

Automated Layout Preservation in Cross Language Translation of Document : An Integrated Approach and Implementation *

Vivek Yadav
International Institute of Information Technology
Bangalore, India
vivek.yadav@iiitb.org

Chandrashekar Ramanathan
International Institute of Information Technology
Bangalore, India
rc@iiitb.ac.in

ABSTRACT

Layout refers to format and placement of content in a document. Cross language document translation is a well known problem and is addressed widely. Such translations usually remain unsuccessful in preserving original format of document because of typographical differences across the script of different languages. To make translated document look aesthetically identical to the original document, preservation of layout is essential. In this paper, we propose an integrated approach to solve various problems that arise during the process of translation pertaining to the layout of document like content flow, table of content, maintaining relative position and aesthetics of content.

Categories and Subject Descriptors

1.7.4 [Document and Text Processing]: Electronic Publishing; 1.7.2 [Document and Text Processing]: Document Preparation—*Format and notation*

General Terms

Algorithm

Keywords

Automated layout, cross language translation, format preservation

1. INTRODUCTION

High quality translation of document typically requires substantial effort in terms of translation and designing process. The translated document tend to loose features like paragraph structure, content flow, drop cap and overall document composition without designer involvement. If the document is highly sophisticated and long, manual approach

*All photographs used have Creative Commons licenses and copyright to their owners.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

Compute'14 Oct 9-11, Nagpur, Maharashtra, India Copyright 2014
ACM 978-1-60558-814-8/00/0014 ...\$15.00.

for layout preservation is inefficient. Professional designing of document requires high marginal cost, therefore translation along with designing is an expensive process. Efficient automation of this process is desired for publishers convenience and consumers best experience. Researchers have addressed issues related to automated translation and document composition [5]. Automated translation is being performed efficiently on raw textual content extracted from document, while automated document composition is handled separately. Both these techniques can not be applied sequentially as textual content obtained from document loses structural information related to the layout of document.

Automated layout preservation in cross language translation of document attempts to automate the whole process of translation and generation of document for which layout is same as the original document. The major problem addressed while translating the document is handling metadata like font family, font size, etc. The difference in script and typographical aspects of languages makes this problem even more challenging. Further more, the heterogeneity of content within documents and across documents requires multiple strategies to be followed. Arranging the two processes of translation and formatting, in the same pipeline raises other constraints such as the need for two processes to be generic for any type of content and must not be tightly coupled with each other. Our integrated approach tries to overcome these problems and enables to deliver impressive result by preserving layout in the translated document.

In the following sections, we will discuss an integrated approach to handle problems related to cross language translation of document. This approach is divided into two major phases:

- Phase I - Font Signature Mapping : Generate font signature mapping for specific font size and family in source and target languages.
- Phase II - Document Processing : Translate document from one language to other with format preservation.

Document processing is further divided into four subtasks:

1. Document preprocessing
2. Drop cap processing
3. Parallel translation and mapping
4. In-place content replacer

The above tasks are explained in detail in the later sections.

2. RELATED WORK

Over the last five decades, machine translation is widely used to automatically translate textual content between different languages. There are several techniques used in machine translation like rule-based machine translation, knowledge based machine translation, dictionary based machine translation etc. The Georgetown-IBM experiment [5] demonstrated in January, 1954 was one of the first rule based machine translator. In order to perform machine translation of the document, there is a need for information extraction system. Several articles have been written on information extraction from document [1]. One of the major survey paper written in this field is by Jung [6]. Optical character recognition for information extraction is one of the major techniques. Most of these techniques can only extract textual information from the document. It is essential to extract other elements of a document like font properties, layout etc along with the textual information. Documents that have separate textual and layout information like hyper text markup language (HTML) and cascading style sheet (CSS) can be translated easily without discrepancy. Since there are typographic differences across different languages, layout changes while translating documents. This, in particular is not an issue in the web pages because HTML is free flowing document as opposed to PDF which is paginated document. Wikibhasha [7] framework is used to create content in multiple languages for wiki site.

Once the information is extracted from the document, automatic document composition can be used to form new document in target language. The survey paper written by Hurst [4] and Lok [8] focuses on automatic document composition. In this process of automatic document composition, in-built templates created by the designer are used. This process is not fully automated and requires human intervention at several stages. Several automatic document composition models have been introduced in past, like probabilistic model [2], grid model [3], constraints model [9] and machine learning model [11]. All these techniques are helpful in generating new layout of document but they are not intended for preserving the original layout of the document.

