Automatic Back Transliteration of Romanized Bengali (Banglish) to Bengali

G. M. Shahariar Shibli[1*], Md. Tanvir Rouf Shawon[1], Anik Hassan Nibir[1], Md. Zabed Miandad[1] and Nibir Chandra Mandal[2]

[1]Ahsanullah University of Science and Technology, Dhaka, Bangladesh.
[2]University of Virginia, Charlottesville,Virginia, United States.

*Corresponding author(s). E-mail(s): sshibli745@gmail.com;
Contributing authors: shawontanvir95@gmail.com;
nibir12011995@gmail.com; zabedmiandad200@gmail.com;
wyr6fx@virginia.edu;

Abstract

Back transliteration of Romanized Bengali to Bengali is the process of converting text written in the Latin alphabet back into the Bengali script. This is often done in order to improve the readability of Bengali text for Bengali speakers by using a simple rules-based system, or an interactive transliteration tool. There are many ways to back transliterate from Romanized Bengali to Bengali, but most of them are either grapheme or phoneme based. This paper introduces a unique pipeline that uses nine open source back transliteration tools to automatically back transliterate Romanized Bengali to Bengali.*The pipeline consists of seven steps: (1) processing the Romanized Bengali input; (2) acquiring human transliteration for performance comparison; (3) employing transliteration tools; (4) generating candidate transliterations; (5) post-processing the candidate transliterations; (6) selecting best candidate transliteration, and (7) evaluating the quality of the transliterations through several performance metrics. Experimental results reveal that our approach produced the highest BLEU-1 score of 81.28, BLEU-2 score of 60.75, BLEU-3 score of 44.45, BLEU-4 score of 30.46, and the lowest average word error rate and Word

*The published version is available at https://doi.org/10.1007/s42044-022-00122-9

Information Lost of 29.21 and 43.68 respectively on **1000** Romanized Bengali texts. In terms of recall, we achieved a Rouge-L score of 0.7190.

Keywords: Transliteration, Back transliteration, Romanized Bengali, Banglish, GPT-3, Google translate, TextRank, Bengali phonetic parser

1 Introduction

Transliteration is a lossless procedure that converts the letters of a source language script into a roughly equivalent, similarly pronounced representation of a target language script. Most transliteration schemes allow the knowledgeable users to determine the original spelling of words that have been transliterated. Forward transliteration is the procedure to convert a word from one language script into its equivalent representation in another language script while retaining the same pronunciation and the reverse procedure is known as backward transliteration. For example, the transliteration of the Bengali sentence "জীবন থেকে নেয়া শিক্ষাই বড় শিক্ষা" into Romanized Bengali is "jibon theke neya sikkhai boro shikkha", and the back transliteration of the Romanized Bengali sentence "Amader apni khoma korben" in source script Bengali is "আমাদের আপনি ক্ষমা করবেন". Sometimes people use Latin script to write Bengali-English Code-Mixed sentences (also known as "Banglish"). For example, "zip file theke apk pabo kivabe?" where "zip" and "file" are English words. These terms are written "জিপ" and "ফাইল" in the transliterated Bengali text "জিপ ফাইল থেকে এপিকে পাবো কিভাবে?" because transliteration is accomplished usually based on pronunciation. Hence, in this work, we consider Bengali-English Code-Mixed language written in Latin script (Banglish) also as Romanized Bengali.

Forward transliteration (Bengali-English) is common among Bengali speakers due to a lack of standard keyboards in the native language and the difficulty in learning the layouts of the ones that do exist. বিজয়[1] (Bijoy) is a frequently used keyboard layout that necessitates some prior familiarity with the keyboard placements of the Bengali letters and the *Avro keyboard*[2] employs a different layout called "Bornona" from OmicronLab, which has a standard English layout and allows users to type in Roman script, which is then automatically transformed into Bengali. However, using "Avro" layout becomes extremely complicated because users need to be familiar with the phonetics of each and every character in order to select the correct transliteration from the options provided for each word. Besides, these keyboard layouts need to be independently installed on different digital devices. Bengali is a complex language due to its irregular phonetics, usage of conjunct consonants (compound letters), and diacritics (markings placed above, below, or occasionally adjacent to a letter in a word to denote a specific intonation). In addition, the Bengali alphabet has many more letters than the English alphabet does. Thus,

[1] http://www.bijoyekushe.net/index.php?action=porichity_bijoy71_win
[2] https://www.omicronlab.com/avro-keyboard.html

the majority of Bengali speakers find it easier to write Bengali using Roman characters when typing on some electronic media or device because they are accustomed to using regular English keyboards like *QWERTY*.

Despite having around 272.7 million speakers and ranking seventh on the list of languages spoken worldwide [1], Bengali is still regarded as a low resource language due to the scarcity of Bengali data in the research community. One of the primary causes of data scarcity is the rising use of Romanized Bengali. However, some recent works utilized Romanized Bengali social media posts/texts alongside Bengali posts/texts, including Event Detection [2], Abusive Content Detection [3], Cyberbullying Detection [4], Sentiment Analysis [5][6], and Named Entity Recognition [6] but most of them did not performed back transliteration (Romanized Bengali to Bengali). Therefore, in order to convert Romanized Bengali texts to Bengali, there is a growing interest in creating reverse transliteration systems.

There have been a plenty of studies conducted to create computational models for back transliteration, however relatively few studies have focused on the English-Bengali back transliteration. To construct such a back transliteration system, grapheme-based machine transliteration approaches [7–11], phoneme-based approaches [12][13], and hybrid approaches [14] were proposed. Sadly, none of these approaches can transliterate perfectly, demonstrating how much more difficult back transliteration from Romanized Bengali to Bengali is than forward transliteration.

There are a few challenges that need to be considered when back transliterating Bengali text. Firstly, the Roman alphabet does not have a one-to-one correspondence with the Bengali alphabet. This means that some letters may be represented by multiple letters in the Bengali alphabet, and vice versa. Secondly, a single target language word might have multiple source language representations and vice versa. Thirdly, as Bengali script has an irregular phonetic structure, a word's pronunciation may not match its orthographic grapheme-phoneme mapping. Finally, the pronunciation of a word might differ from its written form.

