A SURVEY OF LANGUAGE-DETECTION, FONT-DETECTION AND FONT-CONVERSION SYSTEMS FOR INDIAN LANGUAGES

Preetpal Kaur Buttar[1] and Jaswinder Kaur[1]

[1]Department of Computer Science and Engineering, Sant Longowal Institute of Engineering and Technology, Longowal, Punjab, India

ABSTRACT

A large amount of data in Indian languages stored digitally is in ASCII-based font formats. ASCII has 128 character-set, therefore it is unable to represent all the characters necessary to deal with the variety of scripts available worldwide. Moreover, these ASCII-based fonts are not based on a single standard mapping between the character-codes and the individual characters, for a particular Indian script, unlike the English language fonts based on the standard ASCII mapping. Therefore, it is required that the fonts for a particular script must be available on the system to accurately represent the data in that script. Also, the conversion of data in one font into another is a difficult task. The non-standard ASCII-based fonts also pose problems in performing search on texts in Indian languages available over web. There are 25 official languages in India, and the amount of digital text available in ASCII-based fonts is much larger than the text available in the standard ISCII (Indian Script Code for Information Interchange) or Unicode formats. This paper discusses the work done in the field of font-detection (to identify the font of the given text) and font-converters (to convert the ASCII-format text into the corresponding Unicode text).

KEYWORDS

Language Detection, Font Detection, Font Conversion, Unicode

1. INTRODUCTION

1.1. Need for Language Identification

The task of automatic processing of digital texts is dependent on the language or the script in which the text is written. Moreover, the texts available over the web are becoming more and more multilingual. In order to deal with such multilingual texts, we need tools to automatically identify which portions of a document are written in which language.

1.2. Need for Font Identification

The font-format specifies the mapping between the characters (or other units) of a script and the character codes in a range (say, 0 to 255). The tasks of language identification and font-format identification are closely related to each other, because a specific language has a specific set of font-formats. The Indian font-formats are not compatible to each other as they are not based on a single standard mapping. Most of these fonts are proprietary fonts. Taking the example of Devanagari alone, there are more than 55 popular Devanagari font-formats and 5 different keyboard-layouts. If the specific font is not available, a document composed with any of these fonts cannot be read in MS-Word or even in any other application. This problem can be handled

DOI: 10.5121/ijci.2015.4302

by automatically identifying the font-format of the text and then converting the text written in thus identified font-format into a text written in a standard font-format such as Unicode.

1.3. Need for Font Conversion

A lot of electronic texts in Indian languages were made and available in the ASCII-based fonts (legacy fonts). There is a need to convert these non-portable, non-standardised texts into a portable format, so that they can be read/written/edited anywhere, anytime. The text in ASCII-based font formats can be converted into ISCII or Unicode text automatically with the help of font-conversion tools. The standardisation of text also solves the problem of searching text over the web.

2. Related Work on Language and Font Identification

One of the first automatic text processing problems for which a statistical approach was used was that of language identification. In the earlier stages of language identification, the approach of identifying unique strings, which were the characteristics of a particular language, was used. These unique strings were then matched with the test data to identify its language. They were called 'translator approaches'.

Beesley's [1] automatic language identifier for on-line texts used mathematical language models originally developed for breaking ciphers. These models exploited the orthographic features such as the unique characteristic strings of a particular language and frequencies of strings for each language.

Canvar and Trenkle [2] used the method of N-gram based text categorization in the N-gram frequencies of language corpus are calculated and then compared with that of test data to find a match. Cavnar also proposed the pruned N-gram approach i.e. the top 300 or so N-grams give the probability that the test data is correlated with the language, while the lower ranked N-grams give an indication of the topic of the test data. Many approaches similar to Cavnar's have been tried, the main difference being in the distance measure used.

Giguet [3] tried to categorize multilingual sentences based on the understanding of the linguistic properties of the text, mainly the knowledge about the alphabet, word morphology, grammatically correct words etc.. This method does not rely upon the training sets. The output of applying this method was the sentences in a document tagged with the language name.

Kikui [4] used Canvar's method along with some heuristics to identify languages as well as encodings for a multilingual text. He used statistical language model and mappings between languages and encodings.

