Unified Devanagari Rendering Engine for Nepali Language

Team Members

Amar Dura [THA077BCT007]
Ayush Bhandari [THA077BCT014]
Harish Joshi [THA077BCT018]
Sugam Pokharel [THA077BCT044]

Supervised By
Er. Praches Acharya

Co-supervised By
Er. Santa B. Basnet

Department of Electronics and Computer Engineering
Institute of Engineering
Thapathali Campus
Kathmandu, Nepal

PRESENTATION OUTLINE

- Motivation
- Objectives
- Scope of Project
- Project Applications
- Methodology
- Results
- Discussion of Results
- List of Remaining Tasks
- References

MOTIVATION

- Lack of open source tools for pdf rendering for Nepali language
- Inconsistency in composite character representation in different Devanagari fonts [विद्या, विद्या]

OBJECTIVES

- To implement glyph ordering mechanism and standardize composite character representation

  { क, ि } - { ि, क} - कि

- To develop an open source unified Devanagari rendering engine for JVM using Apache PDFBox

SCOPE OF PROJECT

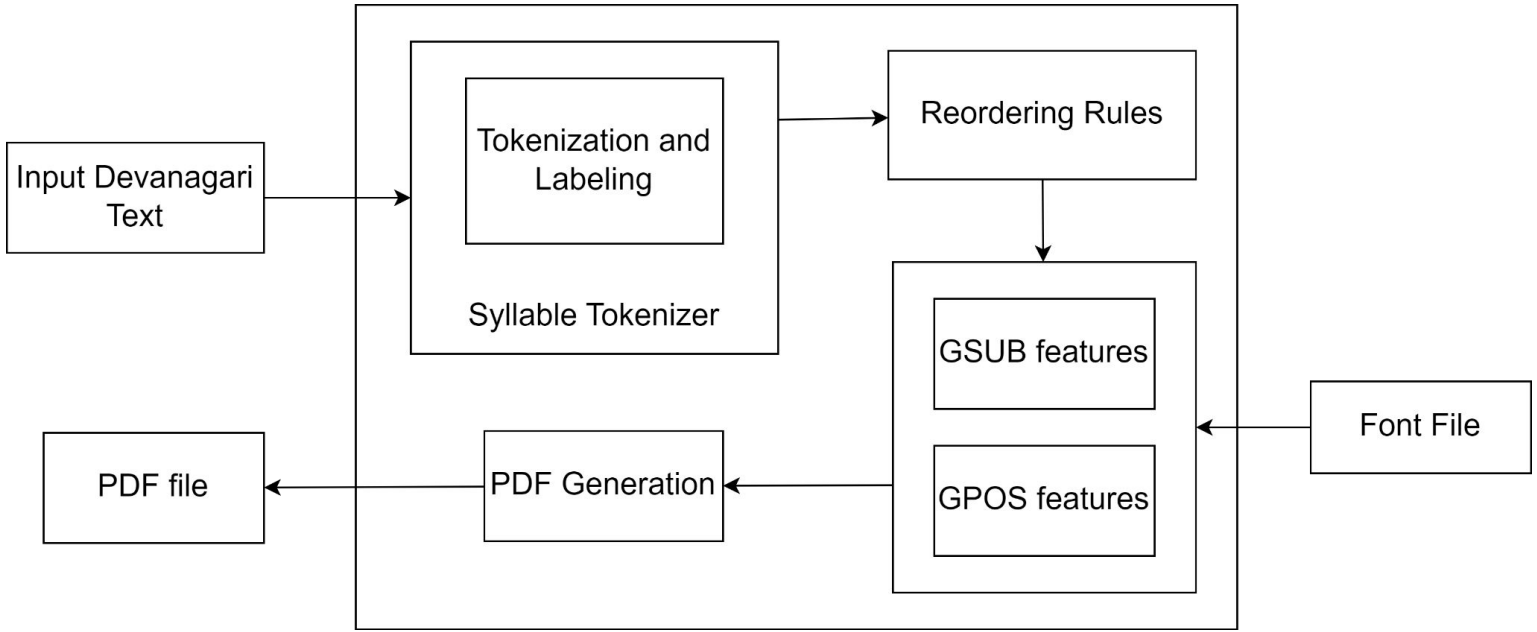
- Ensures the correct ordering of glyphs for Nepali text.
- Focus on pdf rendering only.
- Only implemented for multi-byte Unicode fonts
- Apache PDFBox works on JVM only.

