HBRP PUBLICATION

Recent Trends in Computer Graphics and Multimedia Technology
Volume 4 Issue 1

An Approach towards a Bilingual Keyboard Application

Mohit Kumar Rathod[1], Nidhi Karambelkar[2], Monika Verma[3*]
[1,2]Student, [3]Professor
Department of Computer Science and Engineering, BIT Durg, India.
****Corresponding Author***
E-Mail Id:- *monika.verma@bitdurg.ac.in*

ABSTRACT

In context to all the advancements to date towards translation of one language to another through machine translation, there have been languages that haven't yet been explored, while others have at times not been efficient and semantically correct according to the human inputs. In this paper, we present a keyboard application, particularly for Hinglish Speaking users that can cope with the present scenario of free typing in Hinglish and is equipped with word prediction, spelling correction, and emotion prediction. The primary aim of this keyboard application is to enable Hinglish typing users to type freely without any errors and provide then an interface to ease their daily works. We have also tried to improve efficiency and have a broader dataset that stores all forms of user inputs as well as checks the semantics for more authentic conversions and translations.

Keywords:-*Hinglish keyboard application, natural language processing, machine translation, word prediction, spelling correction, emotion prediction, deep learning on text, spelling correction.*

INTRODUCTION

Communication, through phrases, speech, and written text, is one such ability that only humans possess, which makes them unique creatures to exist. The need for communication has led people from various backgrounds, with linguistic and social differences to interact, despite the language being a barrier.

Also, with the growing technology, and the expansion of civilization and mobility, interaction through various multimedia and electronic devices instead of face-to-face conversations has become a resourceful need. But with it, comes the need for correct meaningful translations, a set of rules to abide by to have the most appropriate conversations and this ability in humans is not very dependable and concrete. In recent times, it is observed that putting up a message on someone's wall, or any communication through texts, has put ease to the way of interaction and has guaranteed zero disruption on the receiver's end. And so, the process of written communication, official paperwork, as well the general chit chat with singular or groups has started finding its way through written texts, it being more proof worthy and efficient.

But the problem is that not every user is equipped with the language medium most prominently, neither are we provided with the best-furnished translation tools we can rely upon and so comes the need for such a tool, which enables us to type in our known language in our way, and then translate it in the medium language in the most meaningful way and can compute all the other possible outputs. While progress has been made in recognizing text, forming language datasets, and even developing keyboards for some languages, a tool that can recognize, predict, validate and store the user's inputs in the way they

consider and convert them into another language, is yet to come.

In this study, we thus focus on developing a keyboard application, better called a Hinglish keyboard that would accept the user inputs in the Hindi English mix language and would convert it to proficient English terms or generate a proper error-free Hinglish text depending upon the needs of a user.

An Indian usually converses in a language that's neither pure Hindi nor pure English. And so, we want to propose an idea that would help them grow to keep their natural habits intact. We focus on applying some deep learning algorithms, text processing, and mining, and natural language processing for the collection, refinement, and formation of a hybrid version of a dataset that makes the keyboard efficient.

LITERATURE SURVEY

To understand the user inputs and their translation into the machine language, initial research was done for the different patterns of text input methods. The diverse Indian languages, but a similar way of writing, led to the conclusion that a principle design can be made, that can be referred to for other languages, as well as for mapping the closest to human intuition. For this, the RICE TRANSLITERATION SCHEME was used which proved to be a better mapping technique with better ease of remembrance.

Also, for understanding the text prediction problem, approaches like the edit distance-based text prediction and text input were applied to form an efficient system[1]. As keyboards in any specific language were made, it was seen that not every language is completely in its pure form when used while any conversation. In a precise way, it was observed that a mixed language existed, which comprised of words from multiple or at least two languages, which was neither the host language nor the foreign language. And so, machine translation of any such text input and its mapping became a difficulty. And hence research and examinations were held to understand the presence of constraints for code-mixing and the existing guidelines to understand the presence of a foreign language in the dataset.