Automated layout preservation in cross language translation is not reported widely. In current literature, information retrieval and document composition exist separately. There is a need to look for ways, where the information retrieval and document composition processes can be carried together at the same time for a document. This paper illustrates an approach to carry these two processes of translation and composition at the same time for the document.

3. PROPOSED APPROACH

Cross language translation of document is a two step process. The first phase named font signature mapping, is an iterative process that correlates typographical information like font-size between two languages. This approach is represented using an algorithm which generates a font signature mapping table. The next phase named document processing, takes document as an input and generates a translated document. It includes four sub-processes: document preprocessing, drop cap processing, parallel translation and mapping, in-place content replacement. Here, we describe these tasks in detail.

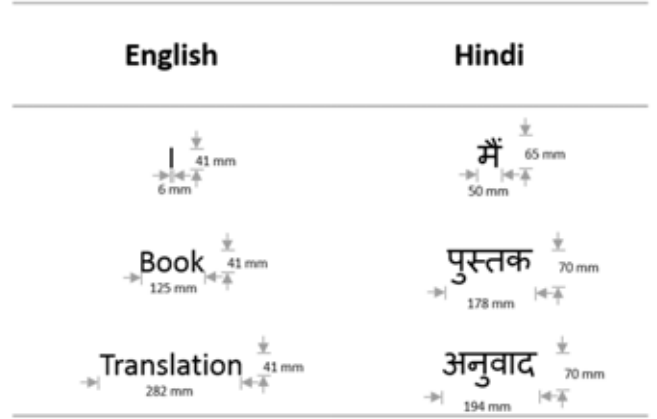


Figure 1: Typographic difference between English and Hindi language

3.1 Font signature mapping

There are typographic differences between the script of different languages. Hence, the amount of area required to layout glyph in one language is different from its translation in another language. This is shown in the Figure 1. These typographical differences change the layout of document when it is translated to other language.

In order to solve the above mentioned problem, we propose a font signature mapper which will map font-size in one language to the other language. This font signature mapping is a one time process which takes set of documents as input and generates a font signature mapping table. In Figure 2, we have explained this process. Every sample document is a one-page document which is entirely filled with text of single font-size and family of source language. Initially, the sample document in source language is converted to target language using translation. If the content of the translated document exceeds one page, we decrease the font size. On the other hand, if the content of the translated document is less than one page, we increase the font size. This process iterates till the content in the translated document fits optimally in a single page. We used a binary search approach for making this iterative process efficient. We repeat the same process on a set of documents having different content however same source language and same font property. The optimal font size of the target language is recorded as an outcome for each document. We calculate the resultant font-size in target language as mean of all these outcomes. When the same process is carried for multiple sets of document, with different font property, font signature mapping table is generated. This table provides mapping between various font property of source and target languages. Once the font signature table is generated, every document which needs to be translated will be processed using this table by phase two - document processing. The major advantage of this approach is that we have to perform this font signature mapping only once for a particular font family across two different languages.

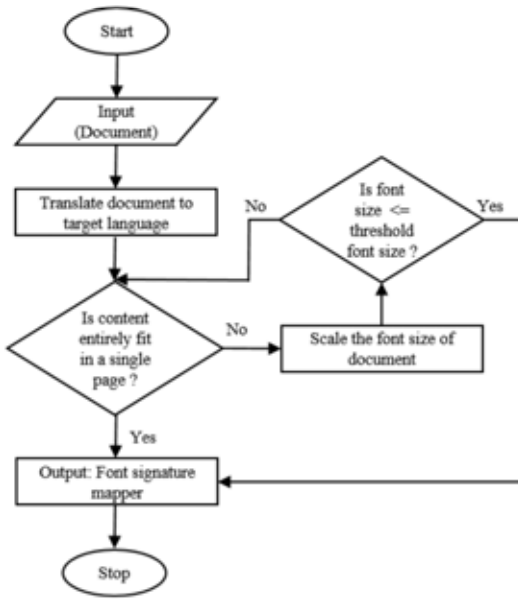


Figure 2: Font signature mapper

3.2 Document processing

Document processing involves four sub-processes: document preprocessing, drop cap processing, parallel translation and mapping, and in-place content replacement. We describe each of these tasks in detail in the following sections.