In this study, we introduce an unique pipeline to automatically back transliterate from Romanized Bengali to Bengali using nine back transliteration open source tools, which are distinct from the current grapheme and phoneme based techniques. In the proposed methodology, a single input text results in nine separate candidate transliterations. The final candidate transliteration that most closely matches the input is then selected using an effective selection model that ranks the texts on a Bi-directional Encoder Representations from Transformers (BERT) based sentence similarity graph. In addition, we compare the quality of all the generated candidate transliterations using a number of performance metrics, including the Bilingual Evaluation Understudy (BLEU) score, the Recall-Oriented Understudy for Gisting Evaluation (Rouge) score, the Word Error Rate (WER), and the Word Information Lost (WIL). In summary, the following are the primary contributions we made in this paper:

- We have proposed a pipeline to back transliterate Romanized Bengali to Bengali using 9 open source tools and an effective text ranking process to select final transliteration that most closely matches the input.
- We have developed a corpus of 5000 Romanized Bengali texts with their associated Bengali transliterations, and given the GPT-3 free usage restriction, we picked 1000 data points at random and carried out all the experiments to assess how well each transliteration tool performed.
- We have briefly discussed some performance metrics like BLEU, Rouge, WER, WIL which can be used to evaluate the quality of transliterations.

The rest of the paper is organized as follows: section 2 presents some of the related previous works. Details of the dataset creation process is described in section 3. Some important background studies about the available transliteration tools and performance evaluation metrics are discussed briefly in section 4 and section 5 respectively. The proposed methodology is explained in section 6 and the experimental results are explained and analyzed in section 7. Finally, the paper ends with a conclusive remark in section 8.

2 Related Works

Depending on how the input words can be segmented into units for linking them to the corresponding units in the target language, the machine transliteration techniques may be broadly classed as (1) grapheme-based approach, (2) phoneme-based approach, and (3) hybrid approach.

The grapheme-based method primarily focuses on the direct conversion of graphemes from one language to another language without the explicit use of word phonology knowledge of any specific language. Transliteration using graphemes use machine learning methods such as transliteration networks, statistical machine translation, joint source channels, or decision trees. Ekbal et al. [7] and Das et al. [8] provided a strategy that maps two languages with dissimilar origins through direct orthography. It takes advantage of the knowledge of possible Bengali conjuncts and phonetic symbols and their English transliterations. For transliterating from English to Bengali and vice versa, the joint source channel model generative learning algorithm and its variations were employed. In order to automatically detect, retrieve, and acquire transliteration unit pairs from the English and Bengali words, Dasgupta et al. [9] employed a joint source channel model based on a bilingual parallel dataset of transliterated English-Bengali word pairs. At the last stage, the model provides the top 10 outcomes that might result from the supplied input text. Later, on a similar work, Dasgupta et al. [10] used the phrase-based statistical machine translation (SMT) model and the joint source channel model, both of which are grapheme-based techniques, for English-Bengali back transliteration. They claimed that phrase-based SMT marginally outperformed the joint source channel model in terms of performance. Sarkar et al [11] proposed an SVM-based name transliteration approach, where the input is a sequence of source graphemes and the classes are distinctive sequence of target graphemes.

The proposed method was compared to some existing transliteration models that made use of a modified joint source channel model [7] and dealt with forward and backward name transliteration from Bengali to Romanized Bengali.

The phoneme-based method pivots on source phonemes to accomplish the transliteration. In general, there are two crucial processes in the phoneme-based approach: (1) conversion of graphemes to equivalent phonemes, and (2) conversion of phonemes to target language graphemes. Using a phonetic encoding approach to produce interlaying code-strings, UzZaman et al. [12] created a sophisticated English to Bengali transliteration technique that works by matching the pronunciations of the input data with their corresponding outputs. They employed lexicon-enabled phonetic mapping and a double metaphone encoding approach to produce the code strings. The input is transformed into the intended Bengali word if the code strings match the phonetic code strings to just one word in the lexicon. If more than one word matches, it presents pertinent Bengali words that correspond to the input and allows the user to choose the desired result. If no match is discovered, direct mapping is used to translate the input into Bengali. When transliterating Bengali words written in English from their non-standard forms to their standard forms, Chaudhuri [13] used a grapheme-to-phoneme converter to create a pronouncing dictionary starting from a list of frequently used Bengali words taken from a corpus. The proposed method for transliterating a word written in Bengali using Roman character into its regular Bengali form was based on heuristic search strategies over the pronouncing lexicon.

When producing target language transliterations, the hybrid technique combines both source language graphemes and phonemes. Despite the fact that there are certain works that use a hybrid transliteration approach, such as English-Korean [15] and English-Japanese [16], we noticed no research works on English-Bengali. Rather very recently, a three-tier strategy is used by Rizvee et al. [14] to present the unique Three-stage Hybrid Transliteration (THT) framework, in which an English word is first turned into an intermediate phonetic form before being translated into Bengali. In order to achieve a more desirable result, it enhances the spelling of the transformed word using a heuristic runtime dynamic programming (HRDP) algorithm. The THT framework generates a list of potential transliterations from which the machine translation system can select the most appropriate one given the circumstances.

One of the primary issues with the discussed machine transliteration models is that the most of them are inaccessible. Aside from these works, we discovered a number of open source English to Bengali transliteration tools, which are briefly mentioned in section 4.

3 Corpus Creation

3.1 Data Collection

According to our findings, there is a scarcity of rich datasets on Romanized Bengali. Initially, we gathered 5000 Romanized Bengali data manually from various sources, including social media websites and tech blogs. The raw data were mostly gathered from a variety of Facebook Group Post Comments, YouTube Comments, Facebook Captions, and Comments from different Blogging and Article Sites[3].

3.2 Dataset Description

The dataset[4] consists of 5000 Romanized Bengali texts. There are two attributes in the dataset: *"Romanized_Text"* and *"Human_annotated_Bengali_text"*. The *"Romanized_Text"* column provides the raw Romanized Bengali gathered data, while the *"Human_annotated_Bengali_text"* column has the corresponding human transliteration in Bengali. Given the GPT-3 free usage restriction, we picked 1000 data points at random and carried out all the experiments using the pipeline described in section 6.