Here came the concept of Morphological Analyzer into which the mixed code was fed and was labelled into the machine-translated language dataset. The algorithm to take a mixed code, segregate the code into a simple host and foreign-language set, then feeding one of the languages to the analyzer, studying it, and converting it to the other language was an efficient way to evaluate the mixed codes.[2] Machine Translation brought with it various new methods and algorithms that can be used for better, faster, and more authentic conversions[3]. Word translations, the concept of decoding and encoding a language, the formation of bilingual, multilingual, and cross-lingual corpuses using Neural Machine Translation gave clarity on the criteria, based on which the further work must be done i.e. Principle-based, knowledge-based, lingual rules-based, and so on. It also suggested the steps for the same and gave a comparative study of the different ways used for better performance[4].

This gave an insight into the concept of deep learning methods with word embedding and back translations[5]. Then, bilingual parallel corpora defined the need for language datasets, as well the text mining approach for language processing which gave birth to the research field in Natural Language Processing methods or NLP for Indian Languages, which is now going on a stronger level for a decade or so[6]. As the translation techniques improved, ways of memory optimization and reliable translation were focused upon

through semantic analysis which focused only on syntax-based translation. It presented the theory that correct syntactic translation of simple words and sentences only would help in the translation of complex ones correctly. It divided the dataset into small chunks that were checked for syntax error, which were corrected and all of it was combined for a better translation[7]. Hand-writing recognition, then gained a lot of attention and research, through neural networks and statistical and dynamic approaches using colloquial Language Recognition and phonetic Spelling variation Transliteration where using Recurrent Neural Networks, CTC, BiLSTM a proper handwriting recognition system was developed[8].

PROBLEM IDENTIFICATION

It was observed from our study, that in India, for some regional languages, the keyboard has been made, like Telugu, with their letters scripted as key[3], while for Balinese Language, a non-QWERTY keyboard layout has been formed[9], and for many others, the main aim for keyboard application being data collection and recognition. However, any such approach is not implemented yet for the two most spoken languages: Hindi and English. And so, we identified that a tool is required that would work for the Hinglish (Hindi + English) users, would help them translate their words to the other language with suggestions and corrections, including their styles of writing. Also, machine translation systems if made have not yet been decentralized to be used by the local public. They are a possession for those who are better versed with it. As a result, we aim to make the keyboard in such a way, that it's viable for the common people.

METHODOLOGY

For any keyboard to function, the needed basics include the development of a keyboard, for which we need input intake, processing, and output display. However, for machine translation, we do need a dataset of the languages used, which can be fed to the memory. So, for our convenience, the whole process is broken into the following modules and carried out step by step.

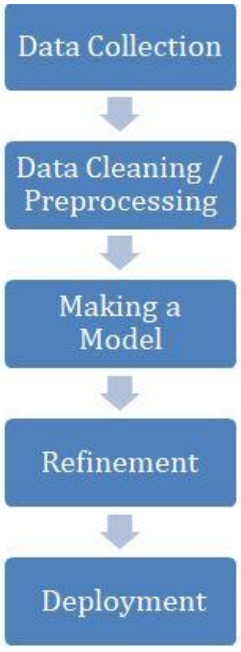

Fig.1:-*Stages involved in Building A Keyboard Application*

DATA COLLECTION

We created a dataset taking multiple inputs from different areas that would help us in our testing. Our basic aim in selecting a dataset was that it contains phrases that give us a wide range of words and also a set of those words that could be spelt differently by different users.

After selecting the data, we arranged it based on their date and time of creation, to have randomness in data for a better and sorted version. The data used is mostly collected from publicly available datasets online. We have focused on the dataset that comprises mostly regular textual data that Indians mostly use while communicating on social media platforms. This data mostly contains the Hindi words written in English.

For ex- "आज का दिन अच्छा गया" is written as "Aaj ka din acha gaya".

These datasets mostly had some problems like many texts were written in Hindi, it also contained emojis along with the text. We needed pure Hinglish text that could be fed into our machines.

DATA CLEANING/PRE-PROCESSING

Once we have collected data, now we aimed to convert the data to be fed into a machine. And since a computer understands a language of 0's and 1's, it was our job to convert the data into a computer-acceptable format with no errors.