The work done in the field of language and font-encoding identification for the Indian languages is listed as under.

2.1 Mutual Cross Entropy based Language and Encoding Identification

Singh [5] used character based N-grams and noted that the language identification problem is that of identifying both language and encoding as the same font-encoding can be used for more than one language (for example, ISCII for all Indian languages which use Brahmi-origin scripts) and one language can have many encodings (for example, Unicode, ISCII, typewriter, phonetic, and many other proprietary encodings for Hindi). He conducted a study on different distance measures. He has proposed a novel measure using mutual entropy resulting improved performance better than other measures for the current problem. He also conducted some experiments on using such similarity measures for language and font-encoding identification.

2.2 TF-IDF based Font Identification

Raj and Prahallad [6] discuss the issues related to font encoding identification and font-data conversion in Indian languages. TF-IDF (Term frequency and Inverse document frequency) weights approach is used for identification of font-encoding, where TF means the number of times a glyph appears in a text, and IDF means the numbers of text documents in which the given glyph has appeared once or more number of times. A combination of unigrams (current glyph), bigrams (current and next glyph) and trigrams (previous, current and next glyph) has been used and weightage has been given to them on the basis of their frequency of occurrence in the text. The system supports font-identification for several languages like Hindi, Punjabi, Tamil, Guajarati, Kannada, Malayalam, Oriya, Bengali etc.

2.3 Intelligent Legacy-font to Unicode Converter

Lehal et. al. [7] describe a language- and font-detection system, and font-conversion system for Gurmukhi and Devanagari. According to the developers, this system provides support for the automatic font-detection and conversion of the text written in around 150 ASCII-based fonts of Gurmukhi script and more than 100 ASCII-based fonts of Devanagari script. They used a statistical N-gram language model based on words as well as characters, for the purpose of automatic identification of language as well as font-encoding. The system has been trained on a number of fonts, by calculating the trigram frequencies of texts written in the fonts to be trained. The input text is analyzed for similarity with the trained fonts and the font having the highest weightage is selected as the detected font. For the purpose of conversion of the detected font of the input text to Unicode, the mapping tables for all the trained fonts have been built which map the font glyphs to the corresponding Unicode characters. Then, the rule based approach is followed for the conversion of input text to Unicode text.

3. RELATED WORK ON FONT CONVERSION

Several attempts were made at font conversion. While many focused on text data, few were on font data embedded in images.

3.1 Finite State Transducer (FST) based Font Converter

Chaudhury et al. [8] propose an approach for font conversion of Oriya language which follows a rule based approach. The generation of font-converter occurs in the following three steps:

- A mapping table is created for the ASCII-based Oriya fonts, which maps the most commonly occurring syllables in Oriya to their corresponding Unicode font code.
- Then, a rule based approach is used which defines the rules to map the ASCII-based font codes to the corresponding Unicode characters. The rules have been defined using a dictionary.
- Then, a Flex scanner is designed to handle the unhandled symbols that appear in the output from the previous step.

The final converted text thus obtained is quite accurate but not absolute.

3.2 Semi-automatic converter for conversion from legacy-font encoding to standard ACII encoding

Bharati et al. [9] propose a font-converter for Devanagari script which supports the conversion of text in written in an unknown font-encoding to standard ACII (Alphabetic Code for Information Interchange) font-encoding, either automatically or semi-automatically. The system takes:

- A text written in an unknown font-encoding, and,
- The same text transliterated in the ACII font-encoding