3.3 Back Transliteration by Human

Transliteration of a Romanized Bengali text into its corresponding Bengali is difficult. For example, a Bengali sentence কেমন আছেন? *(English Translation: How are you?)* can be written in several ways using Romanized Bengali such as *"Kemon achen?", "kamon asen?", "Kamon achan?", "kemon acen?", "kemon achan?"* etc. Humans as well as automatic transliteration systems may interpret these sentences wrongly such as কেমন আসেন?, কামন আছান?, কেমন আচেন? or even ক্যামন আছান?. Such back transliteration is challenging because when a person writes something in the Romanized form, he or she spells it according to how he or she believes the pronunciation of a word sounds and vice-versa. Because of this type of stressful scenario, it is hard to rely on a transliteration made by single human. To mitigate the situation, we performed back transliteration (from Romanized Bengali to Bengali) with the help of three expert annotators. To ensure that they are reliable for the transliteration task, we assessed their trustworthiness score and all three of them have a trustworthiness score higher than 86.667%. In this work, we randomly chose 100 Romanized Bengali sentences (already had Bengali transliteration) as samples from the dataset and constructed 30 control samples to calculate annotators' trustworthiness score. The control samples were simple to comprehend and transliterate. For instance, "apni ki koren?" (In English, "What do you do?"). These control samples were unknown to the annotators. The trustworthiness scores of the three annotators are then calculated individually depending on

[3] https://pastebin.com/3qS9pCKm
[4] https://github.com/nibir1234/banglish_to_bengali

how many control samples were accurately transliterated. Each Romanized Bengali sentence in the dataset is transliterated into corresponding Bengali by two human annotators separately. The final transliteration for each sentence is selected from those two transliterations by the third annotator with the highest trustworthiness score and we consider this selected transliteration as the ground truth for the experiments conducted in this study.

4 English to Bengali Transliteration Tools

4.1 Bengali Phonetic Parser

Phonetic parsing is a computerized algorithmic approach to classical language processing. The *Bengali Phonetic Parser*[5] is a simple phonetic level implementation that generates a phonetic spelling for a word based on Bengali phonetics. We used this Python-implemented package in this study to automatically transliterate Romanized Bengali to Bengali.

4.2 pyAvroPhonetic

The Python adaptation of the popular Bengali phonetic-typing software Avro Keyboard[6] is pyAvroPhonetic[7]. The most recent English to Bengali phonetic typing method is supported by Avro Keyboard. One can write in Romanized Bengali - "ami banglay gaan gai" (English translation: "I sing in Bengali") anywhere and it will be automatically typed in Bengali - আমি বাংলায় গান গাই. The Python package pyAvroPhonetic includes a text parser that transforms Bengali written in Romanic character to its phonetic counterpart in Bengali.

4.3 Google Translate

Google developed Google Translate[8], an extremely large end-to-end multilingual long short-term memory based neural network translation tool. Instead of just learning phrase-to-phrase translations, the Google Nural Machine Translation (GNMT) network attempts crosslingual translation by preserving the meaning of sentences [17]. It supports 133 languages at varying degrees. In this study, we verify that the system can also transliterate a Romanized Bengali sentence into corresponding Bengali sentence. To transliterate our dataset, we have used web scraping with selenium (an open source tools and libraries for browser automation) using google translate where the source and target both language were set to "Bengali".

[5] https://github.com/porimol/bnbphoneticparser
[6] https://www.omicronlab.com/avro-keyboard.html
[7] https://github.com/auvipy/pyAvroPhonetic
[8] https://translate.google.com/

4.4 Indic Transliteration

Indic Transliteration[9] consists of transliteration functions for Sanskrit to convert text written in Latin script to nine Indian scripts such as Bengali, Devanagari, Gujarati, Kannada, Malayalam, Telugu, Tamil, Oriya and Gurmukhi/Punjabi/Panjabi. It is a complex tool that supports eight different romanization styles such as HK, IAST, ITRANS, OPTITRANS, KOLKATA, SLP1, VELTHUIS and WX.

4.5 BNTRANSLIT

BNTRANSLIT[10] is another transliteration tool to transform Romanized Bengali words to Bengali words that utilized deep learning. With batch size of 128, learning rate of 0.001, embedding dimension of 300, and hidden dimension of 512, BNTRANSLIT was trained using Google Dakshina Dataset [18] lexicons train datasets for 10 epochs. It uses an attention-based Long Short Term Memory (LSTM) architecture. We transliterated a whole sentence by tokenizing the words, passing the Romanized words through BNTRANSLIT and combining them to produce the final transliterated sentence. For a particular word, the model can offer the top k transliterations. For example, the top 10 transliterations by the model for the word "aami" (English translation: "I") are আমি, আমী, অ্যামি, আমিই, এমি, আমির, আমিদ, আমই, আমে, আমিতে.

4.6 Google Transliteration IME

Google Transliteration IME[11] is a transliteration typing service for non-Latin alphabet languages (supports 22 languages). It is a virtual keyboard that enables users to directly type in their native language text in any program, removing the need for copying and pasting [19]. It uses dictionary-based phonetic transliteration, which means that whatever Latin characters are input are matched with its vocabulary and transliterated. It also suggests similar phrases for a particular word. To transliterate our dataset using Google Transliteration IME, we have used web scraping with selenium.

4.7 GPT-3

Building rigid Long Short Term Memory (LSTM) machine learning models, which were significantly slower and less accurate, was most desirable option prior to the emergence of transformer-based neural network models. Transformer-based models have gained appeal for several NLP applications, including language translation. Generating text in response to any sort of prompt was the major goal of the invention of artificial intelligence systems like GPT-3 (Generative Pre-trained Transformer 3) [20]. GPT-3 has been pre-trained with 175 billion parameters and 500 billion words from web platforms,

[9] https://github.com/indic-transliteration/indic_transliteration_py
[10] https://github.com/sagorbrur/bntranslit
[11] https://www.google.com/inputtools/try/

making it the most informed model. "Generative pre-training" means that it is trained to predict what the next token will be. GPT-3 has a variety of different underlying engines, each of which can perform tasks at various levels. The 'text-davinci-002' engine is by far the most often utilized model nowadays for almost every GPT-3 task. It is capable of performing tasks with greater accuracy and less training which suggests that it requires fewer language examples on average and less strong language to understand a particular task. In this study, we verified that GPT-3 can transliterate any Romanized Bengali text to corresponding Bengali. For transliteration, we leveraged few shot learning (1-shot, 10-shot and 25-shot) before prompting GPT-3. We set the task description as *"Translate this into Bengali:"*, examples were provided below the task description (for 1 shot) as *"English: version koto vai => Bengali:* ভার্সন কত ভাই*"*, and to prompt GPT-3 we used *"English: Ki Korbo Bolen. => Bengali:"*. The main limitation is that GPT-3 is paid and has free usage restriction.