There was a basic need for pre-processing textual data and perform the following –

1. Noise Cleaning – This method is used to clear out all the irrelevant symbols from the text[10].
2. Tokenization – Since it is easy to convert tokenized text to computer understandable form.
3. Contraction mapping – Since we mostly write any text in a contracted manner, we needed to expand it.
4. All the above-mentioned data cleaning techniques were implemented simply by using python libraries like NLTK & Tensorflow using Python IDE's[11].

The next task is to understand the textual data. The understanding simple English text is rather easy but trying to understand the Hinglish text is quite a difficult task.

There are no sets of rules defined for Hinglish and every person has his way of using a word in a different context. Traditional methods like POS tagging, shallow parsing, chunking; NER cannot be used in understanding Hinglish Text.

Initial researches were done on the methods of understanding texts. One of these is the Morphological Analysis. It is the segmentation of words into their components called morphenes and the assignment of grammatical information to grammatical categories and the assignment of lexical information to a particular lexeme or lemma[12].

There is a great need to define a new set of rules to the Hinglish language for a better understanding of this language.

A translation schema was provided in research on Machine Translation of Bi-lingual Hinglish Text that will take complex code-mixed sentences in Hinglish and convert it either to pure Hindi or pure English form.

FINDINGS FROM THE DATA

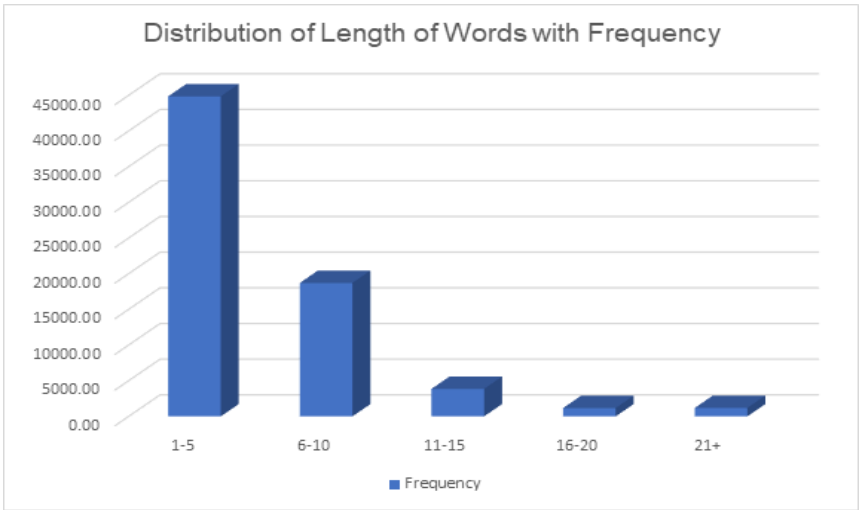

Fig.2:-*Distribution of Length of Words with Frequency*

Fig.3:-*Clustering of Similar Words*

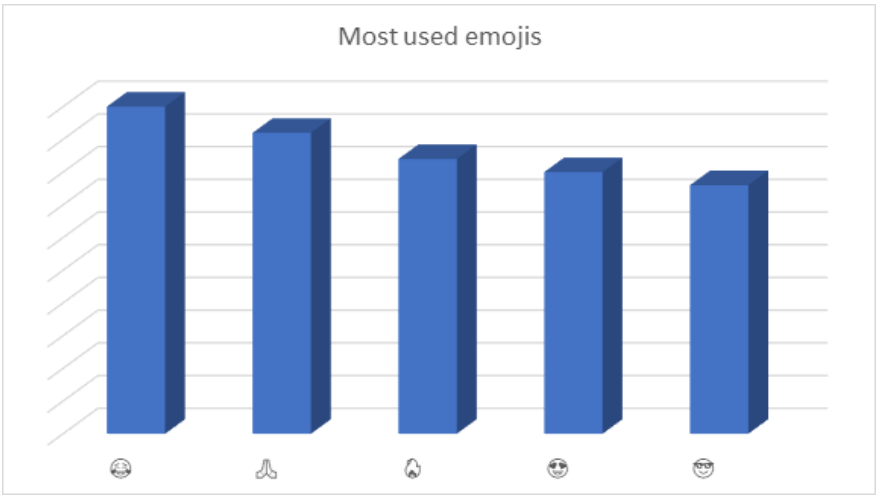

Fig.4:-*Five most used emojis*

Insights from data of over 69000+ lines of words

1. People using Hinglish as their language in typing, generally tend to type shorter messages in multiple lines rather than going to one big message
2. There are many different spellings used by many people for a single word. Understanding those words is a tedious task
3. Uses of emoji are very frequent among the Hinglish typing users. Every 4th line typed by a Hinglish user contains an emoji.
4. Most used emojis are 😂, 🙏, 🔥, 😍, 😎

MAKING A MODEL

A general keyboard application must perform the following tasks:

1. Next Word Prediction
2. Suggestion of Correct Spellings
3. Emoji Prediction

Let us go through the methods we are proposing to perform all the above tasks.

Next Word Prediction model: A smart and successful keyboard application is one that can predict the words even before you type them. Attempts were made to predict the next word specifically to Hinglish. Several Machine Learning models like N-grams, Naïve Bayes Classifier, Support Vector Machines, and Conditional Trees & Random Forests were used to predict the next word precisely. We attempt to use the model that will give us results with the highest accuracy[13]. SVM is known to give the best results among all the methods used in next word prediction[14].

Suggestion of Correct Spellings: The main challenge for correct word prediction in the case of a low-resource language like Hinglish is that there is no formal correct spelling for a word. For instance, the word "han" is written as "haan", "han", "ha" or sometimes simply "h". Furthermore, there are no correct spellings to these words. People use these words at their convenience.

