Layout Analysis of PDF Documents by Two-Dimensional Grammars for the Production of Accessible Textbooks

Kento Kohase, Shunsuke Nakamura, and Akio Fujiyoshi

Graduate School of Science and Engineering, Ibaraki University, Hitachi, Japan
{20nm711h,20nm726r,akio.fujiyoshi.cs}@vc.ibaraki.ac.jp

Abstract. This paper proposes the use of two-dimensional context-free grammars (2DCFGs) for layout analysis of PDF documents. In Japan, audio textbooks have been available for students with print disabilities in compulsory education. In order to create accessible textbooks including audio textbooks, it is necessary to obtain the information of structure and the reading order of documents of regular textbooks in PDF. It is not simple task because most PDF files only have the information how to print them out, and page-layouts of most textbooks are complex. By using 2DCFGs, we could obtain useful information of regular textbooks in PDF for the production of accessible textbooks.

Keywords: Accessible textbooks · Layout analysis · PDF documents · Two-dimensional grammars

1 Introduction

In Japan, textbooks are considered to be the primary teaching instruments that are necessary to guarantee the quality of education, and audio textbooks have been available for students with print disabilities in compulsory education. They are mainly Multimedia DAISY Textbooks [1] (Fig. 1(a)), Multimodal Textbook [2] (Fig. 1(b)), and AccessReading [3] (Fig. 1(c)). DAISY is a world standard for digital audio books, and textbooks in the format are provided by Japan Society for Rehabilitation of Persons with Disabilities (used by 10,039 students in 2018). Multimodal Textbooks are paper-based textbooks with audio support utilizing invisible 2-dimensional codes and digital audio players with a 2-dimensional code scanner, provided by the authors’ laboratory in Ibaraki University (used by 1,110 students in 2018). AccessReading is digital textbooks in DOCX or EPUB format used with a screenreader on a PC or tablet, provided by Research Center for Advanced Science and Technology, the University of Tokyo (used by 520 students in 2018). Audio textbooks as well as other types of accessible textbooks, such as braille textbooks and large-print textbooks, are made by the efforts of volunteer groups on low budgets in Japan. Since Barrier-Free School Textbooks Act was enforced in Japan in 2008, volunteer groups can obtain PDF files of textbooks from textbook publishing companies.
In order to create any type of audio textbooks, it is necessary to obtain the information of structure and the reading order of documents of regular textbooks in PDF. It is not simple task because most PDF files only have the information how to print them out, and page-layouts of most textbooks are complex. Figure 2 is an example of page-layouts of textbooks. A page-layout is composed of a variety of entities such as texts, lines, words, figures, tables and images (Fig. 2(a)). Region of these entities should be assigned functional or logical labels such as header, footer, main content, sub content, page number, image, etc. (Fig. 2(b)). Hence, we want to have an automatic way to analyze the layout of PDF documents. Though machine learning has attracted attention recently, there are not enough training dataset of the page-layouts of textbooks. This paper proposes the use of two-dimensional context-free grammars (2DCFGs) [4] for layout analysis of PDF documents.
In the formal language theory, 2DCFGs are thought as a powerful tool for layout analysis of two-dimensional objects \[5\]. For parsing documents with a 2DCFG, we can adopt classical bottom-up or top-down algorithm like CKY or Earley parser. However, there is a problem of using 2DCFGs because their computational complexity for parsing is exponential or a high degree of polynomial in size of inputs. To deal with this, some studies proposed to parse linear projections of the input by 1D grammars. This approach results in a good performance, but has drawbacks such as a weaker expressive power caused by the simplified parsing. Průša and Fujiyoshi introduced a subclass of 2DCFGs where productions have a nonterminal rank reducing property. Parsing documents with a rank-reducing 2DCFG is much faster because they can generate regular languages in the case of 1D productions and be applied a simple top-down parsing algorithm using 1D grammars to reduce the number of backtracks. We think their expressive power is sufficient to model page-layouts of textbooks.