5 Transliteration Evaluation Metrics

5.1 BLEU score

A metric for comparing a generated sentence to a reference sentence is the Bilingual Evaluation Understudy Score (BLEU) [21]. The score was designed to evaluate the reliability of predictions produced by autonomous machine translation systems. BLEU is concerned with precision: how often the words (and/or n-grams) in the candidate model outputs appear in the human reference. Although it was designed for translation, it can also be used to measure the quality of transliterated sentences. The text of the entire predicted corpus is taken into account while calculating the BLEU score. The method counts the number of n-gram matches between the candidate text and the reference text, where each token corresponds to a 1-gram or uni-gram, and each word pair corresponds to a bi-gram comparison. Individual n-gram scores at all orders between 1 and n are calculated as cumulative scores, and they are then given weight by computing the weighted geometric mean. A perfect match receives a score of 1, whereas a perfect mismatch receives a score of 0. The BLEU score is not perfect since it does not include word meaning, only looks for exact word matches, and ignores word importance and order. However, it is fast and cheap to calculate, language-independent, and, most importantly, correlates highly with human judgement.

5.2 ROUGE Score

ROUGE [22] is an acronym for Recall-Oriented Understudy for Gisting Evaluation which is a collection of measures that can be used to measure the quality of machine transliteration. ROUGE is concerned with how often the words from the human transliteration show up in the candidate transliterations. We have utilized 3 types of Rouge Score for performance comparisons in this study.

1) ROUGE-1: a scoring system based on unigrams (each word) that compares the overlap of each word in the generated and reference transliterations.
2) ROUGE-2: a scoring system based on bi-grams (2-words), which measures how many bi-grams are shared by the reference and generated transliterations.
3) ROUGE-L: a scoring system based on the Longest Common Subsequence (LCS) that naturally considers sentence-level structure similarities. Because it automatically includes the longest common n-grams in sequence, there is no need for a predefined n-gram length.

5.3 Word Error Rate

A common performance metric used primarily for Automatic Speech Recognition is Word Error Rate (WER) [23]. The amount of "errors" in the transcription text generated by an ASR system as compared to a human transcription is calculated by WER. So, why are we integrating the WER metric to evaluate transliteration quality in this study? The reason is that some words may be missed or misinterpreted during back transliteration of a Romanized text (in case of ASR when recognizing speech and transcribing it into text). Word by word comparisons between the reference transliteration and the predicted output by WER determine how many differences there are between the two. In this study, we found that the Romanized word "korchilam" (English translation: "was doing") used in a sentence got different generated transliterations such as করছিলাম, করেছিলাম, কর্চিলাম, করচিলাম, each of which are different words in spelling and pronunciation, where the first word is the actual transliteration but some of the proposed transliterated tools clearly misinterpreted.

5.4 Word Information Lost

WER only states that one system is better than another, not how effective it is. There is no distinction between words that are crucial to the sentence's meaning and those that are not. Furthermore, it makes no distinction between two words that differ only by one character and those that differ fully [24]. Word Information Lost (WIL) [23] overcomes these constraints. WIL is a rough approximation of the percentage of word information lost, has a straightforward probabilistic interpretations, but quantifies the fraction of (mapping sensitive) word information conveyed.

6 Proposed Methodology

This section presents the proposed methodology for transliterating any Romanized Bengali text to corresponding Bengali which is divided into seven major steps. In the proposed methodology, a single input attains nine different candidate transliterations, and the final candidate transliteration that resembles the input most is chosen by an efficient selection model. Furthermore,

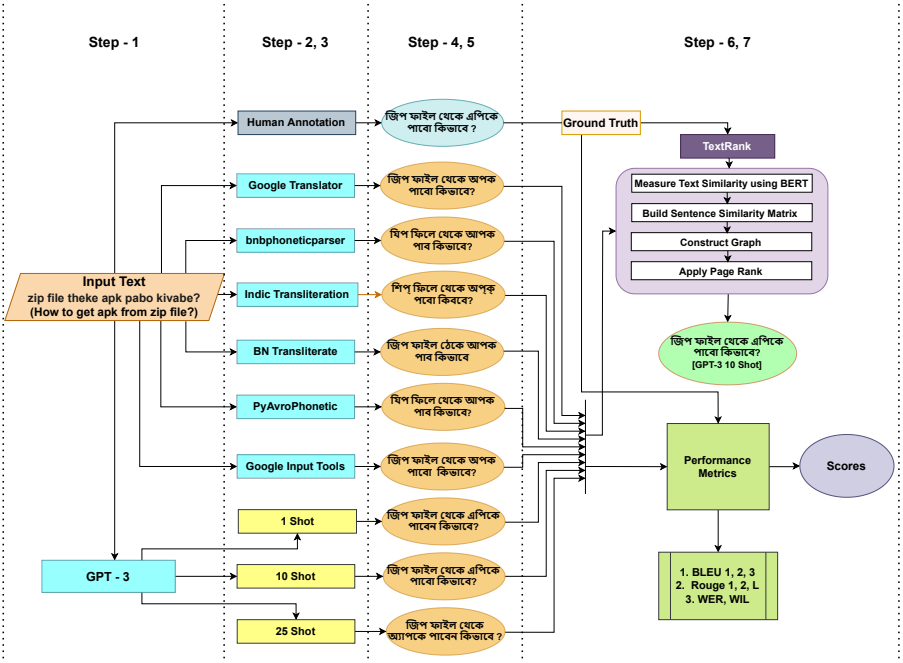

Fig. 1 Methodology using a sample data as input text and transliterated texts as the output for 9 different tools to measure the similarity between Romanized sentence and the ground truth as well as choosing the final transliteration.

through several performance metrics we evaluate the quality of all the candidate transliterations against the reference transliteration. The key phases of the proposed method for transliteration, considering a Romanized Bengali sentence *"zip file theke apk pabo kivabe?" (English translation: "How to get apk from zip file?")* as an example are summarized in figure 1 and are further detailed below.

Step 1) Input Text: Each Romanized Bengali text from the dataset is presented to the proposed model one by one in this step.

Step 2) Acquire Human Transliteration: For each Romanized text, the corresponding transliteration by human is acquired in this step which is considered as the ground truth.

Step 3) Process Transliteration Models: The Romanized text is then passed into 9 different transliteration models. Details of the models are discussed briefly in section 4.