One approach to solve this problem is to identify what spelling of that specific word is most prominently used by that specific user. For that user, that spelling of the word will be the correct spelling. The best way to do this will be to generate a local database. This database will contain all the words used by the user and the correct spelling of the word will be the most used spelling from that dataset.

For the other left-out words, spelling corrections can be performed using Deep Learning Techniques and neural networks[15]. Attempts are already made to provide correct spelling to the words. RNN, LSTM, Character Trigrams models, etc are already in use to generate the correct spelling of the words[13] [16]. NLP Spell checkers are also been used to try and categorize spelling errors as non-word errors and real word errors and correct them[17].

But spelling correction in the Hinglish language is an unexplored part.

Emoji Prediction: Nowadays, who doesn't use emoji to show emotions? It is the easiest way to represent one's current state of emotions. Along with predicting texts, it is of the utmost importance that we predict the correct emoji.

To perform this, it is required that we can understand the sentiment of a person by looking at the text. Several attempts of sentiment analysis have already been made to understand the text. BiLSTM Classifier, CNN Based Classifier, SVM, and other computational approaches are generally used to understand the sentiment of a text [18] [19]. After understanding the sentiment, prediction of correct emoji can be easily performed[20].

Many emojis currently in use do not require sentiment analysis to be performed. By writing a word like "house" must predict an emoji related to a house.

To perform all these tasks in parallel, we need to use a hybrid algorithm that can perform all the tasks simultaneously. We are working on developing such a hybrid algorithm that can perform all the above tasks all at once.

Using a simple application, that would reduce the extra manual efforts and would even allow them to explore, needs to have multitasking features. This is our quest, which we are trying to implement through a combination of every feature, word prediction, emotional classification, transliteration, a faster speed, and optimized memory management, with hybrid techniques, that may help the users by providing them with a standard, grammatically correct output, with inputs closer to their understanding and normal day to day conversations.

With multiple properties, we would try that lessens the switching time between applications, to give a solution for enhanced textual conversations. Also, an application may mean that it is available to the general and remote public, thus providing a solution to issues emerging due to language differences in writing.

REFINEMENT

Once we have made our model, it is now to be optimized, i.e. we have to increase the efficiency of the model. Parameter Tuning is the process of model refinement that is conducted by making modifications to hyper parameters values. Once our hyper parameters are tuned to some better values, training, and evaluations are done again and this process is repeated until we get a model with suitable outcomes.

DEPLOYMENT

Lastly, to get all this work into the picture, we need to deploy our model into a real-time mobile keyboard application. We propose an Android & iOS smart phone application that will perform all the operations.