PROJECT APPLICATIONS

- Bill Generation
- News Monitoring Tools
- Text Preprocessing Pipeline
- Governmental documentation

METHODOLOGY - [1]

System Block Diagram



METHODOLOGY - [2]

Input Devanagari Text

- from text document, database or text repository.
- contains the unicode devanagari text

Example:

“भानुभक्तका हजुरबुवा श्रीकृष्ण आचार्य जुम्ला जिल्लाको सिञ्जा उपत्यकाबाट तनहुँ जिल्लामा बसाइँ सरेका थिए।”

METHODOLOGY - [3]

Syllable Tokenizer

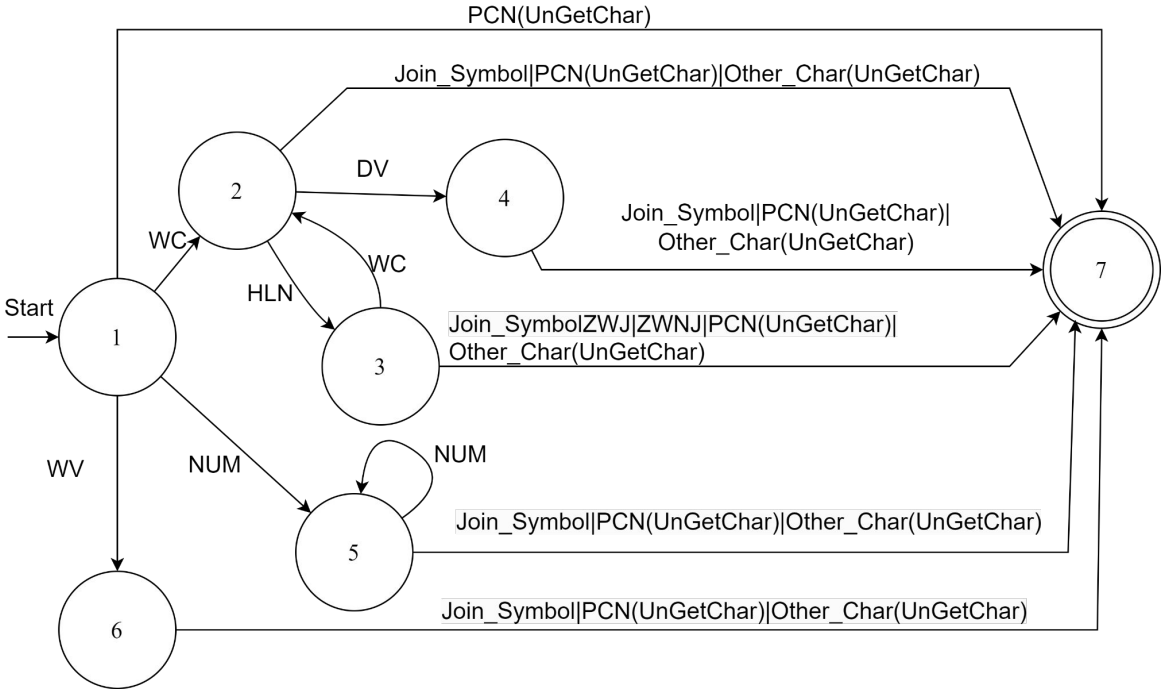


Fig: Finite State Diagram for Tokenizer

METHODOLOGY - [4]

Syllable Tokenizer

Tokenization and Labeling

- Each character is a token
- A character fall into a category
- A word is broken into syllable
- Further processing is done on syllable basis.