3.2.1 Document preprocessing

Document preprocessor takes care of two major tasks. Initially, it converts source document to Microsoft office XML word document. XML document comprises of different elements including text and font property. We extract these text and font properties and store them in two separate lists namely text list and font list. Translation of text is being done using online web service provided by Bing Translator. In order to reduce the network overhead of accessing the translator for each text elements in the text list separately, certain optimization were done in the implementation. We used a non-source language character as a separator to combine multiple elements in the text list and sent them to the translator together. The translator translates each element of the text list into target language. However, it leaves the separators as they are. The translated text was separate into target language text elements using the separators.

3.2.2 Drop Cap processing

In publishing, drop cap is the phenomenon in which the first letter of the first word is enlarged as compared to other letters. For example, the word “India” in Figures 4 and 5, first letter “I” has larger font size than the rest of the letter in the word. “I” and “ndia” have different font property, therefore they are stored separately in internal structure of the document. This will cause problem in translation, as both these of these segments “I” and “ndia” will be translated separately. We have come up with an approach to identify the words with drop cap. The approach is explained using

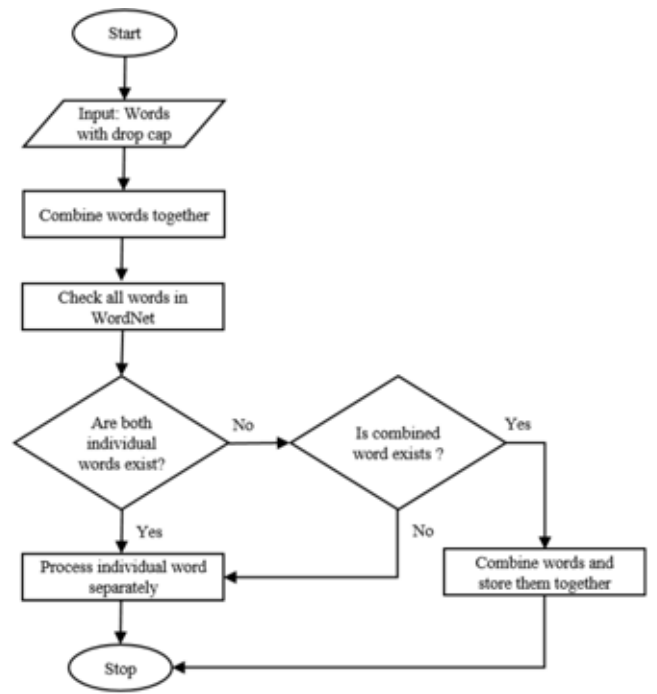


Figure 3: Drop cap processor

I ndia —→ मैं ndia ✗

Figure 4: Translated without drop cap processing

I ndia —→ भारत ✓

Figure 5: Translated using drop cap processing

flow chart in the Figure 3.

These words are combined and sent to the word-net [10] library for verification. If the combined word exists and at the same time individual word does not exist, both these words are combined and stored. Optimized list is processed by drop-cap processor which resolves the drop cap issues in the document. While composing target document, the effects related to every drop-cap are regenerated.