Making an Android application needs the help of Android Studio IDE. Core Programming in java will be used to run Deep Learning based model on Android Smartphones[21].

iOS apps can be developed and deployed on an iOS device or iPhone only via a macOS machine that runs the Xcode IDE. iOS apps are developed using the Swift or Objective-C programming language[21].

ALGORITHMS

Process	Algorithms / Methods
Data Collection	Social Media, Web Scrapping
Data Cleaning / Pre-processing	Noise Cleaning, Tokenization, Contraction Mapping, Stemming, Stop Words Identification, Morphological Analysis
Making a Model	N-grams, SVM, RNN, LSTM, Character Trigrams model, BiLSTM Classifier, CNN Based Classifier
Deployment	Android Studio, Xcode IDE

Table 1:-*Algorithms used in various stages of processing*

FUTURE WORK

Once we have successfully surpassed our prime aim of the word(s) proper prediction and acceptance of user inputs so that the generated output is grammatically approved, we would work upon finding out the finest of the methods and algorithms, that can be applied to generate the output with minimal conversions, and greater efficiency with all the testing systems, with lesser errors in the prediction of words and emotions, keeping a balance that the speculations follow a more generalized pattern and providing the user with a more understandable yet a compact set of predicted words so that maintaining the local dataset for a wider range of public doesn't crash or slow down the system.

On proper functioning, we would try to inculcate graphics and stickers into our database, and also stronger, smart, predictive transliterations, which would help the users, switch to typing and learning the better-equipped language.

On successful results for this one model, we would try to work for other languages, that need attention, exploration and are in a wider range of public demand.

If feasible in any context, we would try to apply this translation scheme to handwritten text as well, so that the manual texts and scripts could be collected, translated, and preserved to a more genuine form. Voice recognition and translation, which may or may not suffice with the current features in the most constructive way, would be another area of exploration for us.

CONCLUSION

This paper presents a set of steps combined that can be deployed and implemented for a translation-based application, with extra features. If we see through the process of identification of tokens, their groupings, their translations, their easy access and storage, their proper use and conversion according to tenses and nouns used, the emotional context and meanings attached, often gives us a pool of sections to study and deal with.

However, there have been many researches on various segments of these processes. And so, we have provided the basic procedure that can be followed to amalgamate it all into a single application where according to us, the most feasible algorithms and techniques are mentioned for a more user-friendly and optimized result. However, languages are an infinite treasure. More and more words and phrases keep getting added, and accordingly, the operating procedures for an efficient result may keep changing.

Ours is a creative research area, where we have tried to explore and attach the already existing systems with our ideology, however, a translation machine with continuous improvement is hard to achieve. And the primitive aim of a coherent and speedy translation tool that can be a blessing for humans is still an ongoing process that we are trying to decipher through our proposed plan.

ACKNOWLEDGEMENT

It is a matter of gratitude for us that we are able to undergo the task of writing a research paper. We are really thankful to our mentor Professor Monika Verma Ma'am, who believed in us, inspired us and guided us from the start till end, giving us a direction on how to proceed, constantly supporting us to do better.

We would like to thank our faculties from the Computer Science Department, BIT, Durg (C.G), who constantly kept helping us and gave the much required information and demonstrations which motivated us and gave us the moral support to take up this task. I would like to thank all the

people who in some or the other way kept boosting us and reminded us to give our best. A special mention of all the educational resources and tools which have helped us gather the information to have a better idea of every needful and insightful data and information. Finally, we would like to thank our parents, for being the strongest source of strength always by our side.