Given these inputs, a font-converter is generated between the given font-encoding and ACII. The font-converter thus generated can be now used to convert a text from that unknown non-standard font-encoding to ACII, and vice-versa. The converter can later be refined manually also. A glyph-grammar for the script of the language is also needed for developing the font-converter, which specifies what possible glyph sequences make up an Akshara. This grammar is structured and is independent of the various font-encodings. It needs to be developed only once, for a script. The glyph-grammar can have three types of rules: Type 1 rules are applicable across all the Indian languages like schwa (‘a’ sound) deletion from an Akshara, Type 2 rules hold for Devanagari, and perhaps for other scripts. Type 3 rules pertain to idiosyncratic combination of glyphs for that particular style. The limitation of this approach is that the performance of the system depends on the quality and the amount of training data in such a way that some glyphs might remain unconverted because the training data might not contain complete information, and the program might not have seen some glyphs as they do not occur in this learning data. This system requires manual intervention at the end of the conversion process to supply the missing code map and the rules (by looking at the remaining part of the glyph file and making intelligent guesses) using which the converter would become complete. The overall converter is able to convert 90 to 95% of the glyphs with a page or two of sample ACII text.mat must not exceed twenty (20) pages in length.

3.3 TBIL Data Converter [10]

The TBIL Data Converter has been developed by Microsoft for Bhasha project. This software supports 9 Indian languages, Gujarati, Hindi, Bengali, Tamil, Telugu, Malayalam, Punjabi, Kannada and Marathi. It supports the conversion of non-Unicode text written in any of these languages into its equivalent Unicode text in that language, and vice-versa. The user has to specify the language and the font-encoding of the input text. It supports document formats such as .txt, .doc, .docx, .xls, .xlsx, .mdb, .accdb and SQL database.

3.4 DangiSoft (Font Converter) [11]

This converter is developed by Er. Jagdeep Dangi. It consists of two parts. The first part is PrakharDevanagari Font Parivartak. It is also called ASCII/ISCII to Unicode Converter. The second part is called UniDev.

3.4.1 PrakharDevanagari Font Parivartak (ASCII/ISCII to Unicode Converter)

This software has been developed to convert the input text in various ASCII/ISCII based Devanagari (Hindi, Marathi and Sanskrit languages) font-encodings into the equivalent Unicode text. This software supports the conversion of text available in more than 258 true type and type-1 Devanagari ASCII-based font-encodings into Unicode text. Some of the popular fonts supported by this software are KrutiDev, Chanakya, Shusha, Shiva, DV-TTYogesh, 4CGandhi, Sanskrit 99, Marathi-Kanak etc.

3.4.2 UniDev

The second part of the converter, UniDev, has been developed to convert the digital text in Unicode (Mangal) based font to its corresponding text in various ASCII/ISCI fonts like KrutiDev, Chanakya etc. for Devanagari Script. This software is useful for those users who use softwares like Corel Draw, Photoshop, PageMaker, Quark Express etc. as these softwares are not capable of supporting the Unicode text.

3.5 ePandit IME [12]

The ePandit font-converter has been developed by Mr. Shirish Benjwal Sharma. The software provides the facility to convert the input text written in ‘Chanakya’ font to its equivalent text in Devanagari Unicode font. There is also an option available for converting Devanagari Unicode text into equivalent text in non-Unicode ‘Chanakya’ font. This tool is specially used for Corel Draw, PageMaker, etc. non-Unicode programs which do not type Unicode Hindi. This software is freely available online.

3.6 Google’s Converters-Scientific and Technical Hindi [13]

This page lists different font changers developed by the Google group. Google provides browser-based font-converter programs which run in the web browser. These programs are portable, thus can be used on any operating system. They can also be saved and used offline. A few examples are: Chanakya font to Unicode converter, KrutiDev font to Unicode converter, Unicode to Agra font converter, etc.

3.7 Gurmukhi Unicode Conversion Application [14]

GUCA has been developed for converting the ASCII font-based Gurmukhi text into its equivalent Unicode text. This software can be used for converting Gurmukhi text based on Dr. Thind's fonts (e.g. AnmolLipi, GurbaniLipi fonts) into the equivalent Unicode text. The conversion is based on the mappings between the ASCII-based fonts and the Unicode font. The software also includes a custom mapping engine using which one can easily add their own font mappings. The software provides the conversion facility between various fonts such as Dr. Chatrik Web to AnmolLipi and Unicode, GurLipi to AnmolLipi and Unicode, Punjabi font to AnmolLipi and Unicode, Satluj to AnmolLipi and Unicode. This software is open-source and has been released under the GNU General Public Licence.

