Converting Text to Unicode

Abstract:

Sometimes text is available in font based encoding means the text is not in readable format. Such texts cannot be run through Machine Translation, Search engines, Dialogue systems etc. For the different languages in the world including Indian, there is a forum available called “Unicode, Inc.” which provides a solution to the localization problem of the world’s languages.

The goal here is to develop a converter which takes the different font encodings and convert into Unicode format. And the converter should handle the different font styles with in a file, like one paragraph in one font encoding and other in different font encoding.

We can solve this problem with different approaches like rule based approaches, machine learning based approaches and optical character recognition based approaches. In this we are presenting only rule based approach.

Related Work:

Unicode as a multilingual standard with reference to Indian languages discussed about how the Unicode system is useful to records of indian complete culture, secret manuscripts and related documents of the respective religions. These documents are written 3000 years ago and they didn’t followed any standard font encoding. At the time of automation in this documents the absence of proper standards, professionals tried to romanize documents as convert into Unicode to accept computers.

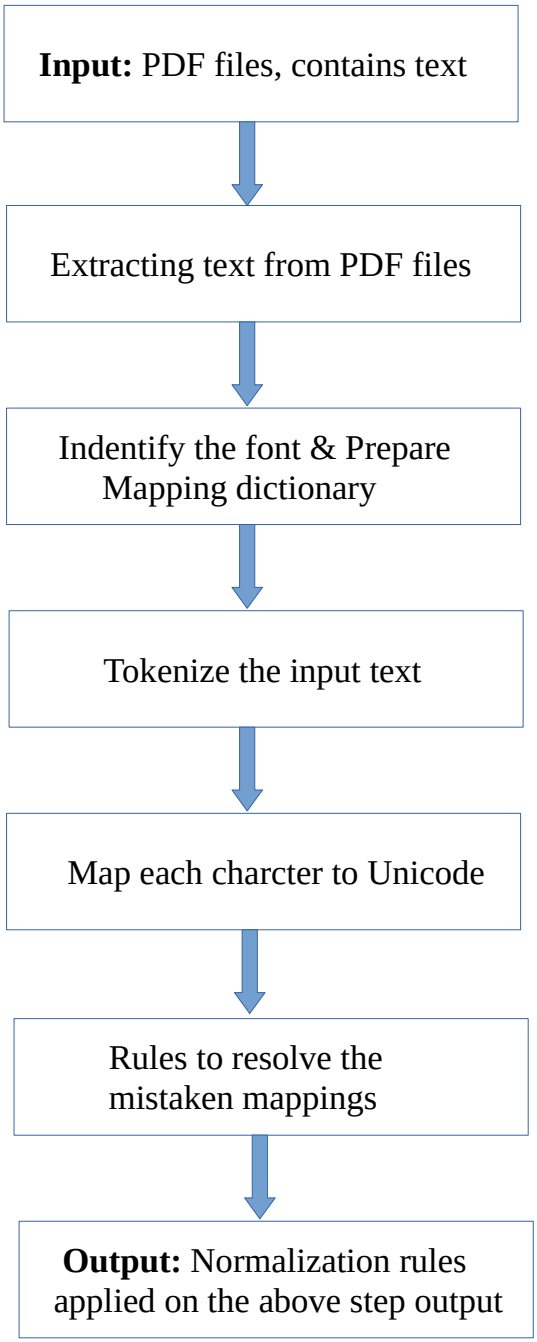
Multilingual conversation ASCII to Unicode in Indic Script paper describes about various ASCII based scripts that are made for Indian languages and the problems associated with these types of scripts. Then discuss the solution that suggest to overcome these problems in the form of “Multilingual ASCII to Unicode Converter”.

Data Set:

We have around 15 to 20 PDF files with SHREE, Chanakya font encodings. A single PDF file contains more than one font encoding, means headings in one font encoding and the paragraphs in different font encoding.

First we converted these PDF files into txt files. Each PDF file contains approximately 8000 lines of text.

Approach:



Mappings:

We created a mapping dictionary. It contains the mapping from glyph in specific font encoding to Unicode in Devanagari. From SHREE701 we extracted 220 glyphs. Created the mapping dictionary for this glyphs. The unicode range of SHREE glyphs is U0020 to U02C7. Below shown some example glyph to unicode mappings

Unicode of Glyph	Glyph	Devanagari Unicode
U+0046	F	छ्
U+0047	G	ि
U+0048	H	ँ
U+0049	I	क
U+004A	J	ल
U+004B	K	फ्
U+004C	L	ण
U+004D	M	ॅ
U+004E	N	ु
U+0050	P	स
U+0051	Q	द्ध
U+0052	R	ू

Modules Explanation:

- The input to our system is a set of books which are looking like Devanagari script in front end, but in backed with different font encodings.