The parsing algorithm is implemented as a function of Multimodal Publication Producer \[6\], which is an unified production system for various type of accessible textbooks, developed by the authors’ laboratory. It is used to create Multimodal Textbooks, and its latest version is available on the website \[7\]. Because of the new parsing algorithm, the cost of production of Multimodal Textbooks was much reduced.

For an evaluation of the proposed method, we compare its result to those of two titles of commercial OCR software and two titles of commercial PDF authoring software, namely, FineReader (ABBYY), Yomitorikakumei (Panasonic), Shunkan PDF (Antenna House), and Acrobat Pro (Adobe). Three types of PDF documents are prepared: (1) a section of a tale (1 page, 204 characters), (2) tables of Chinese characters (Kanji) from a textbook in Japanese (1 page, 869 characters), and (3) a section of conversations from a textbook in English (1 page, 479 characters). Text files are created from the PDF documents using the five programs (the proposed method and the four titles of commercial software). The outputs are normalized by eliminating unnecessary spaces and symbols. The correct answers of reading order were manually created. We calculate Levenshtein distances between the each output and the correct answer. The score of the proposed method is the best for the three types of documents among the five programs.

This paper is organized as follows. Layout analysis by two-dimensional grammars is described in Sect. 2. Evaluation of the proposed method is described in Sect. 3. Finally, a conclusion of the proposed method is described.

2 Layout Analysis by Two-Dimensional Grammars

2.1 Definition of Two-Dimensional Context-Free Grammars

A two-dimensional context-free grammar (2DCFG) is a tuple \(G = (N, T, P, S)\), where \(N\) is a set of nonterminals, \(T\) is a set of terminals, \(S \in N\) is the initial nonterminal and \(P\) is a set of productions in one of the following forms:

\[
(1)\ N \to A,
(2)\ N \to A | B,
(3)\ N \to A / B,
\]
Fig. 3. Example of layout analysis by a 2DCFG

where \( N \in \mathcal{N} \) and \( A, B \in T \cup \mathcal{N} \). The productions follow the Chomsky normal form. A production of type (2) generates a horizontal connection composed of \( A \) and \( B \). Analogously, a production of type (3) generates a vertical connection composed of \( A \) and \( B \).

Figure 3(a) is an example of a 2DCFG, which is used to parse documents composed of regions of a title, an abstract, a page number, and two columns consisting of text lines. When we parse a document (Fig. 3(b)) using the grammar, a derivation tree can be obtained (Fig. 3(c)). Elemental regions are assigned terminals such as “\( \langle\text{textLine}\rangle \)” and “\( \langle\text{number}\rangle \)” . \( \langle\text{textLine}\rangle \)” means a group of adjoining characters of the same font and size, while \( \langle\text{number}\rangle \)” means a group of adjoining digits. A production of type (2) or type (3) divides a set of elemental regions into two parts by assigning a nonterminal to each of them.

2.2 Rank-Reducing Property

We say that a 2DCFG \( \mathcal{G} = (\mathcal{N}, T, \mathcal{P}, S) \) is rank-reducing iff, for each \( N \in \mathcal{N} \), \( N \) is assigned a positive integer which is called a rank of \( N \), and, for each production \( N \to B \), \( N \to A \mid B \) or \( N \to A / B \), the values of rank are \( N \geq B > A \), i.e., a nonterminal generates nonterminals whose rank is not greater than its. Let \( \mathcal{G}_T \) and \( \mathcal{G}_L \) are 1D grammars generated from a rank-reducing 2DCFG as described in [4]. Then, both \( \mathcal{G}_T \) and \( \mathcal{G}_L \) generate a regular language . This fact can be used to reduce a number of backtracks in a parsing process.