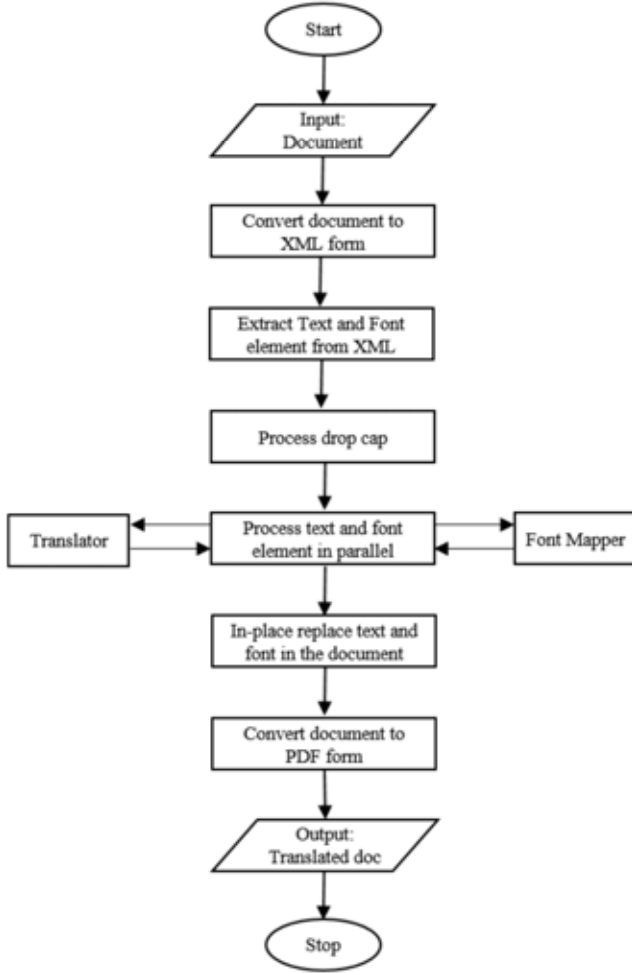


Figure 6: Document processor

3.2.3 Parallel translation and mapping

Parallel translation and mapping is an approach for reducing the time to translate the document. Online translator is used in this process to translate content of documents. Essentially, lot of requests will be sent to the online translator for translation of content. This makes the process inefficient in terms of time. To manage this issue, we have to send minimum number of requests to the online translator. This is achieved by performing an optimization technique on a list as discussed in previous section. The task of font-size mapping on this list is performed by using the table generated from the font signature mapping process. Both the process of translation and mapping will be executed concurrently. The translated list is given to in-place content replacer for

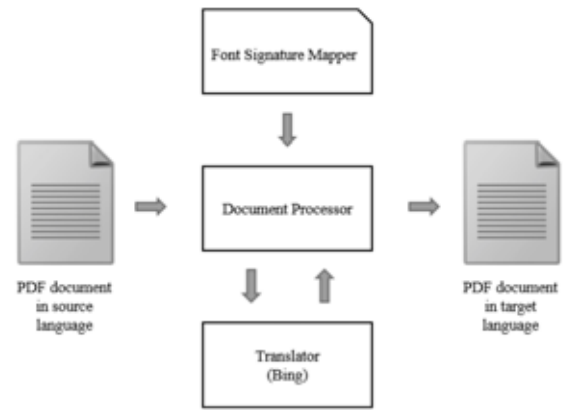


Figure 8: System architecture

further processing.

3.2.4 In-place content replacing

This step takes original document and translated list as input. All the textual content in the document is replaced by its translated content using translated list. This will ensure that the original layout of the document is not changed. Due to font-size mapping, relative position of the content in the document is preserved in terms of layout and as a result the translated document appear aesthetically similar and consistent with the original document.

4. IMPLEMENTATION DETAILS

4.1 System Architecture

Algorithm 1: FONT SIGNATURE MAPPER: Generate font signature mapping for specific font size in source and target languages

Input: A set $D = \{d_1, d_2, \dots, d_n\}$ of document with same font property, where d_1, d_2, \dots, d_n are all single page document entirely filled till the last line of the page

Output: Mean font-size in target language which occupies same area in layout of source language i.e $\sum_{i=1}^n f_i/n$ where f_i is target font size for individual document

```

s ← 0
for i ← 1 to n do
    dti ← translate(di)
    while dti content does not fit in single page do
        scale the font size fi of document dti
    s ← s + fi
return s/n
  
```

It consist of two modules: font signature mapper and document processor. The input and output to the document processor are PDF documents. This document processor is implemented in C# using dot net framework. The proposed system architecture is shown in Figure 8. Algorithm 1 generates font signature mapping table for specific font size and family in source and target language. Algorithm 2 actually