सिञ्जा

Character	स	ि	ञ	्	ज	ा
Category	WC	DV	WC	HLN	WC	DV
Syllables	WC_DV		WC_HLN_WC_DV			

METHODOLOGY - [5]

Reordering Rules

- After tokenization, a buffer of glyphs reordered is maintained.
- Each glyph is tagged according to its position in the syllable.
- All syllables may not contain all types of glyphs.

METHODOLOGY - [6]

Reordering Rules

- The reordering is done in following steps:

1. Find base consonant from syllable.
2. Decompose and reorder matras.
3. Final reordering:
    a. Reorder prebase matra.
    b. Reph reorder.
    c. Reorder prebase consonants.

METHODOLOGY - [7]

Reordering Rules

- Example : सिञ्जा

Syllable	सि		ञ्जा			
Characters	स	ि	ञ	्	ज	ा
Tokenization	WC_DV		WC_HLN_WC_DV			
Reordering	ि	स	ञ	्	ज	ा
Position	POS_PREBASE_MATRA	POS_SYLLABLE_BASE	POS_PREBASE_CONSONANT		POS_SYLLABLE_BASE	POS_POSTBASE_MATRA

METHODOLOGY - [8]

Font File

- A OpenType unicode font is used.
- Contains information about shape of glyphs and lookup tables (GPOS, GSUB)
- GSUB table with substitution features
- GPOS table with positioning features

examples: Mangal.otf, Kalimati.otf

METHODOLOGY - [9]

GSUB Features

- Substitution features of GSUB table of font file is used
- The order of substitution is based in OpenType shaping

Name	Example	Substituted glyphs
Akhanda	ज + ◌् + ञ	ज्ञ
Reph	र + ◌् + क	र्क
Rakaar	भ + ◌् + र	भ्र
Half form	क + ◌् + ख	क्ख

METHODOLOGY - [10]