2.3 Top-Down Parsing Algorithm

Algorithm 1 shows an execution of the top-down parsing. The first call of \textsc{Parse} passes to it a grammar \( \mathcal{G} = (\mathcal{N}, T, \mathcal{P}, S) \), the initial nonterminal \( S \) and the set of terminal regions “terms”. Procedure \textsc{Matches} (\textit{terms}, \( V \)) returns true iff “terms” contains the only terminal region assigned by \( V \in \mathcal{N} \cup T \). Procedure \textsc{FindSplitPoints} (\textit{terms}, \( P \)) detects all vertical/horizontal cuts by production
Given a vertical/horizontal cut (determined by a point $s$), $\text{Split (terms, } s, P)$ divides terminal regions of “terms” into two parts.

**Algorithm 1. Top-down parsing**

**Input:** $G = (N, T, P, S)$, $V \in N \cup T$, terms

1: procedure $\text{PARSE}(G, V, \text{terms})$
2: if $V \in T$ and $\text{MATCHES(terms, } V)$ then
3: return true
4: for each $P \in P$ do
5: if $P = V \rightarrow A$ and $\text{PARSE}(G, A, \text{terms})$ then
6: return true
7: else if $P = V \rightarrow A | B$ or $P = V \rightarrow A / B$ then
8: SplitPoints = $\text{FINDSPLITPOINTS(terms, } P)$
9: for each $s \in \text{SplitPoints}$ do
10: $[\text{termsA, termsB}] = \text{SPLIT(terms, } s, P)$
11: if $\text{PARSE}(G, A, \text{termsA})$ and $\text{PARSE}(G, B, \text{termsB})$ then
12: return true
13: return false

2.4 Implementation

The parsing algorithm is implemented as a function of Multimodal Publication Producer [6], which is an unified production system for various types of accessible textbooks. The system is an application software developed on Java with standard widget toolkit (SWT) and Apache PDFBox library [8]. It runs on multi-platforms: Windows and OS X. Neighboring characters are grouped based on the gap between their bounding boxes. Each group of characters is defined as an elemental region consisting of a bounding box and a word content. We input a list of productions of a 2DCFG as a text file, and the grammar can be changed by editing the text file. Thus, the layout analysis can be used for various types of page-layouts.

2.5 Grammars for Describing of Page-Layouts of Textbooks

There are many types of layouts of a textbook. A textbook in Japanese language has pages such as tales, table of contents, guides, and some types of tables of Chinese characters (Kanji). In order to analyze these pages with different layouts, it is necessary to describe a grammar for each layout of pages. We described 10 different grammars for a textbook in Japanese language, and they are selected automatically or manually in the system. For example, a grammar describing a page of a tale from a textbook in Japanese language composed of 17 nonterminals and 25 productions, and a grammar describing a page of a table of Chinese characters composed of 29 nonterminals and 43 productions.
Figure 4 is a result of layout analysis of a page of a tale. When the page is parsed by the grammar, a derivation tree whose internal nodes are labeled by a nonterminal of the grammar can be obtained (Fig. 4(b)). The reading order and the classification of elements are determined by the derivation tree. Figure 4(a) shows the classification obtained from the derivation tree. A classification consists of a variety of entities such as BODY, HEAD, NOTE and AUXILIARY for a page of a tale.

![Diagram of layout analysis](image)

**Fig. 4.** A Result of layout analysis of a textbook

### 3 Evaluation

For an evaluation of the proposed method, we compare five programs: Multi-modal Publication Producer, FineReader (ABBYY), Yomitorikakumei (Panasonic), Shunkan PDF (Antenna House), and Acrobat Pro (Adobe).

#### 3.1 Methods

Three types of PDF documents are prepared: (1) a section of a tale (1 page, 204 characters), (2) tables of Chinese characters (Kanji) from a textbook in Japanese (1 page, 869 characters), and (3) a section of conversations from a textbook in English (1 page, 479 characters). The orders of elements stored in PDF documents are unchanged as they were made in textbook publishing companies. The elements of the document (2) starts from the bottom line though its appropriate reading order should start from the top line.

