyzing the signature block in an e-mail messages. The geometrical analysis converts the two-dimensional signature block into one-dimensional reading blocks to ensure the coherence of text inside a reading block. The linguistic analysis identifies the functional classes of text in a reading block by taking into account the lexicon and grammar constraints of the signature block and through the use of Weighted Finite State Transducers (WFST).

References

[1] A. Dengel and G. Barth. High level document analysis guided by geometric aspects. *International Journal of Pattern Recognition and Artificial Intelligence*, 2(4):641–655, *1988.*

[2] *I.* Guyon, M. Schenkel, and J. Denker. Overview and synthesis of on-line cursive handwriting recognition techniques. In *Handbook on Optical Character Recognition and Document* Image *Analysis*, chapter 1, pages l-43. World Scientific Publishing Company, 1996.

131 M. Mohri, F. Pereira, and M. Riley. A rational design for a weighted finite-state transducer library. In *Second Int. Workshop on Implementing Automata*, pages 43-53, Ontario, Canada, September 1997.

[4] T. Pavlidis. *Algorithms for Graphics and Image Processing.* Computer Science Press, 1982.

[5] R. Sproat. Multilingual text analysis for text-to-speech synthesis. *Journal of Natural Language Engineering*, pages 369-380, 1997.

[6] A. Takasu, S. Satoh, and E. Katsura. A document understanding method for database construction of an electronic library. In *Proc. 12th CVPR*, pages 263-466, 1994.