Inside GSUB Table

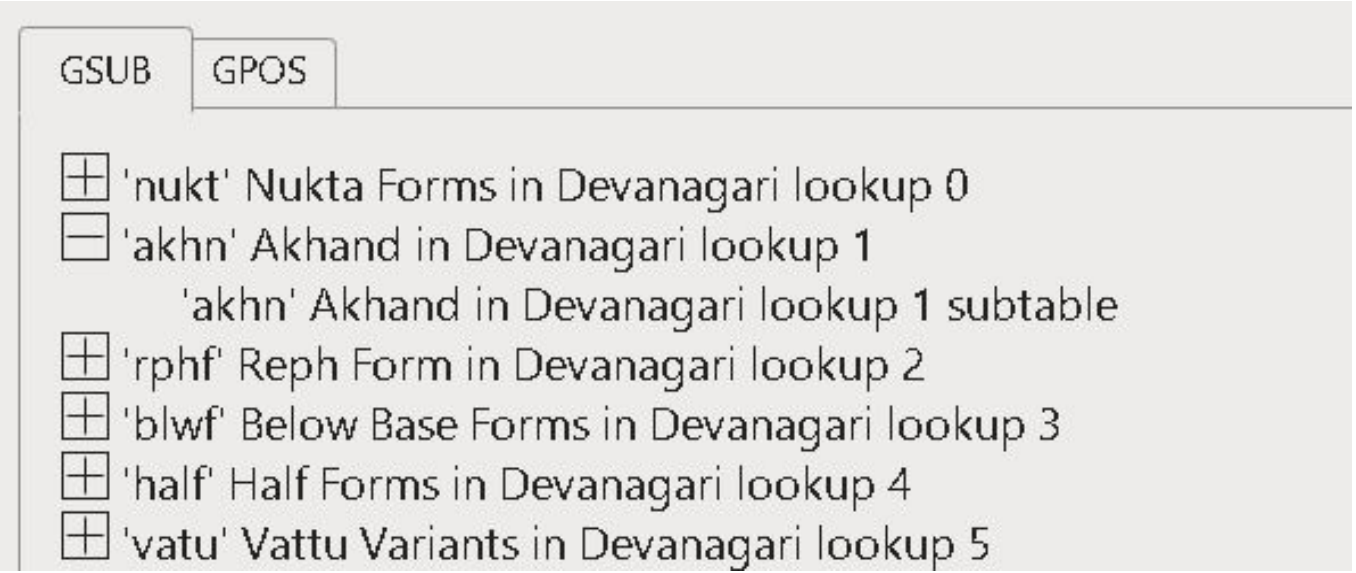


Fig: Structure of GSUB Table

METHODOLOGY - [11]

GPOS Features

- Manage the positions of glyphs relative to each other.
- Includes adjustments for ligatures, kerning, and diacritics.
  example: adjustment of ligatures with dependent vowels.

METHODOLOGY - [12]

Inside GPOS Table

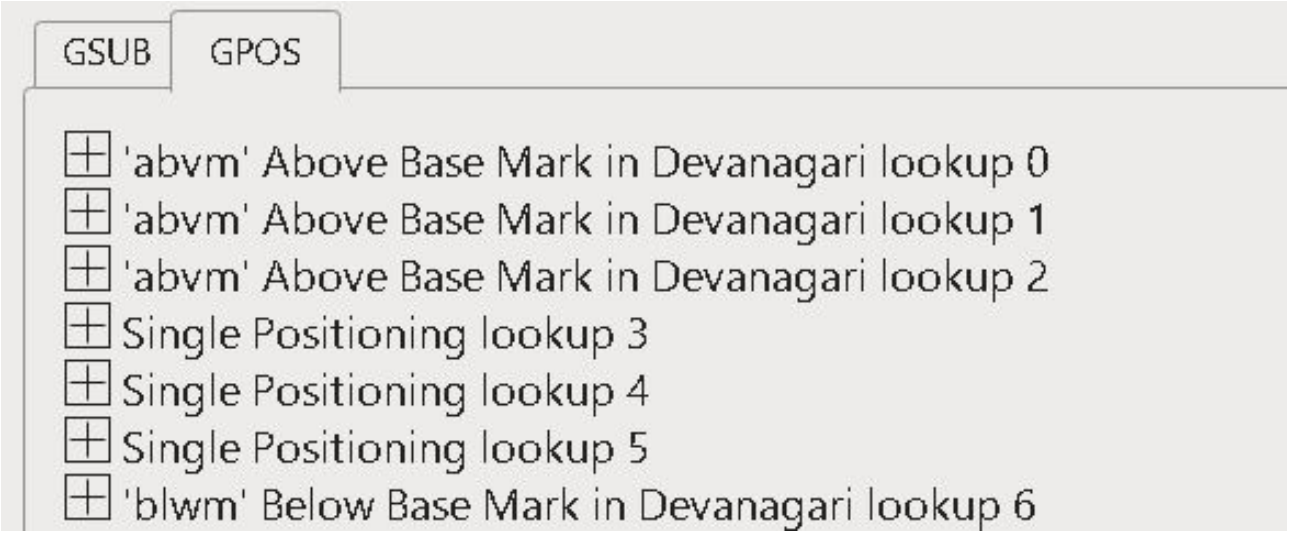


Fig: Structure of GPOS Table

METHODOLOGY - [13]

PDF Generation

- PDF layouts are defined.
- Correctly reordered and shaped glyphs/ligatures are supplied for pdf generation.

PDF File

- It is the final output of the system
- Contains the devanagari text

Correctness measure

$$\text{Accuracy} = \frac{\text{Correctly ordered glyphs}}{\text{Total words}}$$

$$\text{Average accuracy} = \frac{\sum_{i=1}^{k} \text{Accuracy (Category)}_i}{k}$$

where, k = Number of categories

Categories by length of sequence: Uni, Bi, Tri, Quad etc.