3.8 Online Punjabi Font Converter [15]

The online Punjabi font-converter is a tool for performing automatic font conversion among a number of Punjabi true type fonts. This software is freely available online. It can convert Punjabi text written in any true type Punjabi font to its equivalent text in another true type Punjabi font or Gurmukhi Unicode font. Some of the major Punjabi fonts supported by this software are Akhar, DRChatrikWeb, WebAkhar, GurbaniAkhar, GurbaniLipi, Gurmukhi, ISCII, Joy, Punjabi, Satluj etc.

3.9 Akhar Font Conversion [16]

Different Punjabi fonts have different keyboard layouts, so the inbuilt font conversion utility of Windows cannot be used to convert the font of a Punjabi text. Akhar provides the inbuilt facility of converting text in Punjabi from one font to another font without any loss of text or formatting information. The font conversion utility supports more than one hundred twenty commonly used Punjabi Fonts and 37 different Keyboard Layouts for Punjabi.

4. Summary of the Literature Review for Major Indian Font-detection and Font-conversion Systems

4.1 Summary of the Language and Font-detection Systems

The following table describes the summary of major Indian font-detection systems and their salient features.

Table 1. Summary of language and font-detection systems

S. no.	Name of the font-detection system	Language(s) supported	Salient features
1.	Language and font-encoding identification, Singh	The languages considered ranged from Esperanto and Modern Greek to Hindi and Telugu	- Detects language as well as encoding of the text - Based on pruned character *n*-grams, alone as well word *n*-grams - The detection accuracy was 94-100%
2.	Font-encoding identification and font-data conversion, Raj and Prahallad	Hindi, Punjabi, Tamil, Guajarati, Kannada, Malayalam, Oriya, Bengali	- TF-IDF (Term Frequency and Inverse Document Frequency) weights used for font identification - Glyph assimilation rules for font-data conversion - The detection and conversion accuracy of the system is 92-100%
3.	Intelligent Legacy-font to Unicode Converter	Hindi, Punjabi, Marathi, Sanskrit	- Statistical trigram language model used for automatic font-detection as well as language-detection - N-gram based text similarity method for font-conversion - Font detection accuracy is 99.6% and Unicode conversion accuracy is 100%

4.2 Summary of the Font-conversion Systems

The following table describes the summary of major Indian font-conversion systems and their salient features.

Table 2. Summary of font-conversion systems

S. no.	Name of the font-conversion system	Language(s) supported	Salient features
1.	Font-conversion of Oriya language	Oriya	- Rule-based approach is used - Supports the conversion of only two proprietary fonts, 'Sambad' and 'Akruti' to standardized Unicode font - Conversion accuracy is 72%
2.	Font-converter for Devanagari Script	Hindi, Marathi and Sanskrit (i.e. Devanagari script)	- Performs conversions between the given unknown coding scheme of Devanagari and rhe standardized ACII coding scheme - Requires manual interference - Conversion accuracy is 90-95%
3.	TBIL Data Converter	Bengali, Gujarati, Hindi, Kannada, Malayalam, Marathi, Punjabi, Tamil and Telugu	- Conversion between data in legacy-font/ASCII/Roman format in Office documents into a Unicode form - Conversion accuracy is 80-85% - Supports various document formats such as .txt, .doc, .docx, .xls, .xlsx, .mdb, .accdb etc.
4.	DangiSoft Font-converter	Hindi, Marathi and Sanskrit (i.e. Devanagari script)	- Supports ASCII/ISCII to Unicode conversion and vice-versa - Conversion accuracy is 100%
5.	e-Pandit Converter	Hindi	- Supports ASCII-fonts to Unicode conversion and vice-versa - Supports conversion between Unicode and a limited number of ASCII-fonts like Chanakya, KrutiDev and Walkman-Chanakya etc. - Conversion accuracy is 85-90%
6.	Google's Converters-Scientific and Technical Hindi	Hindi	- Web-browser based converter programs - Supports ASCII-fonts to Unicode conversion and vice-versa - Conversion accuracy is 80-90%
7.	Gurmukhi Unicode Conversion Application	Punjabi	- Converst ASCII encoded, font-based Punjabi (Gurmukhi script) text into Unicode - Open-source - Conversion accuracy is 90%
8.	Online Punjabi Font Converter	Punjabi	- Converts Punjabi text in any true type Punjabi font to its equivalent in another font or Gurmukhi Unicode - Conversion accuracy is 85-90%