Step 4) Attain Candidate Transliterations: In this step, for a single Romanized Bengali text, nine candidate transliterations from the models used in the previous step are attained. These candidate transliterations are then passed to the post-processing step.

Step 5) Post-processing: In this phase, we solely verify whether all of the punctuations in the input text have been restored or converted; if not, we

restore or convert the punctuations in the candidate transliterations. Punctuation restoration and conversion is accomplished by comparing the input text token by token. For example, the period punctuation "." is transformed to " ।" in Bengali. We also replaced multiple spaces if they did not present in the input. These post-processed candidate transliterations are then placed to steps 6 and 7.

Step 6) Selection using TextRank: The TextRank [25] algorithm was used in this step to select the final transliteration from the candidate transliterations. TextRank is a graph-based ranking method for Natural Language Processing that uses a similarity matrix to measure the degree of similarity between the texts and graphically ranks them using the PageRank [26] algorithm. Taking into account the structure of the inbound links, PageRank develops a ranking of the graph nodes. The similarity matrix is used to represent texts in a network where the vertices are the texts and the edges are the connections between the texts that represent similarity score. The proposed method to select the best transliteration is to measure the similarity of all the candidate transliterations with the reference transliteration (ground truth), construct a graph by building a sentence similarity matrix, apply PageRank and select the first ranked candidate transliteration. The steps involved in ranking the candidate transliterations are:

(a) Text Similarity Measurement: We used the generator checkpoint of the model BanglaBERT [27], a pre-trained Bidirectional Encoder Representations from Transformers (BERT) based language model for Bengali language, to embed the essence of words in densely bound vectors, converting a sentence into a vector where each value inside the vector has a purpose for holding that value. We obtain contextual sentence embedding with BanglaBERT by mean pooling the *last_hidden_state* tensor that yield 768 values for the reference (human transliteration) and each of the candidate transliterations. We utilized cosine similarity [28] to get the similarity score between the reference and a candidate transliterations. This method assesses whether two vectors are roughly going in the same direction by computing the cosine of the angle between the two vectors.

(b) Sentence Similarity Matrix Construction: We construct a 10×10 sentence similarity matrix (we denote as SSM matrix) where index 0 represents the reference transliteration, indexes 1 to 9 represent the candidate transliterations, and for instance, the value in $SSM[0][5]$ contains similarity score between the reference and 5^{th} candidate transliteration (generated by pyAvroPhonetic according to figure 1). We initialized the matrix with all zeros, and then assign similarity scores in only 0^{th} row and column as we are measuring similarity of all the candidates with the reference transliteration, not among the candidates.

(c) Graph Construction and Ranking: In this step, we construct a graph using the SSM matrix where the vertices are the texts and the edges are the connections between the texts that contain similarity score. To construct the

graph, and apply PageRank we used NetworkX[12] which is a Python library for studying graphs and networks. Finally, we sort the candidate transliterations based on their page rank value and select the top valued transliteration as the output. For the example used in figure 1, the final output "জিপ ফাইল থেকে এপিকে পাবো কিভাবে?" is selected from GPT-3 with 10 shot which resembles the input most.

Step 7) Evaluate Candidate Transliterations: In this step, we evaluate the quality of the candidate transliterations using four types of evaluation metrics such as $BLEU-1, BLEU-2, BLEU-3, ROGUE-1, ROGUE-2, ROGUE-L, WER$, and WIL discussed in section 4. These scores are used in section 7 for individual performance comparison and qualitative result analysis of the transliteration tools.

It should be noted that the transliteration methods used in step 3 may generate candidate transliterations that contain misspelled words (figure 1 exhibits this case). As a result, the precise intended result can not be guaranteed by our proposed methodology. Through the calculation of Levenshtein distance and the use of a unigram strategy, Hossain et al. [29] built a system that can identify a Bengali word that has been misspelled and suggest an appropriate replacement. Each phrase is tokenized into words after transliteration, and for each word, unigram strategy will locate all of the associated words from the corpus that contains almost 20,000 words. Unigram strategy takes into account the word that has been used the most frequently or most frequently. It is decided which word has been correctly spelt based on its usage score, which is the highest. To enhance spell checking, Rizvee et al. [14] used the heuristic HRDP algorithm, which is a modified version of the Edit Distance Dynamic Programming technique. One can incorporate one of these ways to fix or improve word spelling, which will result in better transliterations and we leave this open for future work.

7 Experimental Results

With over 272.7 million native speakers, Bengali is one of the most widely spoken languages in the world. Bengali has a complicated grammar that makes it difficult for native speakers to write it fluently, especially on digital platforms where there are few adjustable keyboard layouts. We are outlining a few of the currently available reverse transliteration tools (from Romanized Bengali to Bengali). The tools we have used displayed a wide range of performance variances.

7.1 Experiments

We have already mentioned the tools used in this study, and they are also visible at step 4 in figure 1 where we have illustrated the results of each model using a specific example. The BLEU, Rouge, WER, and WIL scores were used to quantify similarity. For all of the comparisons, we utilized our transliteration

[12] https://github.com/networkx/networkx

Table 1 BLEU scores of all the transliteration models

Back Transliteration Tools	BLEU 1	BLEU 2	BLEU 3	BLEU 4	WER	WIL
GPT-3 (1 Shot)	67.61	41.60	24.97	11.90	0.5538	0.6942
GPT-3 (10 Shot)	72.91	47.34	29.29	15.75	0.5485	0.6652
GPT-3 (25 Shot)	74.51	49.47	30.99	17.41	0.5191	0.6449
Bengali Phonetic Parser	46.17	21.91	9.66	3.64	0.6141	0.7882
Indic Transliteration	10.20	1.28	0.08	0.05	0.9227	0.9784
BNTRANSLIT	62.69	35.37	18.38	8.82	0.5471	0.7057
pyAvroPhonetic	45.21	21.58	8.77	3.54	0.6205	0.7938
Google Transliteration IME	81.28	60.75	43.18	28.02	0.3220	0.4546
Google Translator	79.41	60.47	44.45	30.46	0.2921	0.4368

by human annotators as a reference. Table 1 contains the BLEU scores for transliterated phrases from all tools, with four different versions of the results computed using n-gram models, the average Word Error Rate (WER), Word Information Lost (WIL) and Rouge-L score for each model.