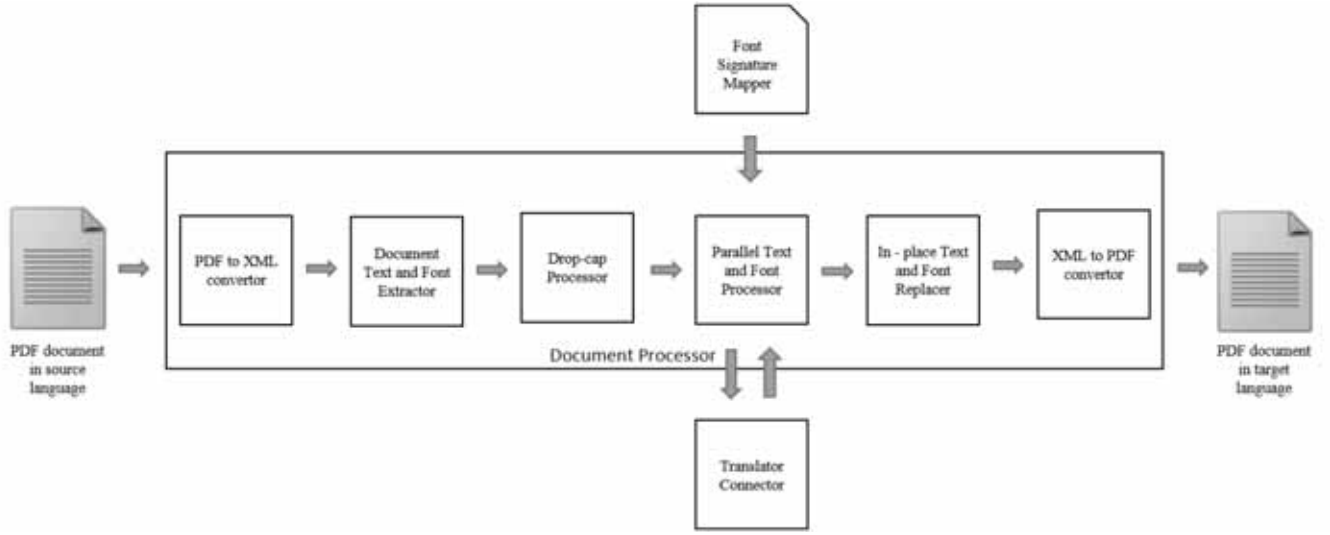


Figure 7: Document processor architecture

translates document from one language to other. The input and output for algorithm 1 are PDF documents. Detail of system architecture of document processing is explained in Figure 7.

Algorithm 2: DOCUMENT PROCESSOR: Translate document from one language to other

Input: A document D_S in source language l_s , target language l_t and mapping font-size table from source to target language i.e $\text{Map}(f_s, f_t)$

Output: A document D_t in target language l_t

$S_{xml} \leftarrow \text{convertToXml}(D_S)$

$\text{List}_{txt} \leftarrow \text{getTextXmlElement}(S_{xml})$

$\text{List}_{ft} \leftarrow \text{getFontXmlElement}(S_{xml})$

$\text{Str} \leftarrow \text{listToString}(\text{List}_{txt})$

$\text{Str}_{dp} \leftarrow \text{processDropcap}(\text{Str})$

$n \leftarrow \text{numberOfSubstring}(\text{Str})$

for $i \leftarrow 1$ **to** n **do**

parallel translate string and store result to Str_r
 along with process font to map target language
 $\text{translate}(\text{substring}(\text{Str}))$
 $\text{fontMapper}(\text{List}_{ft}, \text{Map}(f_s, f_t))$

$\text{List}_{rtxt} \leftarrow \text{stringToList}(\text{Str}_r)$

$D_{temp} \leftarrow \text{openDocument}(\text{Str}_r)$

$D_{xml} \leftarrow \text{inplaceReplacer}(D_{temp}, \text{List}_{rtxt}, \text{List}_{ft})$

$D_t \leftarrow \text{convertToPdf}(D_{xml})$

return D_t

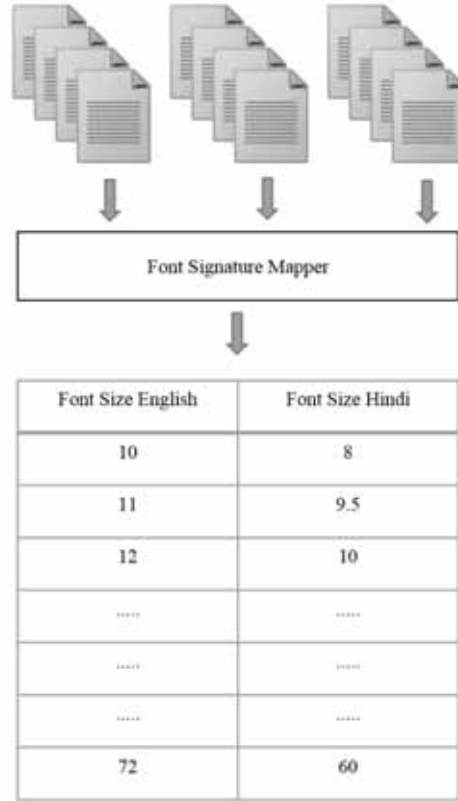



Figure 11: Result of Font signature mapper

5. EXPERIMENTAL RESULT

In this section, we initially demonstrate the result obtained by font signature mapper in Figure 11. We later illustrate the performance of our cross language translator with example a, b in Figures 9 and 10. We have used 50 document set to generate font signature mapping table. Code for this experiment has been written in C# using dot net framework.