REFERENCES

1. B., S. V., & Varma, D. V. (2008). TEXT INPUT METHODS FOR INDIAN LANGUAGES. 5-10.
2. Dwiwedi, S. K., & Sukhadeve, P. P. (2010). Machine Translation System in Indian Perspectives. *Journal of Computer Science 6 (10): 1111-1116, 2010*, 1113-1115.
3. Gupta, V. K. (2019). "Hinglish" Language - Modeling a MessyCode-Mixed Language. 1-3.
4. Islam, M. A., Islam, A. B., & Anik, M. S. (2017). Polygot: An Approach Towards Reliable Translation By Name Identification And Memory Optimization Using Semantic Analysis. *978-1-5386-3288-8/17/$31.00 c 2017 IEEE* , 1-4.
5. Jayan, J. P., R R, R., & Rajendran, D. S. (2011). Morphological Analyser and Morphological Generator for Malayalam - Tamil Machine Translation. *International Journal of Computer Applications*, 15-17.
6. K. Sinha, R. M., & Thakur, A. (2005). Machine Translation of Bi-lingual Hindi-English (Hinglish) Text. 149-151.
7. LEE, C., & YANG, H.-C. (2001). TEXT MINING OF BILINGUAL PARALLEL CORPORA WITH A MEASURE OF. *0-7803-7087-2/01/$10.00 0 2001 TEEE*, 470-473.
8. Mitra, S., Singh, V., Sahu, P. P., Veera, V., & Venkatesan, S. M. (2018). Accommodating Phonetic Word Variations Through Generated Confusion Pairs for Hinglish Handwritten Text Recognition. *Accommodating Phonetic Word Variations*, 101-104.
9. Pramartha, C., & Dwidasmara, I. B. (2014). The Composition Approach Non-QWERTY Keyboard for Balinese Script. *IEEE Canada International Humanitarian Technology Conference - (IHTC)* (pp. 1-3). IEEE.
10. Kubica, J., & Moore, A. (2003). Probabilistic Noise Identification and Data Cleaning. *Third IEEE International Conference on Data Mining (ICDM'03)* (pp. 1-2). IEEE.
11. Gharatkar, S., Ingle, A., Naik, T., & Save, A. (2017). Review Preprocessing Using Data Cleaning And Stemming Technique. *International Conference on Innovations in information Embedded and Communication Systems (ICIIECS)* (pp. 1-3). IEEE.
12. P.J*, A. (2013). Machine Translation Approaches and Survey for. *Computational Linguistics and Chinese Language Processing, Vol. 18*, 47-57.
13. Mahajan, M., Beeferman, D., & Huang, S. (1999). IMPROVED TOPIC-DEPENDENT LANGUAGE MODELING USING INFORMATION RETRIEVAL TECHNIQUES. *IEEE* , 541-542.
14. Sasidhar, T. T., B, P., & K P, S. (2020). Emotion Detection in Hinglish(Hindi+English) Code-Mixed Social Media Text. *Third International Conference on Computing and Network Communications*, 1347-1350.
15. Kaur, G., Kaur, K., & Singh, P. (2019, March). Spell Checker for Punjabi Language Using Deep Neural Network. In *2019 5th International Conference on Advanced Computing & Communication Systems (ICACCS)* (pp. 147-151). IEEE.

16. Rakib, O. F., Akter, S., Khan, M. A., Das, A. K., & Habibullah, K. M. (2019, December). Bangla word prediction and sentence completion using GRU: an extended version of RNN on N-gram language model. In *2019 International Conference on Sustainable Technologies for Industry 4.0 (STI)* (pp. 1-6). IEEE.
17. Singh, S., & Singh, S. (2018, March). Review of real-word error detection and correction methods in text documents. In *2018 second international conference on electronics, communication and aerospace technology (ICECA)* (pp. 1076-1081). IEEE.
18. Binali, H., Wu, C., & Potdar, V. (2010, April). Computational approaches for emotion detection in text. In *4th IEEE international conference on digital ecosystems and technologies* (pp. 172-177). IEEE.
19. Sehgal, A., & Kehtarnavaz, N. (2019). Guidelines and benchmarks for deployment of deep learning models on smartphones as real-time apps. *Machine Learning and Knowledge Extraction*, *1*(1), 450-465.
20. Sproat, R., & Jaitly, N. (2016). RNN approaches to text normalization: A challenge. *arXiv preprint arXiv:1611.00068*..
21. Sun, H., Wang, R., Chen, K., Utiyama, M., Sumita, E., & Zhao, T. (2020). Unsupervised neural machine translation with cross-lingual language representation agreement. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, *28*, 1170-1182.