7.2 Performance comparison

The pipeline we are proposing requires validation to demonstrate its superiority over other current approaches capable of performing reverse transliteration from Romanized Bengali to Bengali. To compare the performance, we included an example from section 5 of the paper [12]. Table 3 provides a comparison of back transliteration between [12] and the proposed approach of this work. Here, we can observe that the proposed approach has a BLEU-1 score of 90.70, which outperforms the output of the direct and phonetic mapping stated in [12] by 37.21 and 30.24 respectively.

7.3 Experimental Results and Analysis

We divided the result analysis portion into two parts, the first of which contains all back transliteration tools besides GPT-3. GPT-3 is discussed separately in the other section because, given how it acts when assigned the task of transliteration, it necessitates a bit more care.

7.3.1 Transliteration tools except GPT-3

The best results for back transliteration have been achieved via **Google Translate**. The superiority of this tool is seen in Table 1, where it has a 79.41 BLEU-1 score based on a uni-gram viewpoint, demonstrating its success. It is quite compatible with maintaining uniformity over several words. The assertion is supported by the BLEU-3 and BLEU-4 scores of the Google Translate tool, which among all the tools used in our study achieved the best scores of 44.45 and 30.46 for 3-gram and 4-gram models, respectively. The consistency of this tool is further supported by the Word Error Rate (WER) and Word Information Lost (WIL) scores in table 1 , which indicate that Google Translate has the lowest WER and WIL scores among the other tools at 0.2921 and

Table 2 ROUGE scores of all the transliteration models

Transliteration Model	Version	Recall	Precision	F1-Score
GPT-3 (1 Shot)	r-1	0.504055	0.517834	0.509015
	r-2	0.291359	0.299790	0.294238
	r-L	0.502748	0.516463	0.507690
GPT-3 (10 Shot)	r-1	0.545571	0.553900	0.546676
	r-2	0.335861	0.342182	0.336794
	r-L	0.544730	0.553156	0.545894
GPT-3 (25 Shot)	r-1	0.551265	0.567159	0.555967
	r-2	0.344854	0.356044	0.348105
	r-L	0.550332	0.566095	0.555011
Bengali Phonetic Parser	r-1	0.394252	0.393860	0.393821
	r-2	0.157498	0.157475	0.157385
	r-L	0.394160	0.393761	0.393726
Indic Transliteration	r-1	0.084295	0.084299	0.084229
	r-2	0.008891	0.008996	0.008939
	r-L	0.084295	0.084299	0.084229
BNTRANSLIT	r-1	0.492930	0.496777	0.494016
	r-2	0.254690	0.254647	0.254389
	r-L	0.492847	0.496701	0.493937
pyAvroPhonetic	r-1	0.382900	0.384486	0.383441
	r-2	0.155173	0.155779	0.155385
	r-L	0.382900	0.384486	0.383441
Google Transliteration IME	r-1	0.719235	0.724069	0.7205644
	r-2	0.519912	0.521972	0.520005
	r-L	0.719011	0.723851	0.720343
Google Translate	r-1	0.716266	0.717336	0.716063
	r-2	0.514005	0.513961	0.513577
	r-L	0.716174	0.717238	0.715968

0.4368 respectively. As can be observed in table 2, this Google tool also significantly outperformed other back transliteration techniques in terms of Rouge score.

Google Transliteration IME performed the second best of all tools for back transliteration jobs. It has attained the best BLEU-1 result of 81.28, however it is unable to maintain consistency while competing with Google Translate for more than one word. Google Transliteration IME displays 28.02 in BLEU-4, behind Google Translate by 2.44. The WER and WIL scores support the BLEU score by exhibiting a substantial rise of 2.9% and 1.7% from Google Translate, respectively, indicating Google Transliteration IME's inability to maintain consistency for more than one word consecutively.

Following Google and GPT-3 based tools, **BNTRANSLIT** has shown a satisfactory performance. BNTRANSLIT scored 62.69 on the BLEU-1 score and 35.37 on the BLEU-2. These results are pretty impressive considering that

Table 3 Performance comparison between existing approaches and our methodology using an example from [12].

Tools	Sentences	BLEU 1	BLEU 2
Transliterated Sentence [12]	ami bhalo achi. tomar khobor ki. ajke shondha bela tumi ki korcho. obak bepar holo, ami ekhon bangla likhte pari inglish diye. aro mojar bepar holo ami dui bhabe likhte pari. ekTa DairekT arekTa phoneTik. tomar desh e koto Taka te ek Dolar.	—	—
Human Annotated Back Transliteration	আমি ভালো আছি। তোমার খবর কি। আজকে সন্ধ্যা বেলা তুমি কি করছো। অবাক ব্যাপার হলো, আমি এখন বাংলা লিখতে পারি ইংলিশ দিয়ে। আরও মজার ব্যাপার হলো আমি দুই ভাবে লিখতে পারি। একটা ডাইরেক্ট আরেকটা ফনেটিক। তোমার দেশ এ কত টাকা তে এক ডলার।	—	—
Output in direct mapping [12]	আমি ভালো অছি. তোমার খোবর কি. আজকে শোনধা বেলা তুমি কি কোরছো. ওবাক বেপার হোলো, আমি এখোন বাংলা লিখতে পারি ইনশলিশ দিয়ে.আরো মোযার বেপার . হোলো আমি দুই ভাবে লিখতে পারি একটা ডাইরেকট আরেকটা ফোনেটিক. তোমার দেশ এ কোতো টাকা তে এক ডোলার.	53.49	39.10
Output in phonetic mapping [12]	আমি বহাল/ভাল/ভালো আছি. তোমার খবর কই/কি/কী. আজকে সন্ধ্যা বেলা তুমি কই/কি/কী করছ.অবাক বেপার/ব্যাপার হল, আমি এখন/এখনো বাংলা/বাঙলা লিখতে পারি/পাড়ি ইংলিশ দিয়ে.আর/আরো/আড় মজার বেপার/ব্যাপার হল আমি দুই ভাবে লিখতে পারি/পাড়ি.একটা ডাইরেকট আরেকটা ফোনেটিক. তোমার দেশ/দেষ এ কত/কোঁত টাঁকা/টাকা তে এঁকো/এক ডলার.	60.46	46.47
This work	আমি ভালো আছি। তোমার খবর কি। আজকে সন্ধ্যা বেলা তুমি কি করছো। অবাক ব্যাপার হলো, আমি এখন বাংলা লিখতে পারি ইংলিশ দিয়ে। আরো মজার বেপার হলো আমি দুই ভাবে লিখতে পারি। একটা ডাইরেক্ট আরেকটা ফোনেটিক। তোমার দেশ যে কত টাকা তে এক ডলার।	90.70	85.69

a BLEU-1 score of more over 60 is frequently regarded as superior to humans. In comparison to other tools created specifically for the back transliteration, such as the Bengali Phonetic Parser, pyAvroPhonetic, and Indic Transliteration, it has also achieved a respectable WER and WIL score. Rouge score also demonstrates the back transliteration tool's potential.