Title → **DRAINAGE**

Picture → 


Paragraph →

The term drainage describes the river system of an area. Look at the physical map. You will notice that small streams flowing from different directions come together to form the main river, which ultimately drains into a large water body such as a lake or a sea or an ocean. The area drained by a single river system is called a drainage basin. A closer observation on a map will indicate that any elevated area, such as a mountain or an upland, separates two drainage basins. Such an upland is known as a water divide.

Heading → **DRAINAGE SYSTEMS IN INDIA**

The drainage system of India are mainly controlled by the broad relief features of the subcontinent. Accordingly, the Indian rivers are divided into two major groups:

- the Himalayan rivers, and
- the Peninsular rivers.

Picture → 

Apart from originating from the two major physiographic regions of India, the Himalayan and the Peninsular rivers are different from each other in many ways. Most of the Himalayan rivers are perennial. It means that they have water throughout the year. These rivers receive water from rain as well as from melted snow from the lofty mountains. The two major Himalayan rivers, the Indus and the Brahmaputra originate from the north of the mountain ranges. They have cut through the mountains making gorges. The Himalayan rivers have long courses from their source to the sea. They perform intensive erosional activity in their upper courses and carry huge loads of silt and sand. In the middle and the lower courses, these rivers form meanders, oxbow lakes, and many other depositional features in their floodplains. They also have well-developed deltas. A large number of the Peninsular rivers are seasonal, as their flow is dependent on rainfall. During the dry season, even the large rivers have reduced flow of water in their channels.

(a)

Title → **जल निकासी**

Picture → 

Paragraph →

वाह्य जल निकासी जहाँ जलवायु एक क्षेत्र का निर्धारण करता है। भौतिक अवस्था पर निर्भर। एक भौतिक अवस्था कि जहाँ जलवायु अवस्था अवस्था निर्धारण में बहुत एक समय आ समय नहीं है। जो जलवायु एक क्षेत्र जहाँ एक क्षेत्र का निर्धारण एक क्षेत्र के रूप में निर्धारण के रूप में। एक क्षेत्र जहाँ जलवायु द्वारा निर्धारण क्षेत्र जल निकासी निर्धारण करता है। एक क्षेत्र पर एक क्षेत्र अवस्था में निर्धारण है कि निर्धारण क्षेत्र क्षेत्र, जैसे कि एक क्षेत्र का एक floodplain, जो जल निकासी निर्धारण की अवस्था करता है। इस क्षेत्र एक floodplain एक जल निकासी के रूप में जलवायु है.

Heading → **द्वैत निकासी क्षेत्र में**

भारत की जल निकासी जलवायु की अवस्था बहुत भूमिगत के द्वारा निर्धारण रूप में निर्धारण का एक निर्धारण। निर्धारण, भारतीय जलवायु की क्षेत्र निर्धारण है:

- हिमालय जलवायु, जो
- Peninsular जलवायु.

Picture → 

दो प्रमुख भौतिक क्षेत्रों में भारत के दो क्षेत्र निर्धारण में, हिमालय और पश्चिमी क्षेत्रों की अवस्था में एक-दूसरे के निर्धारण है। अर्थात् हिमालय जलवायु निर्धारण करता है। इसका निर्धारण है कि निर्धारण नहीं है। इन जलवायु में निर्धारण और क्षेत्र जो जलवायु निर्धारण के निर्धारण क्षेत्र जलवायु क्षेत्र। दो प्रमुख हिमालय जलवायु, सिंधु और ब्रह्मपुत्र जलवायु निर्धारण के जलवायु निर्धारण। वे floodplain जलवायु निर्धारण के निर्धारण है। हिमालय जलवायु निर्धारण के निर्धारण क्षेत्र में जलवायु निर्धारण निर्धारण और जलवायु निर्धारण का निर्धारण क्षेत्र में जलवायु निर्धारण निर्धारण क्षेत्र में, इन जलवायु क्षेत्र, oxbow झील, और कई अन्य depositional भूमिगत में जलवायु निर्धारण क्षेत्र। वे जलवायु निर्धारण क्षेत्र निर्धारण क्षेत्र। पश्चिमी क्षेत्रों की एक क्षेत्र निर्धारण है निर्धारण, उनके निर्धारण के रूप में निर्धारण के जलवायु निर्धारण है. एक क्षेत्र के दो क्षेत्र निर्धारण की है उनके निर्धारण में निर्धारण क्षेत्र निर्धारण क्षेत्र.