METHODOLOGY - [15]

Correctness measure for reordering algorithm

सिञ्जा

Input : { स, ि, ञ, ् , ज, ा }

Label : { ि, स, ञ, ् , ज, ा }

Output after applying reordering rules:

{ ि, स, ञ, ् , ज, ा }

This is compared with label.

RESULT- [1]

Syllable Tokenizer

Word	Syllables	Decomposition
स्त्री	1(uni)	स्त्री {WC_HLN_WC_HLN_WC_DV}
चिट्ठी	2(bi)	चि{WC_DV}, ट्ठी{WC_HLN_WC_DV}
गतम्	3(tri)	ग{WC}, त{WC}, म्{WC_HLN}
कम्प्युटर	4(quad)	क{WC}, म्प्यु{WC_HLN_WC_HLN_WC_DV}, ट{WC}, र{WC}
उपमहानगरपालिका	10(ten)	उ{WV}, प{WC}, म{WC}, हा{WC_DV}, न{WC}, ग{WC}, र{WC}, पा{WC_DV}, लि{WC_DV}, का{WC_DV}

Test Data Preparation

RESULT- [2]

uni	bi	tri	quad	penta	hexa	hepta
कि	अँख्या	अँखडि	अँगालिनु	अँगारधर्मी	अँध्यारखाउडे	अक्रमातिशयोक्ति
क्या	अँट्वा	अँगर्खा	अँचेटिनु	अँधेरीपक्ष	अँध्यारखाउडो	अङ्कपरिवर्तन
क्यु	अक्का	अँगाल्नु	अँठ्याउनी	अँध्यारमुखे	अंशविषयक	अङ्गप्रतिरोपण
क्यू	अक्को	अँगिया	अँठ्याउनु	अंशविहीन	अकमक्याउनु	अङ्गप्रत्यारोपण
क्ले	अक्खा	अँगुच्छा	अँधेरिनु	अंशसर्वस्व	अकल्याणकारी	अतिक्रमणकारी

RESULT- [3]

Test Data Size

Number of Syllables	Count	Sample data
1	110	स्त्री
2	9981	चिट्ठी
3	22153	गतम्
4	27048	कम्प्युटर
5	11508	जलविद्युत्
6	3682	स्वदेशीकरण

Number of Syllables	Count	Sample data
7	597	पदपरिवर्तन
8	119	आवश्यकतापूर्ति
9	21	अन्तरराष्ट्रियकरण
10	2	उपमहानगरपालिका
Total	75221	

DISCUSSION OF RESULT - [1]

- Syllable tokenizer groups the characters of a words into syllable.
- Reordering rule is applied to each syllable separately.
- Wordlist for different sequence length can be tested.
- Error analysis may be need, for better coverage.

LIST OF REMAINING TASK

Labeling of Test Data

- Manually reorder the characters of test sequences

Implementation of algorithm

- Actual coding in the Apache PDFBox codebase

System evaluation

- Testing the reordering algorithm for the test data

REFERENCES - [1]

- Microsoft, "Microsoft Typography Documentation," Accessed: Jul. 19, 2024. [Online]. Available: https://learn.microsoft.com/en-us/typography/script-development/devanagari
- U. Consortium et al., “The unicode standard, version 14.0. 0–core specification,” 2021.
- S. B. Basnet and S. Trishna, “Unification of fonts encoding system of devanagari writing in nepali,” Nepalese Linguistics, vol. 32, no. 2, pp. 130–136, 2017.

REFERENCES - [2]

- M. Boualem, M. Leisher, and B. Ogden, “Encoding script-specific writing rules based on the unicode character set.”
- S. P. Mudur, N. Nayak, S. Shanbhag, and R. Joshi, “An architecture for the shaping of indic texts,” Computers & Graphics, vol. 23, no. 1, pp. 7–24, 1999.
- K. Nepal, "Nepali Font Standards," Kathmandu, Nepal.