Bengali Phonetic Parser and **PyAvroPhonetic** display performance that is very close. Both tools have obtained BLEU-1 values of about 45 and BLEU-4 values of about 3.50. These results show that these tools only provide a subpar transliteration. For these models, the word error rate and word information lost score are both high and more than 0.78. This indicates that roughly 78% of the information is being lost by these techniques. The simplistic phonetic level implementation, which creates a phonetic spelling for a word based on Bengali phonetics, is the cause of the Bengali Phonetic Parser's poor performance. Additionally, pyAvroPhonetic's performance is deteriorating since it is unable to select the best option during the transliteration process, whereas a human can when manually writing with the Avro Keyboard from which it is adapted.

The **Indic Transliteration** tool performs the worst out of all the tools, receiving a BLEU-1 score of only 10.20 and a BLEU-4 value of 0.05. The WIL is 0.9784 and the WER is 0.9227, respectively. It implies that it is losing practically all of the data. Additionally supporting the assertion is the Rouge score, which shows a Rouge-L recall of 0.094295. Figure 1 illustrates how the Indic Transliteration tool is unable to handle complicated letters and diacritics, which are markings that are sometimes placed above, sometimes below, or right next to a letter in a word to indicate a certain intonation in Bengali.

7.3.2 Generative Pretrained Transformer 3 (GPT-3)

Before prompting GPT-3, we used few shot learning (1-shot, 10-shot, and 25-shot) for transliteration. With just 1 sample (1-shot) as a reference, GPT-3 achieved a fantastic score of 67.61 on the BLEU-1 scale. Other than the Google-based tools we previously stated, the outcome it has obtained is superior. The BLEU-1 rises to 72.91 when 10 samples (10-shot) are used as reference, and it rises to 74.51 when 25 examples (25-shot) are used as references. The BLEU-2, BLEU-3, and BLEU-4 score for GPT-3 based models are also increasing dramatically with the growing quantity of reference samples, as shown in table 1. This performance is also supported by WER and WIL scores, with WER for 1-shot and 10-shot scenarios falling from 0.5538 to 0.5191, respectively. The claim is further supported by the rogue score from table 2, where rogue-L is steadily rising from 0.5027 to 0.5503 (Recall) for each increase of the context samples from 1 to 25. The results obtained by GPT-3 demonstrate that it can produce better outcomes with more samples and circumstances. However, by inspecting the output phrases generated by GPT-3, we discovered a whole different aspect of this tool. An example might be a good way to describe GPT-3's behavior. In table 4, the word *"month"* is transliterated in Bengali by human annotators as "মাস্থ" considering the phonetics of every single letter, whereas GPT-3-based tools do "translation" instead of transliteration, hence the word *"month"* is translated to "মাস" in Bengali. We examined each sentence generated by GPT-3 and discovered that it is improving by producing realistic transliteration in some circumstances. It not only does back transliteration but also considers the English words in a transliterated text and translates them to their Bengali meaning. Another notable aspect in the example of table 4 is that it converts *"korcilam"* as "করেছিলাম" in 1-shot and 10-shot learning, but achieves the desired transliteration "করছিলাম" when 25 samples are provided as references. GPT-3 also handles complex letters and punctuation marks quite well, but other models do not because they are simply transliterating phonetically. The output of GPT-3 (10-shot) for the sentence presented in table 3 is a great indication of punctuation mark handling. Considering only the first three sentences of the input Romanized Text, it is apparent that GPT-3 (10-shot) has placed a question mark ("?") after each of the two generated transliterations: তোমার খবর কি? আজকে সন্ধ্যা বেলা তুমি কি করছো?, as it perceives them as interrogative sentences. The results indicate that GPT-3 is not as effective as Google Translate and Google Transliteration IME at phonetic level back transliteration, but it does give us realistic Bengali sentences in which English words are "translated" (rather than transliterated) to their corresponding Bengali meaning and, in some cases, a great deal of improvisation with compound letters and punctuation marks.

Table 4 Back transliteration output comparison among GPT-3 based tools with human transliterated sentence.

Transliteration Tool	Back Transliterated Sentence
Transliterated Sentence	Ai post ami 3 month age korcilam
Human Annotated	এই পোস্ট আমি ৩ মাস্থ আগে করছিলাম
GPT-3 (1 Shot)	এই পোস্টটি আমি ৩ মাস আগে করেছিলাম
GPT-3 (10 Shot)	এই পোস্ট আমি ৩ মাস আগে করেছিলাম
GPT-3 (25 Shot)	এই পোস্টটা আমি ৩ মাস আগে করছিলাম

8 Conclusion

Romanized Bengali is still widely used by Bengali speakers nowadays on blogs and social media. Our suggested method for leveraging the current transliteration tools can be utilized on several internet platforms for the reverse transliteration work from English to Bengali and is a helpful tool for Bengali NLP researchers. Our findings also demonstrate the potential for adopting GPT-3 as a back transliteration tool because it has a remarkable capacity for handling a variety of challenges. The users' spelling errors, which reduce the effectiveness of the transliteration process, have been a major challenge we have encountered throughout our work. A further development of our work might involve correcting the misspelled words before sending them to the pipeline.

9 Declarations

9.1 Ethical Approval and Consent to participate

Not Applicable.

9.2 Consent for publication

Not Applicable.

9.3 Human and Animal Ethics

Not Applicable.

9.4 Availability of supporting data

The links of all the tools used in this work are mentioned in the footnote and the dataset we developed is available at - https://github.com/nibir1234/banglish_to_bengali

9.5 Competing interests

The authors declare that they have no competing interests.

9.6 Funding

No funding was received for conducting this study.

9.7 Authors' contributions

Shibli and Shawon set the research scope, coordinated this research, coded, ran a few experiments, drafted the manuscript. Nibir and Miandad wrote codes, collected data and ran most experiments. Mandal ran a few experiments.