(b)

Figure 10: NCERT social science book page in two-column format and its translated document

Figures 9 (a) and 10 (a) show the source document in different layout having same textual and image content. Figures 9 (b) and 10 (b) illustrate the output PDF obtained after these source document are processed. We can see that algorithm preserves the layout in both the cases. We have used content from NCERT social science book for our experiment. The translated document is rich in aesthetic and shows that the layout is preserved in different documents.

6. CONCLUSIONS

This paper presents a framework for automatic translation of documents that preserves the appearance and layout of the source document. One of the main challenges in document translation is layout preservation, which is addressed in the approach proposed in this paper. The key contributions of this paper include algorithms and approaches for: (a) handling issues with size mismatch in the font glyphs between source and target languages (b) retention of the location of images in the page flow across translations (c) handling translation problems when drop caps are present in the source text (d) efficient and intelligent use of web-based translation APIs for massive improvement in performance.

Our novel approach proves to be advantageous across various aspects related to the document layout preservation like document flow, table of contents, and relative position of contents during translation process. Work is in progress to further enhance the system to automatically create translated versions of a large number school text books available in English into other Indian languages. The process can be easily modified to a hybrid process, if there is a need to introduce manual intervention for validating / fine-tuning the translation (like in Wikibhasha) before finally publishing the document.

7. ACKNOWLEDGMENTS

We would like to thank Microsoft Research India for their support in this research.

8. REFERENCES

- [1] S. Christodoulakis, M. Theodoridou, F. Ho, M. Papa, and A. Pathria. Multimedia document presentation, information extraction, and document formation in minos: A model and a system. *ACM Trans. Inf. Syst.*, 4(4):345–383, Dec. 1986.
- [2] N. Damera-Venkata, J. Bento, and E. O’Brien-Strain. Probabilistic document model for automated document composition. In *Proceedings of the 11th ACM symposium on Document engineering*, pages 3–12. ACM, 2011.
- [3] S. Feiner. A grid-based approach to automating display layout. *Readings in intelligent user interfaces*, pages 249–255, 1998.
- [4] N. Hurst, W. Li, and K. Marriott. Review of automatic document formatting. In *Proceedings of the 9th ACM symposium on Document engineering*, pages 99–108. ACM, 2009.
- [5] W. Hutchins. The georgetown-ibm experiment demonstrated in january 1954. In R. Frederking and K. Taylor, editors, *Machine Translation: From Real Users to Research*, volume 3265 of *Lecture Notes in Computer Science*, pages 102–114. Springer Berlin Heidelberg, 2004.
- [6] K. Jung, K. I. Kim, and A. K. Jain. Text information extraction in images and video: a survey. *Pattern Recognition*, 37(5):977 – 997, 2004.
- [7] A. Kumaran, N. Datha, V. Dendi, and A. Sharma. Wikibhasha: Ourexperiences with multilingual content creation tool for wikipedia. In *Proceedings of the Wikipedia India Conference 2011*. Wikimedia Foundation, December 2011.
- [8] S. Lok and S. Feiner. A survey of automated layout techniques for information presentations. *Proceedings of SmartGraphics*, 2001, 2001.
- [9] T. Masui. Evolutionary learning of graph layout constraints from examples. In *Proceedings of the 7th Annual ACM Symposium on User Interface Software and Technology*, UIST ’94, pages 103–108, New York, NY, USA, 1994. ACM.
- [10] G. A. Miller. Wordnet: A lexical database for english. *Commun. ACM*, 38(11):39–41, Nov. 1995.
- [11] M. Zhou, S. Ma, and Y. Feng. Applying machine learning to automated information graphics generation. *IBM Systems Journal*, 41(3):504–523, 2002.