

Font Recognition for Indian Language Document Viewing

Sunayana Sitaram

Computer Engineering Department,
Sardar Vallabhbhai National Institute of Technology
Surat, India – 395007
sunayana.sitaram@gmail.com

The Problem

An English document written using, for example, the Arial font remains readable even if we view it using another font like Verdana; an Indian language document, on the other hand, makes no sense unless we use the right font. The problem arises because the encoding of the characters in different fonts for the same Indian language is different and no standard is adhered to [1]. This makes the exchange of documents difficult and implementing searching, spell checking and dictionary maintenance and text-to-speech systems difficult to implement in Indian languages. In all these cases, either the document has to use a “standard” font or the document user has to use a font converter to convert the document into a standard font. Simple font converters for some Indian languages are available on the internet [2] but this method works only when the user knows what font the document has been created in. Such knowledge would not be available to a visually impaired reader or to a reader of a text document.

The aim of this paper is to devise a system that automatically determines which of a given set of fonts a Gujarati document uses and then converts the document into a standard Gujarati font for viewing.

The Solution

The font recognition problem was looked at as a problem of solving a Random Substitution Cipher. Experiments on documents were carried out in two phases using slightly different approaches to the solution.

Phase 1

In phase 1 of the experiments, I chose documents using one of four popular Gujarati fonts available on the internet – Krishna, ITXGuj, Gopika and GujaratiLys. Single character frequency analysis was used to recognize the font. For this, a frequency distribution table for Gujarati characters was created by performing frequency analysis on a corpus of 64,000 characters all in the Krishna font.

I used the Krishna frequency table and font conversion tables in order to arrive at standard frequency distributions for the other three fonts. The Euclidean distance was used as a similarity measure between the standard frequency

distributions and the distribution obtained from the document to determine which of the four known font frequency distributions the unknown document’s frequency distribution was closest to.

Results of Phase 1: Experiments were carried out on 26 documents of varying sizes from 2 KB to 6 KB and using different fonts. The results are shown in Table 1.1. The fonts of 24 of the documents were recognized correctly. Of the two errors, in one case, the document was very short and in another, the actual font and the recognized font were very close to one another. A problem I faced in phase 1 while arriving at frequency distributions for the other three fonts was that some characters which were present in the Krishna font were missing in the other three fonts and vice versa. In phase 1, I assigned these characters a frequency of 0.

Font Name	Number of files	Number recognized correctly
Gopika	6	6
Krishna	8	8
ITXGuj	5	3
GujaratiLys	7	7
Total	26	24

Table 1.1

Phase 2

In phase 2 of the experiments, I collected more example documents and used six Gujarati fonts – Krishna, ITXGuj, Gopika, GujaratiLys, Kalapi and EFFGuj. Each document was 2 KB in size on an average.

To eliminate the problem of missing characters in the fonts, I created frequency distributions of all the fonts separately by performing frequency analysis on corpora of approximately 10,000 characters in each font.

Results of Phase 2: With these new frequency distributions for the six fonts, experiments were carried out on 153 documents of varying sizes and using different fonts. Euclidean distance was used as a similarity measure as before. The fonts of all 153 documents were recognized correctly as shown in Table 1.2.

Font Name	Number of Files	Number recognized correctly
Gopika	53	53
EFFGuj	17	17
ITXGuj	28	28
Kalapi	10	10
Krishna	37	37
GujaratiLys	8	8
Total	153	153

Table 1.2

Vakharia and the senior faculty members of my department for their constant help and encouragement.

Once correct font recognition is achieved, it is straightforward to convert the document into a “standard” font using a conversion table. I chose the Krishna font as my standard font and converted the documents into it. I also converted the document from the standard font to a Devanagari font which could then be read out by a Devanagari Text-To-Speech system [3].

With the help of my software, a user can now read a document which uses any of the above six Gujarati fonts or have the document read out by a Text-To-Speech system with just the click of a button. This would be useful to a visually impaired user who can now receive a document in Gujarati and to have it read-out to her without having to ask someone to assist in recognizing the font, downloading it and reading it aloud.

Further Work

In Phase 1 of the experiments, two documents were recognized incorrectly. To improve upon the solution for fonts which have similar encoding, I plan to use more sophisticated techniques like double character frequencies, common word analysis and dictionary lookups.

I plan to extend my work to cover more Gujarati fonts. The problem in font encoding exists in other Indian languages as well, for which the same technique can be used.

References

- [1] Sitaram, Sunayana, *Font Recognition : A Neural Network Approach*, Proceedings of the National Symposium on Security and Soft Computing 2007, Surat, India.
- [2] An example of a simple font converter can be found at <http://www.iit.net/ltrc/FC-1.0/fc.html>
- [3] Text-To-Speech system used: Hindi TTS system from LTRC, IIT Hyderabad.

Acknowledgements

I would like to thank Dr. B.R. Sitaram for valuable discussions regarding this project. I also thank the Head of the Computer Engineering Department, SVNIT, Dr. M.A. Zaveri, Dean of Student Counseling, SVNIT, Dr. D.P.