Multi Character Frequency Based Encoding for Efficient Text Messaging in Indian Languages

Manu Seth, Sourya Basu, Shivam Chaturvedi and Rajesh M. Hegde

Indian Institute of Technology Kanpur, India

Introduction

- The communication protocols used by short message service allow for a maximum of 1,120 bits to be sent per message unit.
- The number of characters one message unit can contain depends inversely on the bits required per character.
- Hence the ultimate goal is to reduce the number of bits required per character for any particular language.
- As the Indian languages like Gujarati, Marathi, Hindi and Tamil have more than 165 character we require atleast 8 bits to represent one character.
- Table marker algorithm is proposed which encodes bi-grams based on frequency analysis.
- Results obtained show that the algorithm achieves 6.5 bits per character for all the Indian language listed above.

Construction of Encoding Tables

- Bi-grams formed by the characters with higher probability are assigned Table 1 and the remaining bi-grams are assigned Table 2.
- The characters pairs in the same table are represented by the same number of bits.

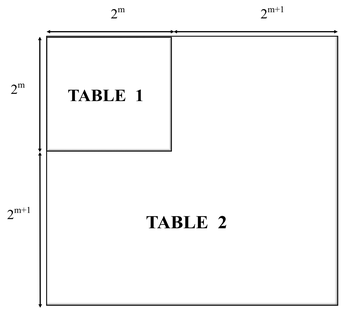

Figure 1: Encoding table structure of the proposed encoding scheme with bi-gram modelling.

Average Number of Bits per Character

Let the total number of Characters in the message =

$$2^m + 2^{m+1} = 3 \cdot 2^m \tag{1}$$

Hence the, total number of pairs that can be formed from $3 \cdot 2^m$ characters is given by

$$(3 \cdot 2^m)^2 = 9 \cdot 2^{2m} = 2^{2m} + 8 \cdot 2^{2m} = 2^{2m} + 2^{2m+3} \tag{2}$$

- Equation 2 suggests a method of segregation of bi-grams into two tables which can be encoded using $2m$ and $2m+3$ bits respectively.
- Since one bit is assigned to denote the table, the average number of bits required to encode two characters lies in the range $[2m+1, 2m+4]$ and it is close to $[2m+1]$.
- This implies that the number of bits required per character for the given encoding scheme is close to $[m+0.5]$

Flowchart of Encoder

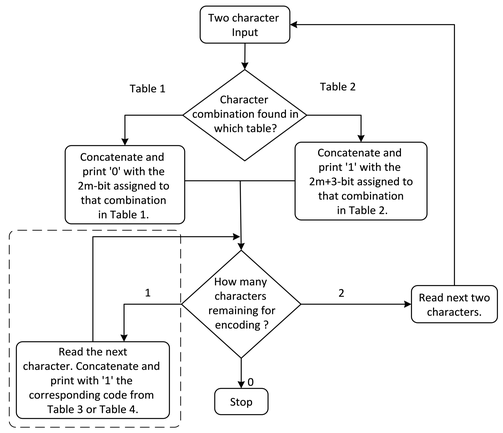

Figure 2: Flowchart of the encoding scheme using multi-character frequency based method. Number of bits to encode a bi-gram from Table 1 and Table 2 are $2 \cdot m + 1$ and $2 \cdot m + 4$ respectively.

Flowchart of Decoder

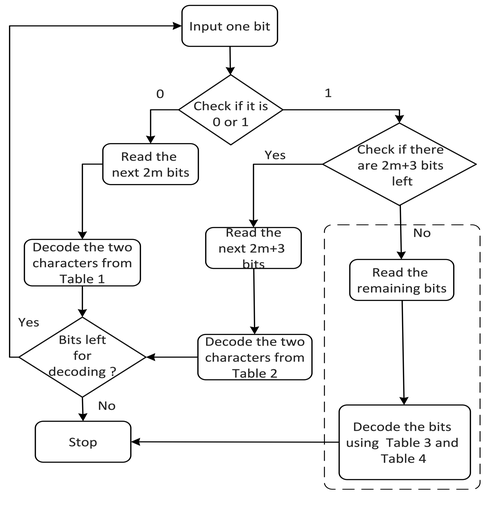

Figure 3: Flowchart representation of the decoding scheme using multi-character frequency based method.

SMS Database for Indian Languages

- SMS text database is not publicly available, a database of tweets from Twitter has been developed.
- Twitter also has a character limit of 140 characters per message similar to SMS text message.