9.8 Acknowledgments

This version of the article has been accepted for publication, after peer review (when applicable) but is not the Version of Record and does not reflect post-acceptance improvements, or any corrections. The Version of Record is available online at: https://doi.org/10.1007/s42044-022-00122-9. Use of this Accepted Version is subject to the publisher's Accepted Manuscript terms of use https://www.springernature.com/gp/open-research/policies/accepted-manuscript-terms.

References

[1] List of languages by total number of speakers. en.wikipedia.org. [Online; accessed 2022-08-25] (2019). https://en.wikipedia.org/wiki/List_of_languages_by_total_number_of_speakers

[2] Dey, N., Rahman, M.S., Mredula, M.S., Hosen, A.S., Ra, I.-H.: Using machine learning to detect events on the basis of bengali and banglish facebook posts. Electronics **10**(19), 2367 (2021)

[3] Sazzed, S.: Abusive content detection in transliterated bengali-english social media corpus. In: Proceedings of the Fifth Workshop on Computational Approaches to Linguistic Code-Switching, pp. 125–130 (2021)

[4] Ahmed, M.T., Rahman, M., Nur, S., Islam, A., Das, D.: Deployment of machine learning and deep learning algorithms in detecting cyberbullying in bangla and romanized bangla text: A comparative study. In: 2021 International Conference on Advances in Electrical, Computing, Communication and Sustainable Technologies (ICAECT), pp. 1–10 (2021). IEEE

[5] Hassan, A., Amin, M.R., Al Azad, A.K., Mohammed, N.: Sentiment analysis on bangla and romanized bangla text using deep recurrent models. In: 2016 International Workshop on Computational Intelligence (IWCI), pp. 51–56 (2016). IEEE

[6] Hossain, M.S., Nayla, N., Rassel, A.A.: Product market demand analysis using nlp in banglish text with sentiment analysis and named entity

recognition. In: 2022 56th Annual Conference on Information Sciences and Systems (CISS), pp. 166–171 (2022). IEEE

[7] Ekbal, A., Naskar, S.K., Bandyopadhyay, S.: A modified joint source-channel model for transliteration. In: Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions, pp. 191–198 (2006)

[8] Das, A., Saikh, T., Mondal, T., Ekbal, A., Bandyopadhyay, S.: English to indian languages machine transliteration system at news 2010. In: Proceedings of the 2010 Named Entities Workshop, pp. 71–75 (2010)

[9] Dasgupta, T., Sinha, M., Basu, A.: A joint source channel model for the english to bengali back transliteration. In: Mining Intelligence and Knowledge Exploration, pp. 751–760. Springer, ??? (2013)

[10] Dasgupta, T., Sinha, M., Anupam, B.: Resource creation and development of an english-bangla back transliteration system. International Journal of Knowledge-based and Intelligent Engineering Systems **19**, 35–46 (2015). https://doi.org/10.3233/KES-150307

[11] Sarkar, K., Chatterjee, S.: Bengali-to-english forward and backward machine transliteration using support vector machines. In: International Conference on Computational Intelligence, Communications, and Business Analytics, pp. 552–566 (2017). Springer

[12] UzZaman, N., Zaheen, A., Khan, M.: A comprehensive roman (english)-to-bangla transliteration scheme (2006)

[13] Chaudhuri, S.: Transliteration from non-standard phonetic bengali to standard bengali. In: Satellite Workshop, p. 41 (2006)

[14] Rizvee, R.A., Mahmood, A., Mullick, S.S., Hakim, S.: Arobust three-stage hybrid framework for english to bangla transliteration. International Journal on Natural Language Computing **11**(1) (2022)

[15] Lee, J.S., Choi, K.-S.: English to korean statistical transliteration for information retrieval. Computer Processing of Oriental Languages **12**(1), 17–37 (1998)

[16] Bilac, S., Tanaka, H.: Improving back-transliteration by combining information sources. In: International Conference on Natural Language Processing, pp. 216–223 (2004). Springer

[17] Schuster, M., Johnson, M., Thorat, N.: Zero-shot translation with google's multilingual neural machine translation system. Google AI Blog **22** (2016)

[18] Roark, B., Wolf-Sonkin, L., Kirov, C., Mielke, S.J., Johny, C., Demirsahin, I., Hall, K.: Processing south asian languages written in the latin script: the dakshina dataset. arXiv preprint arXiv:2007.01176 (2020)

[19] Google IME. en.wikipedia.org. [Online; accessed 2022-08-27] (2012). https://en.wikipedia.org/wiki/Google_IME

[20] Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J.D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., *et al.*: Language models are few-shot learners. Advances in neural information processing systems **33**, 1877–1901 (2020)

[21] Papineni, K., Roukos, S., Ward, T., Zhu, W.-J.: Bleu: a method for automatic evaluation of machine translation. In: Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, pp. 311–318 (2002)

[22] Lin, C.-Y.: Rouge: A package for automatic evaluation of summaries. In: Text Summarization Branches Out, pp. 74–81 (2004)

[23] Morris, A.C., Maier, V., Green, P.: From wer and ril to mer and wil: improved evaluation measures for connected speech recognition. In: Eighth International Conference on Spoken Language Processing (2004)

[24] Errattahi, R., El Hannani, A., Ouahmane, H.: Automatic speech recognition errors detection and correction: A review. Procedia Computer Science **128**, 32–37 (2018)

[25] Mihalcea, R., Tarau, P.: Textrank: Bringing order into text. In: Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing, pp. 404–411 (2004)

[26] Page, L., Brin, S., Motwani, R., Winograd, T.: The pagerank citation ranking: Bringing order to the web. Technical report, Stanford InfoLab (1999)

[27] Bhattacharjee, A., Hasan, T., Samin, K., Islam, M.S., Rahman, M.S., Iqbal, A., Shahriyar, R.: Banglabert: Combating embedding barrier in multilingual models for low-resource language understanding. arXiv preprint arXiv:2101.00204 (2021)

[28] Han, J., Kamber, M., Pei, J., *et al.*: Getting to know your data. In: Data Mining, vol. 2, pp. 39–82 (2012). Morgan Kaufmann Boston, MA

[29] Hossain, M.M., Labib, M.F., Rifat, A.S., Das, A.K., Mukta, M.: Auto-correction of english to bengali transliteration system using levenshtein distance. In: 2019 7th International Conference on Smart Computing &

Communications (ICSCC), pp. 1–5 (2019). IEEE