5. Conclusions

The paper described the language-detection, font-detection and font-conversion techniques in context of the Indian languages. Mostly, the language and font identification systems are based on the statistical approaches, such as the character-based n-grams and the word-based n-grams. For the purpose of font conversion, mapping tables are used, which map the legacy-based font glyphs to the corresponding Unicode characters. On this metadata, either rules have been defined or the machines have been trained to learn using some parallel corpora.

6. Future Work

Although the existing systems are already demonstrating good performance, there is considerable room for further work. The approach used for identification of language and font-encoding can be extended to handle another important issue of multilingualism of the documents, specially the Web pages, which may contain text in more than one language and font-encoding. An ideal language and font-encoding identification tool should be able to mark which parts of the page are in which language and font-encoding. It will be interesting to apply the methods used for language and font identification for the purpose of text categorization or topic identification and other related problems.

References

[1] K. Beesley, “Language identifier: A computer program for automatic natural-language identification of on-line text,” in *Languages at Crossroads: Proceedings of the 29th Annual Conference of the American Translators Association*, October 1988, pp. 47-54.

[2] W. B. Canvar, and J. M. Trenkle, “N-gram-based text categorization,” in *Proceedings of SDAIR-94, 3rd Annual Symposium on Document Analysis and Information Retrieval*, Las Vegas, US, 1994, pp. 161-175.

[3] Emmanuel Giguet, “Multilingual sentence categorization according to language,” in *Proceedings of the European Chapter of the Association for Computational Linguistics, SIGDAT Workshop, From Text to Tags: Issues in Multilingual Analysis*, Dublin, Ireland, March 1995, pp. 73-76.

[4] G. Kikui, “Identifying the coding system and language of on-line documents on the internet,” in COLING, 1996, pp. 652-657.

[5] Anil Kumar Singh, “Study of some distance measures for language and encoding identification,” in *Proceedings of the Workshop on Linguistic Distances*, Sydney, Australia, July 2006, pp. 63-72.

[6] Anand Arokia Raj, and Kishore Prahallad, “Identification and conversion of font-data in Indian languages,” in *International Conference on Universal Digital Library (ICUDL2007,)*, Pittsberg, USA, November 2007.

[7] G. S. Lehal, T. S. Saini, and P. K. Buttar, “Automatic bilingual legacy-fonts identification and conversion system,” in *Research in Computing Science*, Volume 86, pp. 9-23 (2014).

[8] Sriram Chaudhury, Shubhamay Sen, and Gyan Ranjan Nandi, “A Finite State Transducer (FST) based converter,” in *International Journal of Computer Applications*, Volume 58, No.17, November 2012, pp. 35-39.

[9] Akshar Bharati, Nisha Sangal, Vineet Chaitanya, Amba P. Kulkarni, and Rajeev Sangal, Generating converters between fonts semi-automatically,” In *Proceedings of SAARC conference on Multi-lingual and Multi-media Information Technology*, CDAC, Pune, India, September 1998.

[10] “TBIL data converter,” Microsoft Corporation, [Online]. Available: http://www.bhashaindia.com/Downloads/Pages/Home.aspx.

[11] “DangiSoft India,” [Online]. Available: http://www.4shared.com/u/5IPlotQZ/DangiSoft_India.html.

[12] “ePandit,” [Online]. Available: http://epandit.shrish.in/labs/en/epanditime/

[13] “Scientific and technical Hindi,” Google Sites, [Online]. Available: http://sites.google.com/site/technicalhindi/home/converters/

[14] “Gurmukhi Unicode conversion application,” [Online]. Available: http://guca.sourceforge.net/applications/guca/

[15] “Punjabi font converter,” [Online]. Available: http://punjabi.aglsoft.com/punjabi/converter/?show=text

[16] "Akhar font conversion,” [Online]. Available: http://www.akhar.org/Font_Conversion.htm