Probabilistic Comparison of Uni-gram and Bi-gram Based Encoding

- Assuming the probability of a random character belonging to Table 3 is p and the probability that it belongs to Table 4 is $(1-p)$. Then, the average bit length of a character using the uni-gram model is given by the

$$p \cdot (m+1) + (1-p) \cdot (m+2) = m + 2 - p. \tag{3}$$

- Thus probability that the character pair lies in Table 1 is p^2 and in Table 2 is $1-p^2$. Therefore, the average bit length of a character using the bi-gram model is given by

$$p^2 \cdot (m+0.5) + (1-p^2) \cdot (m+2) = m + 2 - (3/2) \cdot p^2 \tag{4}$$

- Therefore, the proposed bi-gram encoding scheme gives better results than uni-gram encoding when

$$m + 2 - (3/2) \cdot p^2 < m + 2 - p \tag{5}$$

$$p > 2/3 \tag{6}$$

- This condition has been verified by analysis on short text message database developed for four Indian languages.

Encoding Performance for Hindi

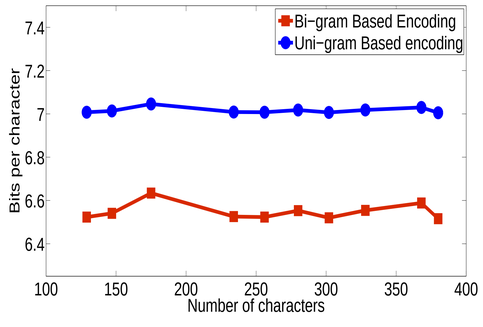

Figure 4: Comparison of encoding efficiency (bits per character) using uni-gram and bi-gram modelling for Hindi language.

Variance Analysis for Hindi

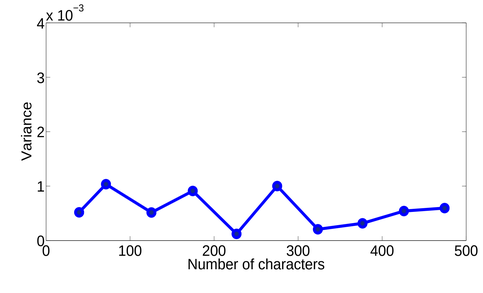

Figure 5: Illustration of variance in bits per character with number of characters in each text message for Hindi language.

Encoding Performance for Four Indian Languages

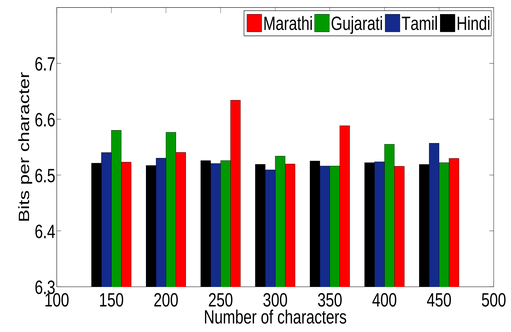

Figure 6: Histogram illustrating the variation of number of bits per character with actual number of characters in a text message for Hindi, Gujarati, Tamil and Marathi languages.

Encoding Scheme for Odd Number of Characters

The structure of Table 3 and Table 4 used to encode the last character when an odd number of characters is parsed in the text message is illustrated in figure below.

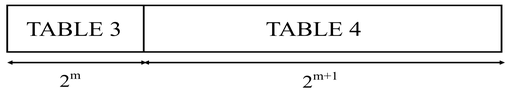

Figure 7: Structure of the encoding table used to code the last character of a text message containing odd number of characters.

- The 2^m most frequent characters are encoded using m bits and are assigned to Table 3.
- The rest of the 2^{m+1} characters are encoded using $m+1$ bits and are assigned to Table 4.

References

1. Jalan, Ankit, Ketan Rajawat, and Rajesh M. Hegde. New encoding schemes for efficient multilingual text messaging. Communications (NCC), 2014 Twentieth National Conference on. IEEE, 2014.
2. Interface, Resource Function Processor MRFP Mp. 3rd Generation Partnership Project; Technical Specification Group Core Network and Terminals; Multimedia Resource Function Controller (MRFC)Multimedia Resource Function Processor (MRFP) Mp Interface; Stage 3. (2012).
3. Adam Green, JavaScript programming for Twitter API 1.1, Lexington, Massachusetts: Adam Green Press.
4. Salomon, David. Data compression: the complete reference. Springer Science and Business Media, 2004.
5. Jones, Gareth A., and J. Mary Jones. Information and coding theory. Springer Science & Business Media, 2012.