Text files are created from the PDF documents using the five programs, and then, they are normalized by eliminating unnecessary spaces and symbols. The correct answers of reading order were manually created. Levenshtein distances between each output and the correct answer are calculated.
3.2 Results

The Levenshtein distances between outputs and the correct answer are shown in Table 1. Acrobat Pro just outputs a text file in the order of elements stored in a PDF file. The results of the document (1) shows that the other programs do a kind of layout analysis because their outputs are better than Acrobat Pro. The results of the document (2) shows that layout analysis does not always work properly. The layout of the document (2) requires groupings of elements as shown in Fig. 5(a). However, Shunkan PDF put the elements in a simple order without groupings as shown in Fig. 5(b). The results of FineReader and Yomitorikakumi tend to be poor because they first convert PDF documents into raster images and then use their OCR engines to recognize characters. Misrecognitions occur for some characters.

| Document                      | (1) | (2) | (3) |
|-------------------------------|-----|-----|-----|
| Multimodal Publication Producer | 10  | 0   | 46  |
| FineReader                    | 53  | 430 | 142 |
| Yomitorikakumi               | 40  | 630 | 231 |
| Shunkan PDF                   | 16  | 780 | 512 |
| Acrobat Pro                   | 62  | 280 | 423 |

Fig. 5. Groupings of elements of the document (2)
3.3 Discussion

If the order of elements in a PDF document is maintained correctly, Acrobat Pro outputs without problems. Shunkan PDF can analyze accurately if layout is simple. FineReader and Yomitorikakumei are equipped with OCR engines, so they can recognize characters in images and vector graphics.

By describing a grammar to model page-layouts, the proposed method can analyze page-layouts with high accuracy. We think the expressive power of rank-reducing 2DCFGs is sufficient to model page-layouts of textbooks used in compulsory education in Japan.

4 Conclusion

This paper proposed the use of two-dimensional context-free grammars (2DCFGs) [4] for layout analysis of PDF documents. Implementing the parsing algorithm as a function of Multimodal Publication Producer [6], a detailed layout analysis of the textbooks becomes possible.

At present, we have only grammars for page-layouts of textbook for compulsory education in Japan. In the future, we want to design a grammar which is suitable to analyze all page-layouts of the documents.

The latest version of Multimodal Publication Producer with the parsing algorithm for 2DCFGs is available on the website [7].

References

1. DAISY. https://www.dinf.ne.jp/doc/daisy/book/
2. Fujiyoshi, A., Fujiyoshi, M., Ohsawa, A., Ota, Y.: Development of multimodal textbooks with invisible 2-dimensional codes for students with print disabilities. In: Miesenberger, K., Fels, D., Archambault, D., Peñáz, P., Zagler, W. (eds.) ICCHP 2014. LNCS, vol. 8548, pp. 331–337. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-08599-9_50
3. AccessReading. https://accessreading.org/
4. Pruša, D., Fujiyoshi, A.: Rank-reducing two-dimensional grammars for document layout analysis. In: 2017 14th IAPR International Conference on Document Analysis and Recognition, pp. 1120–1125 (2017)
5. Álvaro, F., Cruz, F., Sánchez, J.-A., Ramos Terrades, O., Benedito, J.-M.: Structure detection and segmentation of documents using 2D stochastic context-free grammars. Neurocomputing 150, 147–154 (2015). https://doi.org/10.1016/j.neucom.2014.08.076
6. Takaira, T., Tani, Y., Fujiyoshi, A.: Development of a unified production system for various types of accessible textbooks. In: Miesenberger, K., Bühler, C., Penaz, P. (eds.) ICCHP 2016. LNCS, vol. 9758, pp. 381–388. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-41264-1_52
7. Multimodal Publication Producer. http://apricot.cis.ibaraki.ac.jp/MultimodalPublicationProducer/
8. PDFBox. https://pdfbox.apache.org/