- Extract the text from PDF files using **pdftotext** application. This extraction still gives font encoded text only. We need to covert this text into Unicode.

- Manually identified the font encodings in PDF files, those encodings are SHREE701, Chanakya, KruthiDev, Avanthi, Times New Roman and etc. But most of the PDF files contains the Chanakya encodings as headings and SHREE encodings as body.

- Created the mapping dictionaries for different font encodings. Each dictionary contains a character symbol, corresponding glyph in that specific font encoding and corresponding Devanagari Unicode for that glyph.

- Take the text, tokenize by word level and tokenize each word by character level. By using above mapping rules map each character to it's corresponding Unicode in Devanagari. By this step will get just unicode mapped output. In this output will get lot of wrong mappings. By writing some additional mapping rules we resolve the wrong mappings. And we wrote the some normalization rules also to overcome wrong mappings.

Additional Rules:

- ि करण → किरण
  In above word the ि is attaching to previous character. So we need to move that halant till we get a full character/consonant.

- दर्शन → दर्शन
  The र् matra attaching to previous character, so we need to backward the र् matra till we get a consonant.

- सावभ ्रौम → सार्वभौम
  Same like above, but here we have two problems. One is with र् matra another is ौ matra. We need to move the र् matra till get a consonant and move forward this ौ till we get a consonant.

- राष्ट ीयता → राष्टीयता
  The ी matra move backward till we get a consonant.
  → **And so more rules also there.**

Normalization Rules:

- For आ, its giving अ **(Unicode)** + ◌ा **(Unicode)** but its not correct type of rendering. Whenever will get अ + ◌ा, directly replace with आ.
- Whenever will get अ + ◌ा + ◌ै, normalize it with औ.
- If we get ◌ा + ◌ै replace with ◌ौ **(Unicode)**
- Whenever will get आ + ◌े + ◌ं replace with ओं **(Unicode)**
- If we get a character ए + ◌े replace with ऐ
- **..etc**

Sample Results:

Input	Output
Ó«‡hÒ £~∂D P´¿„h»œÎÎb£ ±Ÿ ¿bµbGŸ»	मध्यस्थ दर्शन सहअस्तित्ववाद पर आधारित
<Î•Te±	विकल्प
yÎ}	एवं
¿«‡‡D <kE£È	अध्ययन बिन्दु
±–LC»b yÎ} JC⁄•T	प्रणेता एवं लेखक
y. Db≥ŸbÆ	ए. नागराज
Ó«‡hÒ £~∂D P´¿„h»œÎÎb£	मध्यस्थ दर्शन सहअस्तित्ववाद
±–•Tb~•T #	प्रकाशक :
ÆaÎD <ÎÀb ±–•Tb~D	जीवन विद्या प्रकाशन
<£—‡±Ò P}hÒbD	दिव्यपथ संस्थान
¿ÓŸ•}T^>•T, <ÆJb ¿DØ±±NŸ - îòîòòñ Ó.±–. BbŸ»	अमरकंटक, जिला अनूपपुर – 484886 म.प्र. भारत
y. Db≥ŸbÆ	अमरकंटक, जिला अनूपपुर – 484886 म.प्र. भारत
PÎb∂<µ•TbŸ ±–LC»b yÎ} JC⁄•T •CT ±bP PNŸ<[»	ए. नागराज
P}h•TŸL # íêëë	सर्वाधिकार प्रणेता एवं लेखक के पास सुरक्षित
ÓNæL # ëî ÆDÎŸa íêëñ	संस्करण : 2011
P´‡bC≥ Ÿb<~ # ïê/- ı±‡C	विकल्प अस्तित्व मूलक मानव केन्द्रित चिन्तन है ।
ÆbD•TbŸa #	मध्यस्थ दर्शन–सहअस्तित्व में, से, के
P£È±‡bC≥ Da<» #	लिए मानव का अध्ययन संभव हो गया है ।
•Tb ¿«‡‡D P|±ED D´a* ´È¿b $ ‡´ PØ™Db £C»C ´Èy ±–P¤»b •Tb ¿DNBÎ •TŸ»b ´RH G•T	स विकल्प में आपको अवगत कराने का प्रयत्न है
<Jy ÓbDÎ •Tb ¿«‡‡D P}BÎ ´bC ≥‡b ´º $	कि मानव का अध्ययन मानव

Error Analysis:

- We compare the 10 pdfs and output data side by side, most of the input data converted correctly.
- For some glyphs in SHREE701, there s no corresponding unicode symbol. Those glyphs are deleted in current ttf file but the pdf files are written based on old ttf file. This is one issue.
- In input data, some spaces are misplaced because of this we are getting some errors.

Roll Numbers

2018701021
2018701022